

Integration of Machine Learning and Building Information Modelling for predictive railway maintenance

By

Jessada Sresakoolchai

A thesis submitted to the University of Birmingham for the degree of Doctoral of Philosophy

School of engineering

Department of civil engineering

University of Birmingham

August 2023

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

© 2023 University of Birmingham

All rights reserved. This copyright is held by the author. No part of this thesis may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the author.

Abstract

The aims of this thesis are to integrate machine learning or artificial intelligence (AI) and building information modelling (BIM) for predictive railway maintenance. The study process starts by collecting data. Data sources in this thesis consist of two main sources which are field data and numerical data. For field data, it is mainly from MRS Logística S.A. which is a railway authority in Brazil. Examples of field data are track design, track measurement, defect inspection report, maintenance report, and maintenance manual. For numerical data, data are numerically generated using simulations such as multi-body simulations and finite element models. To ensure that numerical data are reliable and realistic, every simulation and finite element model is validated using relevant field data. If differences between numerical data and field data are in the acceptable range, it demonstrates that numerical data can be used as representatives of field data. An advantage of using numerical data is it can create data diversity and variation which is required for machine learning model development. Then, data are prepared and processed to make them in forms that can be used to train machine learning models. Different machine learning models will be used in this thesis. The techniques can be grouped into three types which are supervised learning, unsupervised learning, and reinforcement learning. The application of these machine learning models is to develop predictive maintenance in the railway system such as defect detection, defect severity evaluation, deterioration prediction, and maintenance plan preparation. To fulfil the aim of the thesis, BIM models are developed and integrated with machine learning which can be conducted using different techniques. Results from the study

show that the developed machine learning models can fulfil their purposes with high performance and the developed BIM models can be fully integrated with machine learning as a data management platform. Contributions of the study are the developed approach can improve the maintenance efficiency and support predictive maintenance in the railway, machine learning models can support decision making and have high performance according to their functions, and the railway system can improve reliability, availability, maintainability, safety, including passenger comfort when applies the developed approach proposed in this thesis.

Acknowledgement

I like to thank my supervisor, Dr Sakdirat Kaewunruen, for his support and guidance in my PhD. His expertise and wisdom are invaluable to me.

I am grateful for the financial support provided by the Thai government, which made this research possible.

Special thanks to Andre Luis Oliveira de Melo for providing valuable data used in this thesis.

Many thanks to my friends in the TOFU research group for the friendship and memorable times during my stay in Birmingham, in particular, Junhui Huang, Chayut Ngamkhanong, Panrawee Rungskunroch, Dan Li, Xu Huang, Hao Fu, Xia Qin, Ting Li, Mehmet Zahid Hamarat, and Pasakorn Sengsri.

Finally, I would like to thank my family for their encouragement and support throughout this challenging path. Their unwavering belief in me has been a constant source of strength and inspiration.

List of publications

This doctoral thesis is written in the format of a collection of articles. This doctoral thesis is predominantly constituted of the following significant scientific articles:

Journal papers

Sresakoolchai, J. and Kaewunruen, S., 2021. Wheel flat detection and severity classification using deep learning techniques. *Insight-Non-Destructive Testing and Condition Monitoring*, *63*(7), pp.393-402.

Sresakoolchai, J. and Kaewunruen, S., 2021. Detection and severity evaluation of combined rail defects using deep learning. *Vibration*, *4*(2), pp.341-356.

Sresakoolchai, J. and Kaewunruen, S., 2021. Integration of building information modeling and machine learning for railway defect localization. *IEEE Access*, *9*, pp.166039-166047.

Sresakoolchai, J. and Kaewunruen, S., 2022. Railway defect detection based on track geometry using supervised and unsupervised machine learning. *Structural health monitoring*, *21*(4), pp.1757-1767.

Sresakoolchai, J. and Kaewunruen, S., 2022. Prognostics of unsupported railway sleepers and their severity diagnostics using machine learning. *Scientific reports*, *12*(1), pp.1-10.

Sresakoolchai, J. and Kaewunruen, S., 2022. Track Geometry Prediction Using Three-Dimensional Recurrent Neural Network-Based Models Cross-Functionally Co-Simulated with BIM. *Sensors*, *23*(1), p.391.

Sresakoolchai, J., Hamarat, M. and Kaewunruen, S., 2023. Automated machine learning recognition to diagnose flood resilience of railway switches and crossings. *Scientific reports*, *13*(1), p.2106.

Sresakoolchai, J. and Kaewunruen, S., 2023. Railway infrastructure maintenance efficiency improvement using deep reinforcement learning integrated with digital twin based on track geometry and component defects. *Scientific Reports*, *13*(1), p.2439.

Conference papers

Sresakoolchai, J., Kaewunruen, S. and Melo, A., 2021, July. Application of Artificial Intelligence (AI) for Prediction of Rail Infrastructure Degradation Rate. In *The Second International Conference on Rail Transportation (ICRT2020): ICRT 2020.* Wangjiang Hotel, China.

Sresakoolchai, J., and Kaewunruen, S. (2022). Integration of Building Information Modeling (BIM) and Artificial Intelligence (AI) to Detect Combined Defects of Infrastructure in the Railway System. In *Virtual Conference on Disaster Risk Reduction (VCDRR): VCDRR.* National Institute of Technology, Karnataka, India

Sresakoolchai, J., Hamarat, M. and Kaewunruen, S., 2022, August. Prediction of turnout support deterioration through dynamic traintrack interactions integrated with artificial intelligence. In *Inter-noise 2022: Inter-noise 2022*. Institute of Acoustics, Scotland.

Sresakoolchai, J., Li, T. and Kaewunruen, S., 2022, June. Ballastless track support deterioration evaluation using machine learning. In *The 17th East Asia-Pacific Conference on Structural Engineering and Construction (EASEC-17): EASEC-17*. National University of Singapore.

Sresakoolchai, J., Kaewunruen, S. and Hamarat, M., 2022, June. Railway turnout support deterioration estimation under flooding condition using machine learning. In *21st Nordic Seminar on Railway Technology*. Tampere University, Finland.

Sresakoolchai, J. and Kaewunruen, S., 2023, June. Carbon emission reduction in railway maintenance using reinforcement learning. In *The 8th International Symposium on Life-Cycle Civil Engineering (IALCCE 2023): IALCCE 2023.* Politecnico di Milano, Italy.

In addition, during the period of postgraduate study, the following papers were published:

<u>Journal papers</u>

Kaewunruen, S., **Sresakoolchai, J.** and Zhou, Z., 2020. Sustainability-based lifecycle management for bridge infrastructure using 6D BIM. *Sustainability*, *12*(6), p.2436.

Kaewunruen, S., **Sresakoolchai, J.**, Ma, W. and Phil-Ebosie, O., 2021. Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions. *Sustainability*, *13*(4), p.2051.

Kaewunruen, S., **Sresakoolchai, J.** and Stittle, H., 2022. Machine Learning to Identify Dynamic Properties of Railway Track Components. *International Journal of Structural Stability and Dynamics*, *22*(11), p.2250109.

Kaewunruen, S., **Sresakoolchai, J.** and Peng, J., 2019. Life cycle cost, energy and carbon assessments of Beijing-Shanghai high-speed railway. *Sustainability*, *12*(1), p.206.

Kaewunruen, S., **Sresakoolchai, J.** and Zhu, G., 2021. Machine learning aided rail corrugation monitoring for railway track maintenance. *Struct. Monit. Maint*, *8*(2), pp.151-166.

Huang, X., Wasouf, M., **Sresakoolchai, J.** and Kaewunruen, S., 2021. Prediction of healing performance of autogenous healing concrete using machine learning. *Materials*, *14*(15), p.4068.

Kaewunruen, S., **Sresakoolchai, J.** and Yu, S., 2020. Global warming potentials due to railway tunnel construction and maintenance. *Applied Sciences*, *10*(18), p.6459.

Kaewunruen, S., **Sresakoolchai, J.** and Thamba, A., 2021. Machine learning-aided identification of train weights from railway sleeper vibration. *Insight-Non-Destructive Testing and Condition Monitoring*, *63*(3), pp.151-159.

Kaewunruen, S., **Sresakoolchai, J.** and Lin, Y.H., 2021. Digital twins for managing railway maintenance and resilience. *Open Research Europe*, *1*, p.91.

Huang, X., Zhang, J., **Sresakoolchai, J.** and Kaewunruen, S., 2021. Machine learning aided design and prediction of environmentally friendly rubberised concrete. *Sustainability*, *13*(4), p.1691.

Huang, X., **Sresakoolchai, J.**, Qin, X., Ho, Y.F. and Kaewunruen, S., 2022. Self-Healing Performance Assessment of Bacterial-Based Concrete Using Machine Learning Approaches. *Materials*, *15*(13), p.4436.

Kaewunruen, S., **Sresakoolchai, J.** and Sun, H., 2021. Causal analysis of bus travel time reliability in Birmingham, UK. *Results in Engineering*, *12*, p.100280.

Kaewunruen, S., **Sresakoolchai, J.**, Huang, J., Harada, S. and Wisetjindawat, W., 2021. Human activity vibrations. *Data*, *6*(10), p.104.

Kaewunruen, S., **Sresakoolchai, J.** and Xiang, Y., 2021. Identification of weather influences on flight punctuality using machine learning approach. *Climate*, *9*(8), p.127.

Kaewunruen, S., **Sresakoolchai, J.**, Huang, J., Zhu, Y., Ngamkhanong, C. and Remennikov, A.M., 2022. Machine Learning Based Design of Railway Prestressed Concrete Sleepers. *Applied Sciences*, *12*(20), p.10311.

Borjigin, A.O., **Sresakoolchai, J.**, Kaewunruen, S. and Hammond, J., 2022. Digital twin aided sustainability assessment of modern light rail infrastructures. *Frontiers in Built Environment*, *8*, p.796388.

Conference papers

Harada, S., Wisetjindawat, W., **Sresakoolchai, J.**, Huang, J., Kaewunruen, S. and Sakikawa, S., 2022, June. Real-time intelligent recognition of transportation modes via smartphones. In *World Congress on Railway Research 2022: WCRR2022*.

Contents

1 Introduction 1-1

1.1 Background and research problems 1-1

1.2 Aim and objectives 1-3

1.3 Scope of work 1-5

1.4 Novelties and contributions of work 1-5

1.5 Thesis structure 1-7

2 Literature review 2-1

2.1 Machine learning in the railway maintenance 2-1

2.2 BIM in the railway industry and maintenance 2-9

2.3 Research gaps 2-14

3 Data collection, preparation, and processing 3-1

3.1 Field data 3-1

3.2 Numerical data 3-3

3.2.1 Multibody simulation and validation 3-4

3.2.2 Finite element method (FEM) model development and validation 3-9

3.3 Data processing 3-11

3.3.1 Data frequency 3-17

3.3.2 FFT 3-19

3.3.3 Padding 3-21

4 Machine learning and BIM applications 4-1

4.1 Machine learning 4-1

4.1.1 Types of machine learning and their characteristics 4-1

4.1.2 A general framework of machine learning 4-6

4.1.3 Machine learning development platform 4-9

4.1.4 Python libraries 4-14

4.2 BIM 4-15

4.2.1 BIM software 4-16

4.2.2 Dimensions of BIM 4-18

5 Supervised learning for railway infrastructure maintenance 5-1

5.1 Classification problems in railway maintenance 5-3

5.1.1 Wheelflat detection and severity classification 5-3

5.1.2 Railway combined defect detection and severity classification 5-22

5.1.3 Track component defect prediction based on track geometry 5-39

5.1.4 Unsupported sleepers and the number of unsupported sleeper prediction 5-50

5.1.5 Turnout support condition evaluation under flooding 5-66

5.2 Regression problems in railway maintenance 5-78

5.2.1 Wheel burn localization 5-78

5.2.2 Turnout support deterioration prediction under flooding 5-82

5.2.3 Slab track stiffness prediction 5-86

5.2.4 Track geometry prediction 5-93

6 Unsupervised learning for railway infrastructure maintenance 6-1

6.1 K-mean clustering 6-1

6.1.1 Insights of track component defects based on track design 6-1

6.2 Association rules 6-7

6.2.1 Insights of track component defects based on relevant defects 6-7

7 Reinforcement learning for railway maintenance scheduling 7-1

7.1 Railway maintenance efficiency improvement 7-1

7.2 Carbon emission reduction for railway maintenance 7-25

8 6D BIM and machine learning integration and development 8-1

8.1 BIM model development and machine learning integration 8-1

8.1.1 Civil 3D 8-1

8.1.2 Ferrovia 8-7

8.1.3 Navisworks 8-17

8.1.4 Dynamo and VBA for data integration 8-18

8.2 GUI development for deployment 8-27

9 Machine learning and BIM model deployment 9-1

9.1 Python code 9-1

9.2 EXE 9-3

9.3 Replit 9-5

10 Conclusions and recommendations 10-1

10.1 Research findings 10-1

10.2 Recommendations for future research 10-3

11 References 11-1

List of Figures

Figure 1-1 Thesis structure 1-9

Figure 3-1 Example of wheel-rail contact force from D-Track..... 3-6

Figure 3-2 Example of ABA from D-Track 3-7

Figure 3-3 irregularities available in D-Track 3-9

Figure 3-4 Example of FEM model 3-10

Figure 3-5 An example of data with outliers 3-14

Figure 3-6 Example of data transform using padding 3-14

Figure 3-7 Example of data transform using trimming..... 3-15

Figure 3-8 Example of data transform using max pooling..... 3-15

Figure 3-9 Example of data dimensionality adjustment using stacking 3-16

Figure 3-10 Example of data dimensionality adjustment using a flatten layer..... 3-16

Figure 3-11 Example of processed data using FFT; (a) original data and (b) processed data using FFT 3-21

Figure 3-12 Example of data processing using padding 3-22

Figure 4-1 An example of DNN architecture 4-4

Figure 4-2 An example of CNN architecture for image processing 4-4

Figure 4-3 An example of CNN architecture for time-series data 4-5

Figure 4-4 An example of LSTM architecture 4-5

Figure 4-5 General framework of machine learning 4-8

Figure 4-6 Example of RapidMiner operation 4-10

Figure 4-7 Anaconda platform 4-11

Figure 4-8 VS Code 4-12

Figure 4-9 Example of railway track BIM model developed using Revit 4-17

Figure 4-10 Side views of railway BIM model developed using Revit 4-17

Figure 5-1 Relationship between length and depth of wheelflats 5-8

Figure 5-2 Examples of ABAs from D-Track 5-10

Figure 5-3 Examples of ABAs processed using FFT 5-11

Figure 5-4 Result from pooling layer 5-13

Figure 5-5 The process of the flatten layer 5-15

Figure 5-6 DNN architecture for wheelflat detection and severity classification 5-17

Figure 5-7 CNN architecture for wheelflat detection and severity classification..... 5-17

Figure 5-8 LSTM architecture for wheelflat detection and severity classification 5-18

Figure 5-9 The relationship between epoch and accuracies 5-20

Figure 5-10 Example of ABA when there is no defect 5-26

Figure 5-11 Example of ABA when a track is defective 5-27

Figure 5-12 Geometry of the track with dipped joint (2.5 mm) and short settlement (20 mm) .. 5-27

Figure 5-13 Workflow of a machine learning model for dipped joint and settlement detection and severity evaluation..... 5-29

Figure 5-14 True values and predicted values of dipped joints' size from developed machine learning models..... 5-38

Figure 5-15 True values and predicted values of settlements' size from developed machine learning models..... 5-38

Figure 5-16 Track geometry parameters 5-40

Figure 5-17 Performances of machine learning models on predicting track component defects 5-48

Figure 5-18 FEM model for studying the interaction between rolling stocks and railway infrastructure 5-56

Figure 5-19 Detailed models of railway infrastructure..... 5-56

Figure 5-20 Track irregularity from field data..... 5-57

Figure 5-21 Examples of ABAs for unsupported sleeper analysis (a) the speed is 100 km/h and (b) the speed is 200 km/h 5-62

Figure 5-22 Examples of data processed using FFT (a) the speed is 100 km/h and (b) the speed is 200 km/h 5-63

Figure 5-23 FEM model for turnout support under flooding condition study [155]..... 5-71

Figure 5-24 Example of ABA under flooding events 5-73

Figure 5-25 Example of crossing nose displacement under flooding events 5-73

Figure 5-26 Relationship between actual and predicted values of turnout support stiffnesses .. 5-85

Figure 5-27 FEM model for evaluating slab track stiffnesses 5-87

Figure 5-28 Detailed track structure for evaluating slab track stiffness 5-88

Figure 5-29 Example of ABA for predicting slab track stiffness..... 5-90

Figure 5-30 Irregularity of the Suining-Chongqing section 5-91

Figure 5-31 Relationship between true and predicted values of slab track stiffness 5-92

Figure 5-32 Overview of features used for predicting track geometry parameters 5-98

Figure 5-33 True values and prediction of track geometry parameters (a) superelevation, (b) longitudinal level (right rail), (c) longitudinal level (left rail), (d) alignment (right rail), (e) alignment (left rail), (f) gauge, and (g) twist 5-103

Figure 6-1 Results of k-mean clustering 6-5

Figure 6-2 Tree of clustering 6-5

Figure 7-1 Overview of training process of the reinforcement learning model for maintenance efficiency improvement 7-7

Figure 7-2 Threshold of track geometry parameters 7-10

Figure 7-3 Workflow of A2C model development 7-11

Figure 7-4 Correlation between defects 7-13

Figure 7-5 Track geometry parameter improvement when tamping is performed..... 7-15

Figure 7-6 Track geometry parameter deterioration when tamping is not performed..... 7-16

Figure 7-7 Changes in probability of defect occurrence when rail grinding is and is not performed 7-17

Figure 7-8 Loss from A2C model training..... 7-18

Figure 7-9 Policy entropy from A2C model training 7-19

Figure 7-10 The comparison between field data and the A2C model..... 7-20

Figure 7-11 Examples of the efficiency improvement in railway maintenance (a) maintenance activities, (b) track geometry defects, (c) track component defects, and (d) all defects..... 7-22

Figure 7-12 Examples of superelevation from the A2C model and field data 7-23

Figure 7-13 Sensitivity analysis of (a) learning rate, (b) log(epsilon), and (c) discount factor .. 7-24

Figure 7-14 Overview of training process of the reinforcement learning model for carbon emission reduction..... 7-28

Figure 7-15 Loss from PPO model training 7-30

Figure 7-16 Policy entropy from PPO model training..... 7-30

Figure 7-17 The comparison between field data and the A2C model..... 7-32

Figure 7-18 Examples of the efficiency improvement in carbon emission reduction (a) maintenance activities, (b) defects, and (c) carbon emission 7-33

Figure 7-19 Examples of superelevation from the PPO model and field data 7-35

Figure 8-1 An example of a BIM model created using Civil3D 8-3

Figure 8-2 An example of a BIM model in the format of an IFC file 8-3

Figure 8-3 Property set definitions used to integrate information with BIM models..... 8-4

Figure 8-4 A command used to add the schedule to the BIM model..... 8-5

Figure 8-5 Result of exported schedule 8-5

Figure 8-6 A command to export the schedule to XLS files..... 8-6

Figure 8-7 A window for configuring the exported XLS files 8-7

Figure 8-8 Outcome from DTM and alignment creation 8-8

Figure 8-9 Outcome from profile creation 8-9

Figure 8-10 Cross section of single track 8-9

Figure 8-11 Cross section of double track 8-10

Figure 8-12 Cross section of railway tunnels 8-10

Figure 8-13 3D BIM model created using Ferrovia for single track 8-11

Figure 8-14 3D BIM model created using Ferrovia for double track 8-11

Figure 8-15 3D BIM model created using Ferrovia for railway tunnels 8-12

Figure 8-16 IFC file for single track 8-12

Figure 8-17 IFC file for double track 8-13

Figure 8-18 IFC file for railway tunnels 8-13

Figure 8-19 BIM model with included terrain 8-14

Figure 8-20 BIM model of the rail 8-14

Figure 8-21 BIM model of the base plate 8-15

Figure 8-22 BIM model of the sleeper 8-15

Figure 8-23 BIM model of the ballast 8-16

Figure 8-24 BIM model of the retaining wall 8-16

Figure 8-25 BIM model at the city level 8-17

Figure 8-26 A BIM model integrated with Navisworks 8-18

Figure 8-27 Examples of data integration in BIM models 8-19

Figure 8-28 Hyperlink stored in BIM models to contain time-series data 8-20

Figure 8-29 Dynamo for exporting property set values 8-21

Figure 8-30 Python script for exporting property set values..... 8-22

Figure 8-31 Example of VBA application for including hyperlinks in BIM models..... 8-22

Figure 8-32 Dynamo for updating property set values..... 8-23

Figure 8-33 Dynamo for including hyperlinks in BIM models..... 8-23

Figure 8-34 Dynamo for creating the notification system when defects are detected 8-24

Figure 8-35 BIM model when there are no defects detected 8-25

Figure 8-36 BIM model when defect detected..... 8-25

Figure 8-37 Workflow for BIM and machine learning integration 8-27

Figure 8-38 The GUI developed by using Customtkinter..... 8-29

Figure 8-39 Function enabling when available data are selected 8-30

Figure 8-40 An example of a description of machine learning models' functions..... 8-30

Figure 8-41 An example of reports from pre-trained machine learning models 8-31

Figure 8-42 An example of charts from pre-trained machine learning models..... 8-32

Figure 8-43 Result from adjusting the range of the x-axis 8-33

List of Tables

Table 2-1 Summarized related studies 2-14

Table 3-1 Examples of D-Track inputs..... 3-8

Table 4-1 Examples of Python libraries 4-15

Table 5-1 Depth and length of wheelflats based on levels..... 5-9

Table 5-2 Number of samples in each class..... 5-9

Table 5-3 Precision and recall of wheelflat predictive model 5-20

Table 5-4 Precision and recall of wheelflat detection 5-21

Table 5-5 Precision and recall of wheelflat severity classification 5-21

Table 5-6 Summary of varied parameters for studying dipped joints and settlements 5-25

Table 5-7 Labels of dipped joints and settlements based on sizes..... 5-28

Table 5-8 List of hyperparameters tuned for combined defect detection and severity classification 5-30

Table 5-9 Performance on dipped joint and settlement detection..... 5-31

Table 5-10 Tuned hyperparameters for the CNN model to detect dipped joint and settlement. 5-31

Table 5-11 Performance on dipped joint and settlement detection using separate predictive models..... 5-32

Table 5-12 Tuned hyperparameters for the CNN model to detect dipped joint and settlement separately..... 5-33

Table 5-13 Performance of the predictive models on severity classification 5-34

Table 5-14 Confusion matrix of dipped joint severity classification using the CNN model 5-35

Table 5-15 Confusion matrix of settlement severity classification using the LSTM model..... 5-35

Table 5-16 Tuned hyperparameters for the predictive model to classify dipped joint and settlement severity 5-35

Table 5-17 Performance of the predictive models on defect size prediction 5-36

Table 5-18 Tuned hyperparameters for predicting the size of dipped joints and settlements 5-39

Table 5-19 List of hyperparameters tuned for predicting track component defects..... 5-46

Table 5-20 Precisions and recalls of the DNN and CNN models on predicting track component defects..... 5-48

Table 5-21 Tuned hyperparameters for the predictive model to detect track component defects 5-49

Table 5-22 The comparison of parameters from field data and the FEM model for unsupported sleepers 5-58

Table 5-23 List of hyperparameters for predicting unsupported sleepers 5-59

Table 5-24 Parameter variation for unsupported sleeper conditions 5-60

Table 5-25 Accuracies on unsupported sleeper detection 5-63

Table 5-26 Tuned parameters for unsupported sleeper detection 5-64

Table 5-27 Accuracies on unsupported sleeper severity classification 5-65

Table 5-28 Tuned parameters for unsupported sleeper severity classification 5-66

Table 5-29 Different parameters based on different levels of water under flooding events ... 5-72

Table 5-30 List of hyperparameters tuned for classifying the turnout support deterioration under flooding events 5-74

Table 5-31 Performance on turnout support condition classification under flooding events using ABAs 5-75

Table 5-32 Performance on turnout support condition classification under flooding events using crossing nose displacement 5-75

Table 5-33 Performance on turnout support condition classification under flooding events using ABAs and crossing nose displacement 5-75

Table 5-34 List of tuned hyperparameters tuned for classifying the turnout support deterioration under flooding events 5-77

Table 5-35 Parameter variation for wheel burn localization 5-79

Table 5-36 List of hyperparameters for localizing wheel burns 5-80

Table 5-37 Tuned hyperparameters for localizing wheel burns 5-81

Table 5-38 Performance on wheel burn localization 5-82

Table 5-39 Performance on turnout support condition prediction under flooding events 5-84

Table 5-40 The comparison of parameters from field data and the FEM model for slab track stiffness prediction 5-88

Table 5-41 List of hyperparameters for predicting slab track stiffness 5-89

Table 5-42 Parameter variation for slab track stiffness prediction 5-89

Table 5-43 Performance on slab track stiffness prediction 5-92

Table 5-44 Tuned hyperparameters for predicting slab track stiffness 5-92

Table 5-45 Urgency of response [168] 5-95

Table 5-46 Urgency of response based on longitudinal levels [168] 5-95

Table 5-47 List of hyperparameters for predicting track geometry parameters 5-99

Table 5-48 Performance on track geometry parameter prediction 5-100

Table 5-49 Tuned hyperparameters for predicting track geometry parameters 5-103

Table 6-1 Cluster and sample distribution related to each other 6-6

Table 6-2 association rules with a confidence higher than 0.8 6-10

Table 7-1 Carbon emission calculation for railway maintenance 7-29

Glossary

Term	Explanation / Meaning / Definition
A2C	Advantage Actor Critic
A3C	Asynchronous Advantage Actor-Critic
ABA	Axle Box Acceleration
ADAMS	Automated Dynamic Analysis of Mechanical Systems
AEC	Architecture, Engineering, and Construction
AI	Artificial Intelligence
AMMF	Adaptive Multiscale Morphological Filtering
ANN	Artificial Neural Network
AR	Augmented Reality
BIM	Building Information Modelling
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CPU	Central processing unit
CSV	Comma-Separated Values
DARTS	Dynamic Analysis of Rail Track Structure
DDPG	Deep Deterministic Policy Gradient
DFT	Discrete Fourier Transform
DIFF	Vehicle-Track Dynamic Analysis Model
DNN	Deep Neural Network
DQN	Deep Q-Learning
DT	Decision Tree
DTM	Digital Terrain Model
FCN	Fully Convolutional Network
FEM	Finite Element Method
FFT	Fast Fourier Transform

Term	Explanation / Meaning / Definition
GA	Genetic Algorithms
GB	Gradient Boosting
GBT	Gradient-Boosted Tree
GIS	Geographic Information Systems
GPU	Graphic Processing Unit
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HSR	High Speed Rail
IDE	Integrated Development Environment
IFC	Industry Foundation Classes
IoT	Internet of Things
KNN	K-Nearest Neighbor Classifier
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MBS	Multibody Simulation
MCMC	Markov Chain Monte Carlo
MCTS	Monte Carlo Tree Search
ME	Maximum Error
MGT	Million Gross Tonnages
MPE	Mean Percentage Error
MR	Multiple Regression
MSE	Mean Square Error
NDT	Non-Destructive Testing
NUCARS	New and Untried Car Analytic Regime Simulation
PPO	Proximal Policy Optimization
ReLu	Rectified Linear Unit
ResNet	Residual Network

Term	Explanation / Meaning / Definition
RF	Random Forests
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SD	Standard deviation
SNCF	Société Nationale des Chemins de Fer Français
SUBTTI	Subgrade-Train-track Interaction
SVM	Support Vector Machine
SVR	Support Vector Regression
TGC	Track Geometry Car
TQI	Track Quality Index
TRPO	Trust Region Policy Optimization
VBA	Visual Basic for Application
VIA	Vehicle Interaction with Track Analysis Model
Visual-MBS	Visual-Multi-Body-Simulation
VR	Virtual Reality
VS Code	Microsoft Visual Studio Code
WO	Work Orders
XGBoost	Extreme Gradient Boosting

1 Introduction

1.1 Background and research problems

The demand for railway transportation is increasing over time [1]. To meet the increasing demand, speed and the load of rolling stock have to be increased as well. As a result, railway infrastructure is applied by the increasing load. This is a reason for the rapid deterioration of railway infrastructure. To maintain railway infrastructure in a standard condition, passenger comfort, and safety, railway maintenance has to be done more frequently and efficiently otherwise the infrastructure may suffer from a bad condition or excessive maintenance costs [2]. Based on many studies, maintenance cost is one of the majority costs in a railway project due to long service life and the maintenance stage is also the longest stage in a railway project. Without efficient maintenance, the railway project may suffer from poor serviceability, availability, reliability, passenger comfort, efficiency, and safety of the overall railway system.

Railway operators can apply many approaches to railway maintenance. Three main maintenance approaches are corrective maintenance, preventive maintenance, and predictive maintenance. Corrective maintenance is conducted when something wrong takes place such as defects. Based on the nature of the corrective maintenance, it is unplanned which results in high maintenance costs and disruption because the occurrences of defects or problems are unknown and uncontrollable. In addition, railway operators may be fined by a railway authority based on the disruptions. Therefore, this approach is not desirable nowadays although it is unavoidable sometimes. Preventive maintenance is an approach to keeping everything in good condition. It

is planned to perform routinely. The frequency of the maintenance may be defined based on experience or historical data. It can be seen that preventive maintenance can remove unforeseen events when corrective maintenance is applied [3]. Therefore, defects and problems tend to less occur which is beneficial in terms of maintenance cost and disruption. However, a significant disadvantage of preventive maintenance is excessive unnecessary maintenance costs because railway infrastructures with the good condition may be maintained although they do not need to be maintained. A trendy approach that is developed to dominate the disadvantages of corrective and preventive maintenance is predictive maintenance. Predictive maintenance has a goal to minimize the maintenance as much as necessary to keep railway infrastructure in good condition. It can be done by predicting railway components that possibly have defects in the future and maintaining them before defects or problems occurs. On the other hand, the maintenance activities are limited to components that really need to be maintained. Therefore, excessive maintenance activities are limited as well as defects and problems. However, the deployment of corrective maintenance cannot be possible without potential tools that allow railway operators to predict occurring defects and problems. Together with the limited budget, railway maintenance requires tools for efficient maintenance.

In terms of the amount of information and data, the maintenance stage of a railway project also creates an enormous amount of information and data not only the long duration of the stage but also the complication of the railway system. In addition, data will be added or occur during the stage of the railway project from construction, test and commission, operation, maintenance, and decommissioning [4]. Therefore, efficient data management is required to make sure that

the management will be done properly. In addition, a benefit gained from efficient data management is data can be utilized to improve railway maintenance [5].

From the long period, high costs of maintenance, and complication of data, an approach to managing all of these efficiently is required. The main problems of railway maintenance are the amount of data and lack of tools to utilize information including the efficiency of railway maintenance. Therefore, this thesis will explore the potential of varied tools to resolve the problems and summarize the results of the exploration of tools to resolve this problem.

1.2 Aim and objectives

Based on the background and research problems in the previous section, the main problems focused on in this thesis are the lack of tools to manage occurring data during the railway maintenance stage efficiently and to utilize the data. From the problems, the integration between building information modelling (BIM) and machine learning can resolve the problems. BIM will be used as a tool to develop a platform for managing data and information. It is worth noting that data in this thesis refers to raw data which are unprocessed while information refers to processed data and ready to be used to deploy for specific purposes. BIM can be the data management platform that allows projects to create data such as design during the design and construction stages, store data from different stages of projects, exchange data between stages, or import and export data from BIM models [6]. It can be seen that the use of BIM can significantly solve problems with the amount of data because data and information can be stored in a BIM model seamlessly. When the enormous amount of data can be solved, the next issue is how to

utilize the data. Machine learning can be used to develop machine learning models to perform predictive maintenance which is the main goal of this thesis. Based on the nature of machine learning, it requires a large amount of data [7] that conforms to the nature of the problems in this thesis. Therefore, the integration between BIM and machine learning proposed in this thesis will definitely solve the problems in railway maintenance. In addition, results from machine learning can be used to support the decision-making of railway operators because they can make a decision based on data that is not biased. In addition, the railway system is complicated and railway maintenance is no exception. Decision-making based on experience or opinion may not be proper. Machine learning can reduce the negative effect of experience and opinion. If the data are comprehensive enough, the railway operators can even apply the concept of data-driven which is the ultimate goal of machine learning applications when every decision is completely made based on data. It is expected that the overall efficiency of railway maintenance will be improved when machine learning is applied.

To summarize, the thesis aims to develop an approach to integrate BIM and machine learning for railway infrastructure maintenance to solve problems with data and railway maintenance efficiency. From the aim, the objectives of this thesis can be described as follows:

1. To explore the different approaches to integrate BIM and machine learning for railway infrastructure maintenance.
2. To select software and tools to integrate BIM and machine learning for railway infrastructure maintenance.
3. To develop 6D BIM models that are appropriate for machine learning integration.

4. To define parameters and variables to be stored in BIM models and develop predictive models using machine learning techniques.
5. To discover insights from the data using machine learning techniques.
6. To evaluate the influences of different parameters and variables on railway infrastructure defects and deterioration using statistical and machine learning techniques.
7. To assess the performance of developed predictive models.

1.3 Scope of work

The scope of the work focuses on the railway maintenance stage and maintenance activities. The components to determine are railway infrastructure such as track structures, track geometry, and wheel-rail interface. The thesis will propose the development of an approach to integrate BIM and machine learning together to solve the research problem as mentioned. For the BIM models, the development aims for developing 6D BIM models to allow the BIM models to include comprehensive data not only data relevant to maintenance but also data occurring during construction such as design, schedule, and cost. For machine learning or artificial intelligence, the thesis will focus on supervised learning, unsupervised learning, and reinforcement learning. In other words, all main types of machine learning models will be used.

1.4 Novelties and contributions of work

In general, BIM is applied during the design and construction stages. Use of BIM during the maintenance stage is limited especially in the railway industry where BIM has not been widely used. In terms of machine learning, machine learning application in the railway system is also

limited. This thesis will explore the feasibility to apply machine learning for predictive maintenance in the railway system. In addition, various forms of data will be explored for the potential to develop machine learning models which have not been explored in previous studies. Specific novelties and contributions will be discussed in detail in Chapters 5-9.

The general contributions of this thesis can be defined as follows:

1. BIM and machine learning can be integrated which improves the overall efficiency of railway infrastructure maintenance.
2. The proposed 6D BIM model concept can be a data exchange platform in the railway industry.
3. Data and information stored in BIM models can be used to develop machine learning models.
4. Results from the developed machine learning model will be beneficial for railway predictive maintenance and make predictive maintenance feasible.
5. Significant parameters and variables related to railway infrastructure maintenance are known and beneficial for railway predictive maintenance.
6. Insights from railway infrastructure maintenance are discovered.
7. The machine learning models can be applied for railway infrastructure predictive maintenance.
8. Railway maintenance plans can be prepared using the developed approach.
9. The developed approach is beneficial for supporting the decision-making of railway operators or parties who are responsible for railway infrastructure maintenance.

1.5 Thesis structure

This thesis consists of ten chapters including the introduction and conclusion. Chapter 1 presents the introduction to provide some background about the content of this thesis. This chapter is separated into five sections which are background and research problems, aim and objectives, the scope of work, novelties and contributions of work, and thesis structure. Chapter 2 will present the literature review conducted by reviewing previous studies related to this thesis. Two main keywords are used to conduct the literature review which are machine learning and BIM. However, keywords are not limited to these two terms. Railway maintenance, different techniques of machine learning, or sustainability are also used to perform the literature review. At the end of the literature review, a summarized table will be provided to present the overview of research in this area. Research gaps are discussed including how this thesis can fulfill those research gaps. Chapter 3 presents the first step to develop the approach of BIM and machine learning integration which is data collection, preparation, and processing. This chapter will discuss methods used to make data ready for machine learning development including characteristics of data. Chapter 4 presents how to set up a computer, environment, and other related software for developing machine learning models. Machine learning models can be developed using different methods. This thesis will explore and review before summarizing the differences between each method including providing some detail about the method used in this thesis. Chapters 5-7 will discuss applications of machine learning and how this thesis applies them to railway infrastructure predictive maintenance. Chapters 5-7 will present the application and how to develop machine learning models using each machine learning category in each chapter.

In each chapter, further details will be provided. For supervised learning in Chapter 5, they can mainly be separated into two parts which are classification and regression. Classification is a problem when labels are classes or discrete while regression refers to a problem when the labels are continuous values or real numbers. Chapter 6 discusses unsupervised learning for railway maintenance. This chapter will explore the insights of data using unsupervised learning which other machine learning techniques or humans cannot recognize [8, 9]. In Chapter 7, reinforcement learning applications will be discussed. The aims of reinforcement learning application in this thesis are to apply reinforcement learning in improving railway predictive maintenance efficiency and reducing carbon emissions from railway maintenance activities. As a result, this chapter will demonstrate how to prepare a railway maintenance schedule using reinforcement learning and predictive maintenance concept. At this stage, machine learning models have been already developed and are ready for application. To fulfil the aim to integrate BIM with machine learning in this thesis, Chapter 8 will present how to create BIM models to achieve 6D BIM models and the developed BIM models have the capability to integrate data and information to allow the integration with machine learning. Then, Chapter 8 will also discuss how to integrate BIM with machine learning using different tools to make the integration seamless. The last part of Chapter 8 will present how to develop Graphical User Interface (GUI) to make the application easier for people who do not have a background in coding and allow them to deploy the developed approach with ease. Chapter 9 will explore and demonstrate the deployment of the developed approach using a different platform. A summarized table is prepared to demonstrate the advantages and disadvantages of each platform to make sure that people who want to apply the developed approach will be able to choose the platform that best suits them.

The last chapter is Chapter 10 which is the conclusion and recommendations. This chapter will discuss the overall findings in the thesis and recommendations for further study which can be improved based on this thesis. The overview structure of this thesis can be shown in Figure 1-1.

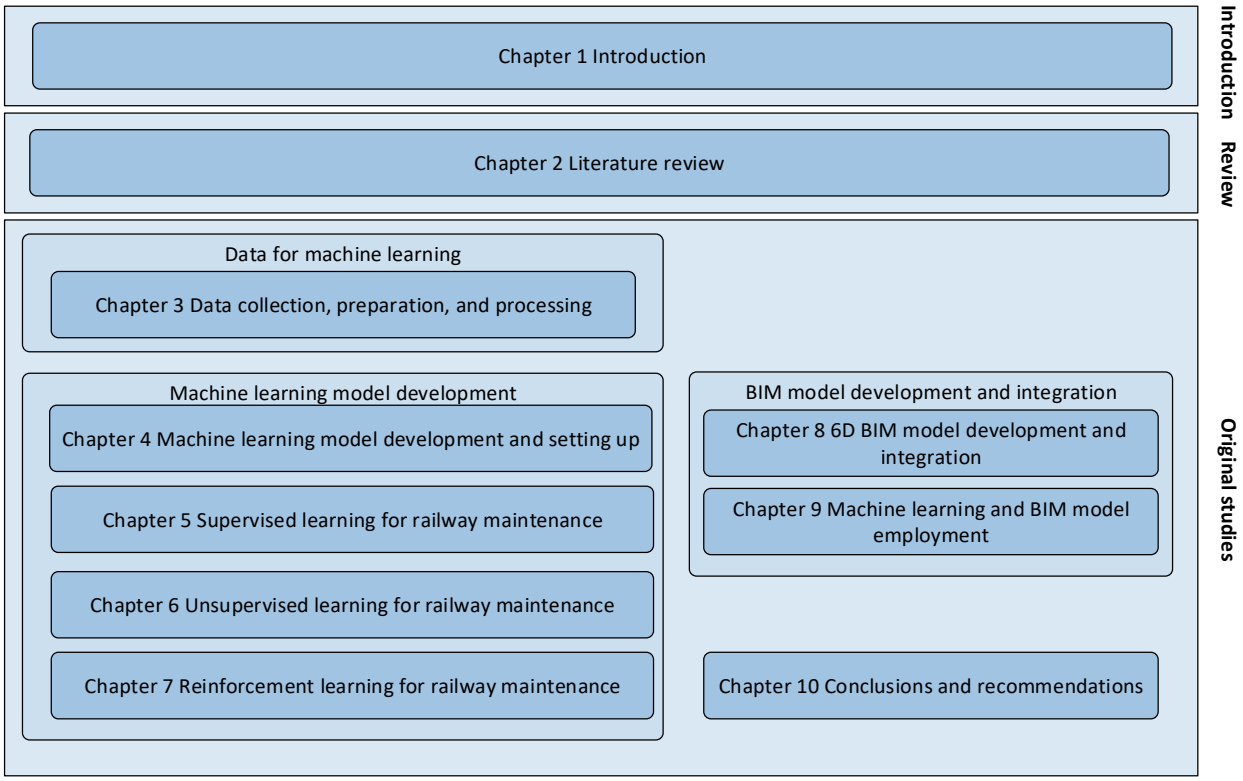


Figure 1-1 Thesis structure

2 Literature review

2.1 Machine learning in the railway maintenance

For railway maintenance, the first step to schedule maintenance efficiently is defect inspection and detection. Railway inspection can be done using different approaches such as site inspection, acoustic emission [10], laser [11], electromagnetic tomography [12], machine vision [13], or image processing [14]. When defects are detected, railway operators will prepare maintenance schedules to deal with those defects in terms of corrective maintenance. For preventative maintenance, the mentioned approaches can be used to monitor and evaluate the conditions of tracks. When the size or severity of defects reaches the trigger value, the railway operators can plan the maintenance schedule to keep railway infrastructure in a standard condition. As mentioned, predictive maintenance attracts more attention nowadays. To be able to conduct predictive maintenance, a capable tool is required otherwise it cannot be done. With the increasing computational power, machine learning is one of most powerful tools to detect and predict defect in railway infrastructure as well as the preparation of the maintenance schedule. This chapter will present studies related to the application of machine learning for railway maintenance.

Machine learning has been widely deployed in other industries such as Architecture, Engineering, and Construction (AEC) Industries. However, its application in the railway system is relatively low compared to other industries. In addition, the application of machine learning for railway maintenance especially railway predictive maintenance is even more limited. However, many

scholars have paid more attention to its potential and tried to apply its concept to the railway industry. Some relevant studies can be discussed as follows.

Guler [15] developed a predictive model using an artificial neural network (ANN) for predicting railway track geometry deterioration. They applied machine learning models to predict vertical settlements, wear and contact fatigue, general deterioration, and change in deterioration when there were significant changes in traffic conditions. In that study, a vertical settlement was focused on. The findings of that study were that the deterioration was too complex to predict due to the uncertainty of various parameters such as dynamic train loads, substructure, material, and environmental conditions, affecting the deterioration mechanism. Although there was no clear relationship found in that study, the study could draw some conclusions. It said the deterioration rate was generally a function of time and cumulative train loads and ANN had the potential to predict the track geometry deterioration. in terms of limitation, that study focused on ballastless tracks and the maintenance requirement was relatively low compared to ballast tracks. Therefore, the application of the developed model proposed in that study is limited. Many maintenance activities were ignored because the ballastless track does not require some maintenance activities such as tamping, ballast cleaning, and track lining.

Letot et al. [16] applied Monte Carlo simulation to reduce the maintenance cost with probabilistic models. They used this concept to prepare maintenance schedules when the developed model was based on track conditions. They found that the outcome was satisfying because the overall maintenance cost could be reduced when compared to the maintenance cost from the preventive maintenance or regular maintenance that was used as the benchmark. In that study,

a maintenance activity that they focused on was tamping. That study demonstrated the potential of the data-driven approach in improving maintenance efficiency. The main limitation of that study is only one maintenance activity was considered. Therefore, the complexity of railway maintenance has not been explored well because different maintenance activities have relationships between them that results in the improvement or deterioration of the railway infrastructure.

Lee et al. [17] combined ANN and support vector regression (SVR) to predict the deterioration rate of railway tracks. The data used in that study were historical maintenance data and track measurement data. The ballasted track was used as a case study in that study when the railway system was a high speed rail (HSR). Parameters used to develop the predictive model were types of substructures, track curvature, and velocity properties, number of tamping after ballast renewal, million gross tonnages (MGT), and initial track quality index (TQI) values. They found that ANN performs better than SVR. In that study, they assumed that the HSR structure was new and the weather condition was not related to the deterioration of the track structure and TQI which might not be right in many situations.

Sharma et al. [18] tried to optimize railway maintenance by developing a data-driven condition-based policy for the inspection and maintenance of track geometries. The size of the track section used in that study was one foot. They combined the concept of preventive and corrective maintenance and used the data from the maintenance as features or inputs. The machine learning techniques they used were random forests (RF) and Markov chain Monte Carlo (MCMC). They found that they could reduce the maintenance cost by up to 10% compared to the reference

data when applying machine learning models. However, their study has a limitation of limited maintenance activities which they considered each activity independently. Combinations of different maintenance activities have not been considered so there will be only maintenance activities would be performed in a section based on their developed approach. It can be seen that many studies had to simplify railway problems because of the complexity of the system.

Gibała and Konieczny [19] also applied ANN to develop a predictive model for the durability of railway switches. The developed model would predict the requirement of the reparation of the switches. Seven maintenance activities were concerned in that study which were half-switches exchanging, screw tightening, crushed stone thickening, the width of basic trails regulating, the width of turning trails regulating, sleeper exchanging, and crossing nose exchanging. In that study, features that they used to develop the machine learning model were service speed in the basic trail, service speed in the turning trail, type of fixation, time from the first measurement, bend of railway switch, and radius of turning trail. They found that the ANN model had the potential to be used to diagnose and prepare maintenance schedules for switches. However, the main limitation of that study is they had a very limited number of samples. The total number of samples was 15 only which might not be sufficient for machine learning because it requires a big enough size of the sample to train the machine learning model efficiently.

Bukhsh et al. [20] utilized tree-based classification techniques which were decision tree (DT), RF, and gradient-boosted tree (GBT) to support the decision-making of railway maintenance. The developed models had the capability to determine the need for maintenance, maintenance activities, and trigger conditions for maintenance. The scope of their study was switches and

crossings. Features used in that study were characteristic of switches and crossings components. They found that the GBT model had the highest performance in identifying the maintenance need while the RF model performed the best in identifying the maintenance activities required and the trigger's status prediction. Different machine learning techniques are suitable for different problems. Therefore, it cannot be concluded which machine learning technique is the best without testing. Because they applied only tree-based machine learning techniques in their study, there are many other machine learning techniques that might deliver better performance. Therefore, other techniques should be explored for the potential as well.

Gerum et al. [3] developed a predictive model using discounted Markov decision process model. They tried to optimize the inspection and maintenance schedules. Features were rail and geometry defects, time and location of the defect, type and severity of the defect found, type of inspection, the segment inspected, and time of inspection. They found that they could improve the overall efficiency of the inspection and maintenance by using the developed model. However, they still had limitations in the technique they used because it was not completely clarified as a machine learning technique which might have a better performance than the technique, they used to develop the predictive model.

Gao et al. [21] used ANN and genetic algorithms (GA) to develop machine learning models for detecting wheelflats. Features in that study were noise. From the model development, they could detect wheelflats 10% better than the use of the conventional method that they were using in the industry.

Krummenacher et al. [22] developed a machine learning model by using a support vector machine (SVM) and convolutional neural network (CNN) to detect flat spots, shelling, and non-roundness. They used vertical force between wheels and rails as features of the machine learning

Mohammadi et al. [23] did research on the relationship between track geometry and the occurrence of rail defects. They applied machine learning techniques which were extreme gradient boosting (XGBoost), SVM, RF, and logistic regression to detect defects. In their study, the problem was binary classification which means there were two classes for the prediction, with or without defect. From their study, XGBoost performed the best and the accuracy was 80%. The reason for the high accuracy was there were only two classes for the model to classify. However, the defect severity classification will be more challenging because the problem is more complicated.

Although machine learning can be used for railway maintenance as demonstrated in previous studies, some researchers found that there are some issues when the complexity of problems is increased. Sysyn et al. [24] found that they had a long processing time when the data were complicated or the size of features was big. In their study, they applied computer vision to detect contact fatigue at crossings. In the study, they suggested that the use of deep learning might be able to solve the long processing time issue.

Franca and Vassallo [25] used computer vision to detect and assess sleeper defects. They used both wooden and steel sleepers as samples to train the machine learning model. data were collected using cameras. The accuracies in their study were up to 93%.

Yella et al. [26] tried to assess the sleepers' conditions using SVM. Wooden sleepers were samples in the study. From their studies, manual inspection was normally used to inspect the condition of sleepers in many countries. However, this method was slow, expensive, and human power-consuming. Therefore, they wanted to develop an approach to improve the efficiency of sleeper inspection. They developed a machine learning model that could classify the condition of sleepers into two classes, good or bad conditions. The development tools were MATLAB and LNKNET. They claimed the developed approach could achieve an accuracy of 86%. Other researchers also aimed to detect sleeper conditions using machine learning and computer vision such as Zheng et al. [27] who applied YOLO V3 and Wang et al. [28] who applied CNN for detecting sleeper cracks. Both of them stated that the accuracies were more than 90%. This finding was also confirmed by other researchers such as Sysyn et al. [29] who applied SVM and k-nearest neighbour classifier (KNN) to detect sleeper defects even though the highest accuracy in that study was about 60% only. However, they stated that the use of more complicated and advanced machine learning techniques might result in better performance.

Lee et al. [30] tried to predict using SVM. They were interested in two track geometry parameters which were longitudinal level and alignment. The R^2 was 0.84. However, it can be seen that the contribution of the study is limited because only two track geometry parameters were studied. Hu and Liu [31] applied the same technique to predict track degradation. In that study, they were interested in longitudinal level, superelevation, and dip. Features were traffic volume, track class, and time interval. The R^2 was 0.81 which most previous studies could achieve.

Sysyn et al. [32] used axle box acceleration (ABA) as a feature for a machine learning model for detecting turnout defects. The problem in their study was binary-class classification. The accuracy of their study was 90%.

Lopes Gerum et al. [33] applied RF and recurrent neural network (RNN) to detect defects. The best outcome was from the RNN model that the accuracy was 80%. However, the number of classes was limited to two classes which were minor and major defects.

Consilvio et al. [34] applied SVM and k-mean for the maintenance of earthwork and track circuits. They also found that machine learning could save maintenance costs and improve railway maintenance efficiency.

Mohammadi and He [35] applied Deep Q-learning (DQN) to prepare maintenance schedules. They applied TQI and hazard index as indicators. The scope of their study was focused on five maintenance activities only which were preventive tamping and grinding, condition-based tamping and grinding, and renewal. They claimed that the TQI and hazard index were lower which means that the track was in better condition.

The above are some examples of relevant studies. However, the comprehensiveness of the research has not been high because of the limited components of the railway infrastructure or maintenance activities are concerned. There are research gaps for machine learning applications in predictive railway infrastructure maintenance.

2.2 BIM in the railway industry and maintenance

BIM is a platform that is used to create virtual models as a representative of a physical object. A main characteristic of BIM is the developed BIM model can contain information. In terms of the project, the BIM model can contain information on the project from the beginning of the project until the end of the project [36]. Different parties can utilize BIM models along different stages of the project such as design, construction, operation, maintenance, and decommissioning. BIM models can promote the visualization of projects significantly because apart from demonstrating models in 3D, BIM can be used to include information from different parties in a single model. For example, structural design can be integrated with system design. This will allow crash or conflict detection can be done to see whether the design is free from conflict and can be constructed. This is a reason why BIM is mainly used and a requirement for the construction stage in many countries nowadays. There are other benefits of BIM. BIM can be used to support the integration and collaboration between parties. Different parties can include their information on projects in a BIM model. At the same time, those parties can share the BIM model between Them or even add information responsible for their parties at the same time. BIM can be considered as level. According to the UK government's 2011 Construction Strategy, the BIM level is classified into four levels [37]. Level 0 means 2D computer-aided design (CAD) is used to represent the design and data sharing is done on a paper basis. Therefore, collaboration is limited. When the model is developed to be 3D, it can be classified as BIM level 1. At this level, information can be exchanged electronically. BIM level 2 is the most popular nowadays and the aim for many countries such as the UK that wants to achieve in 2016. At this level, the BIM model is 3D and

information are contained in the model. Therefore, related parties can integrate information of each party and share the model between them. The highest level of BIM is level 3 when a single model is developed and shared in the project. The information exchange is completely conducted between parties. It can be stated that BIM level 3 is the ultimate goal for everyone to achieve.

Besides the level of BIM, BIM can be separated using dimensions. First, a BIM model is created as a 3D model. BIM can also be developed to be 4D, 5D, or 6D depending on the information contained in the BIM model. The increasing dimension represents the complexity and purpose of the information in the model. Although there is no standard and definition for each dimension of BIM, a popular system can be described. The fourth dimension (4D) is the time or schedule when the project schedule is included in the BIM model. The fifth dimension (5D) represents cost and quantity when the quantity take-off can be done using the BIM model. Then, the cost is calculated based on the resulting quantity take-off. In general, the fourth and fifth dimensions are defined as mentioned. The sixth dimension (6D) is more flexible. Some researchers include information related to sustainability in the BIM model. Examples of information contained in the sixth dimension are energy consumption or carbon emissions [38]. The maintenance aspect can be included in the BIM model in the form of the sixth dimension instead of sustainability or the additional seventh dimension.

Although BIM is popularly used in the AEC industry, the railway system tends to apply BIM more in the present due to the government's requirements. BIM can also be integrated with Geographic Information Systems (GIS) for the benefit of location identification. Location integration is beneficial for project planning and decision-making [39]. At present, BIM is more

popularly used in the railway industry [40]. The railway system can gain the advantages from BIM application as other industries. From the previous studies, in general, BIM can reduce construction costs by up to 10% because BIM can prevent crashes and conflicts. Cost estimation's accuracy can be improved by 3% using BIM and the time consumption for cost estimation can be reduced by 80% because cost estimation can be done almost automatically. Finally, BIM can reduce construction time by 7% [36].

Bensalah et al. [41] applied BIM for a railway project in Morocco. They found that the application of BIM had many advantages which were confirmed by many studies. Shin et al. [42] studied the benefit of BIM application in the railway industry. They reviewed the data from seven railway projects in South Korea. From the analysis, they found that BIM could prevent 12 errors and mistakes in the projects. The errors cost more than $50,000. BIM can be also used for sustainability in the railway industry such as a study done by Kaewunruen and Xu [43]. In that study, they developed a 6D BIM model to calculate carbon emissions from the renovation and reconstruction. Then, the optimized option could be selected using the results from the BIM models. Kaewunruen and Lian [44] developed a BIM model for the life cycle management of turnouts in the subway system. They used the BIM model for auditing the condition of turnout. Kaewunruen et al. [45] performed a vulnerability analysis of the subway station using BIM and digital twins.

In terms of railway maintenance, Société Nationale des Chemins de Fer Français (SNCF) was the first railway authority to apply BIM in railway maintenance. They created BIM models from existing drawings and the current designs using reverse engineering. They also apply predictive

maintenance based on the developed BIM models. Their first project was established in 2015 [46].

Aziz et al. [47] reviewed the opportunities and value of using BIM with big data for the asset management of highway projects. They found that information taking place at the beginning of the project during the design and construction stages was valuable for the maintenance stage. In addition, data during the operation and maintenance stages could be collected by sensing for the benefits of asset management. However, the amount of information was huge from different sources of information such as sensors, mobiles, assets, and Open Data. Therefore, big data and a platform used for integrating this huge information were required.

Davtalab [48] studied the benefits of real-time data-driven BIM models for facility management in terms of time-saving during the operation and maintenance stages. The study found that 6D BIM could improve the timesaving of facility management during the operation and maintenance stages and the benefit of timesaving was up to 80%. This was because, for a building, there were many related systems. Monitoring these different systems was time-consuming and costly. Therefore, integrating a sensing system with BIM would be beneficial for this issue.

GhaffarianHoseini et al. [49] applied nD-BIM for sustainability purposes. They reviewed studies related to building energy efficiency and found that the application of BIM could reduce energy consumption during the post-construction stage.

Peng et al. [50] combined data mining and BIM together to identify problems of improper records during the operation and maintenance stages of the airport terminal project. Cluster analysis was

used to detect the problems by classifying the outliner of data. As a result, the proposed approach generated helpful suggestions and warnings about the operation and maintenance for supporting decision making which would be beneficial for prediction, warning, and decision-making that improved asset management.

Liu et al. [51] applied digital twins with the Internet of Things (IoT) for predictive maintenance in the aerospace industry. The benefits of this approach were data acquisition and information processing. They identified that the data taking place during the operation and maintenance stages was huge and needed a platform to be used to manage this huge information.

McArthur et al. [52] integrated ML with BIM to improve the quality of information collection. They said work orders (WO) were unstructured information. Therefore, it was difficult to be classified and managed efficiently. The developed automatic system from the mentioned approach would support the facility management team to deal with these WOs better.

Chen et al. [53] proposed the integration of BIM and IoT to develop an approach to support the decision-making of facility managers to do predictive maintenance. Data was collected by IoT sensors and 2 techniques of ML, SVM, and ANN were used to predict the conditions of equipment. They concluded that BIM and IoT integration could support predictive maintenance and the feasibility of a dynamic maintenance strategy of facility management.

From the above literature review, the application of BIM for railway maintenance has not been widespread and the integration between BIM and machine learning is still limited. In general, BIM is used for the design and construction stages only. After that, the benefits of BIM during

the operation and maintenance stages are limited because it is not used at its' full potential. In addition, many studies indicated that it is feasible to apply BIM for railway maintenance, and the potential can be further explored, improved, and investigated.

2.3 Research gaps

From the previous sections, the integration of machine learning and BIM in railway maintenance still has a research gap because BIM has not been widely used for railway maintenance. At the same time, although there are many studies using machine learning in the railway industry, comprehensive studies of railway infrastructure and railway maintenance have not been done from many perspectives such as the use of unsupervised learning and BIM integration with machine learning, the relationship between railway defect and track geometry parameters, or the application of deep or advanced machine learning for railway infrastructure predictive maintenance. Therefore, this thesis will try to fill some gaps in this area.

Besides the mentioned studies in the literature review, other relevant studies of machine learning and BIM in other industries and applications can be summarized in Table 2-1.

Table 2-1 Summarized related studies

Title	Methodology	Purpose	System	References
Data-driven predictive maintenance scheduling policies for railways	Data-driven	Maintenance scheduling	Railway	[3]
Predictive maintenance using tree-based classification techniques A case of railway switches	Tree classification	Maintenance scheduling	Switch	[20]

Title	Methodology	Purpose	System	References
Application of Artificial Neural Networks to Predict Railway Switch Durability	AI-ANN	Switch durability prediction	Railway	[19]
Data-driven optimization of railway maintenance for track geometry	Optimization	Maintenance of track geometry	Railway	[18]
Prediction of Track Deterioration Using Maintenance Data and Machine Learning Schemes	Machine learning	Track deterioration	Railway	[17]
An adaptive opportunistic maintenance model based on railway track condition prediction	Probabilistic model	Maintenance scheduling	Railway	[16]
Prediction of railway track geometry deterioration using artificial neural networks A case study for Turkish state railways	AI-ANN	Track geometry deterioration	Railway	[15]
Wheel defect detection with machine learning	CNN	Wheel defect detection	Railway	[24]
Optimization of railway operations using neural networks	AI-ANN	Operation	Railway	[54]
Railway Passenger Traffic Volume Prediction Based On Neural Network	AI-ANN	Passenger volume prediction	Railway	[55]
Application of an artificial neural network on railway passenger flow prediction	AI-ANN	Passenger flow prediction	Railway	[56]
Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways	AI-ANN/Genetic algorithms	Scheduling	Railway	[57]
A Study on the Use of Machine Learning Methods for Incidence Prediction in High-Speed Train Tracks	Machine learning	Incident prediction	HSR	[58]
Prediction of Railway Foundation Settlement Based	AI-ANN	Foundation settlement	Railway	[59]

Title	Methodology	Purpose	System	References
on the BP Neural Network Model				
A study on structural deformation of platform screen door based on artificial neural network	AI-ANN	Structural deformation	PSD	[60]
An artificial neural network model as a preliminary track design tool	AI-ANN	Design	Track	[61]
Damage detection in railway bridges using Machine Learning application to a historic structure	Machine learning	Damage detection	Railway bridge	[62]
A fault detection strategy for wheel flat scars with wavelet neural networks and genetic algorithm	ANN and genetic algorithms	Wheelflat detection	Railway	[21]
Insulation defect localization through partial discharge measurements and numerical classification	SVM	Insulation defect localization	Electrode	[63]
Automatic localization of casting defects with convolutional neural networks	CNN	Casting defect detection	Manufacturing	[64]
Principal components analysis and track quality index: A machine learning approach	SVM	TQI prediction	Railway	[65]
Track geometry big data analysis: A machine learning approach	RF	Substructure condition evaluation	Railway	[66]
Prediction of railway track geometry defects: a case study	Machine learning	longitudinal level prediction	Railway	[67]
Discrete train speed profile optimization for urban rail transit: A data-driven model and integrated algorithms based on machine learning	RF and SVM	Speed profile control	Railway	[68]
Learning From Accidents: Machine Learning for Safety at Railway Stations	DT	Fatal analysis	Railway	[69]

Title	Methodology	Purpose	System	References
Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study	Machine learning	Squats and corrugation detection	Railway	[70]
Improvements in Axle Box Acceleration Measurements for the Detection of Light Squats in Railway Infrastructure	Machine learning	Light squat detection	Railway	[71]
BIM- and IoT-Based Data-Driven Decision Support System for Predictive Maintenance of Building Facilities	BIM/IoT	Predictive maintenance	Building	[53]
Machine learning and BIM visualization for maintenance issue classification and enhanced data collection	BIM/Machine learning	Maintenance	Facilities	[52]
The role of data fusion in predictive maintenance using digital twin	Digital twin	Predictive maintenance	Aerospace	[51]
A hybrid data mining approach on BIM-based building operation and maintenance	BIM/Data mining	O&M	Building	[50]
Application of nD BIM Integrated Knowledge-based Building Management System (BIM-IKBMS) for inspecting post-construction energy efficiency	nD BIM	Energy efficiency	Building	[49]
Benefits of real-time data driven BIM for FM departments in operations control and maintenance	BIM and Data-driven	O&M	Building	[48]
Leveraging BIM and Big Data to deliver well maintained highways	BIM/Big data	Maintenance	Highway	[47]
Exploitation of BIM in Planning and Controlling the Construction Phase On-site	BIM 6D	Carbon emission (construction)	Building	[72]

Title	Methodology	Purpose	System	References
Carbon Emissions A 6D BIM Case Study				
Designing a unique Digital Twin for linear infrastructures life cycle management	Digital Twin	Life cycle management	Linear infrastructure	[73]
Digital twin aided sustainability-based lifecycle management for railway turnout systems	Digital Twin	Sustainability management	Turnout	[44]
Design and development of BIM models to support operations and maintenance	BIM	O&M	Asset management	[74]
Integrating BIM and GIS in Railway Projects	BIM/GIS	Integration	Railway	[39]
A BIM-based PSS Approach for the Management of Maintenance Operations of Building Equipment	BIM	O&M	Building equipment	[75]
Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model	BIM	Maintenance	Bridge	[76]
Digital Twin for Sustainability Evaluation of Railway Station Buildings	Digital Twin	Sustainability management	Station	[43]
Benefit–Cost Analysis of Building Information Modeling (BIM) in a Railway Site	BIM	B/C analysis	Railway	[46]
BIM Integration to Railway Projects - Case Study	BIM	Integration	Railway	[41]
Integrating BIM in Railway Projects Review and Perspectives for Morocco and Mena	BIM/GIS	Integration	Railway	[40]
BIM authoring for an image-based bridge maintenance system of existing cable-supported bridges	BIM	Maintenance	bridge	[77]
BIM-based framework for automatic scheduling of	BIM	Maintenance scheduling	Facility	[78]

Title	Methodology	Purpose	System	References
facility maintenance work orders				
Building information modelling to support maintenance management of healthcare built assets	BIM	Maintenance	Healthcare building	[79]
THE BIM-FOR-RAIL OPPORTUNITY	BIM	Integration	Railway	[80]
A case study towards transferring relevant safety information for facilities maintenance using BIM	BIM	Safety information for maintenance	Facilities	[81]
A Study on BIM Application for the Efficient Maintenance Management of Bridge Structures	BIM	Maintenance	Bridge	[82]
A CMMS Expert using BIM for IBS Building Maintenance	BIM	Maintenance	Building	[83]
BIM and QR Code for Operation and Maintenance	BIM/QR code	O&M	Building	[84]
BIM for building refurbishment and maintenance: current status and research directions	BIM	Refurbishment and maintenance	Building	[85]
BIM performance framework for the maintenance and refurbishment of housing stock	BIM	Refurbishment and maintenance	House	[86]
BIM: An integrated model for planned and preventive maintenance of architectural heritage	BIM	Plan and preventive maintenance	Heritage	[87]
BIM for sustainable whole-of-life transport infrastructure asset management	BIM	LCA	Infrastructure	[88]
Open BIM-based Information Modeling of Railway Bridges and its Application Concept	BIM	Integration	Railway bridge	[89]
BIM in the operations stage: bottlenecks and implications for owners	BIM	Operation	Building	[90]

Title	Methodology	Purpose	System	References
A knowledge-based BIM system for building maintenance	BIM	Maintenance	Building	[91]
Building Information Modeling in Support of Sustainable Design and Construction	BIM	Sustainable design and construction	Building	[92]
Introduction of Construction management integrated system using BIM in the Honam High-speed railway lot No. 4-2	BIM	Integration	HSR	[93]
Sustainability-Based Lifecycle Management for Bridge Infrastructure Using 6D BIM	BIM	Carbon emission and energy consumption	Bridge	[38]
Digital twin aided sustainability and vulnerability audit for subway stations	Digital twin	Sustainability and vulnerability	subway	[45]

3 Data collection, preparation, and processing

This section will start by presenting how to collect data and process to acquisition of data. Then, the characteristic of each type of data will be presented. Data have to be prepared and processed to suit the machine learning model training. Different machine learning techniques require different forms of data. Therefore, data preparation and processing are important steps for machine learning model development. Sources of data are varied depending on machine learning requirements and problems to be solved. More detail is described in the following section and Chapters 5-7.

3.1 Field data

Field data are directly collected from measurement or observation in real situations and settings. For example, field data of railway surface defects are collected by operational staff who visualize the rail defect in the field or using track geometry cars (TGCs) that collect track geometry parameters. If the field data are collected and measured properly, field data are valuable and reliable because they are data from real situations. In this thesis, field data were collected by MRS Logística S.A. which is a freight rail company in Brazil. The duration of the available data is ranged between 2016-2019. The interesting section has a length of 30 km. Sources of field data are ranged from track geometry measurements collected using TGCs, track design demonstrating grade, alignment, and curve details, maintenance records, and inspection reports.

For data collected by TGCs, the track geometry parameters are measured every foot which can be considered that it is not detailed enough for some use cases. To handle missing data, some data processing might be required. Data processing to handle this issue can be data interpolation to estimate unavailable data or data combination with other sources such as defect inspection reports to obtain additional data that TGCs cannot collect. Collected track geometry parameters comprise 12 parameters namely superelevation, longitudinal level (10 m chord) of the right rail, longitudinal level (20 m chord) of the right rail, longitudinal level (10 m chord) of the left rail, longitudinal level (20 m chord) of the left rail, alignment (10 m chord) of the right rail, alignment (20 m chord) of the right rail, alignment (10 m chord) of the left rail, alignment (20 m chord) of the left rail, gauge (without load), gauge (with load), and twist (20 m chord). In addition, the location and date of measurement are also available.

Track design provides information on vertical alignments, horizontal alignments, vertical curves, horizontal curves, and grades based on the station or locations. Track design can be used to be a feature for supervised learning to develop predictive models. At the same time, they can be used as features for unsupervised learning to discover insights.

For defect inspection reports, different defects are inspected and recorded. There are more than 81 types of defects based on track components such as boltless joints, damaged frogs, broken rails, and broken welds.

The above are some examples of field data sources. However, they are not limited to data collected by railway operators. Some other sources can be included for more comprehensiveness

of data and further analysis. For example, the railway inspection and maintenance manuals will be integrated to analyze railway defects. Machine catalogues will be used to collect data about energy consumption, operation cost, fuel consumption, or carbon emissions.

3.2 Numerical data

Different from field data, numerical data are not data occurring in the field. The main characteristics of data are data in the form of numbers that can be measured or quantified. In addition, they can be calculated for further values. Numerical data can be generated using computers, models, or simulations to mimic real-world phenomena. The advantages of numerical data are they can be generated with various conditions. Therefore, numerical data will be rich in terms of data variation which is beneficial for machine learning that required data variation and comprehensiveness. In addition, simulations can be done under different scenarios to test the different hypotheses which may not be able to be done in a real situation because of many limitations such as time, costs, safety, and data availability. Moreover, numerical data have a benefit in complicated scenarios when those scenarios may not be feasible to be done in the real world. However, numerical data may have an incredibility issue if numerical data or their sources are not validated. Therefore, validation is a crucial step before any simulations or models can be used to do any analysis or generate numerical data. In this thesis, there are two main techniques used to generate numerical data. Detail will be presented including methods to validate their reliability.

3.2.1 Multibody simulation and validation

Multibody simulation (MBS) is a technique that uses computers to consider the dynamic behaviour and relationships between components in simulated models. Relationships between components take place because of the forces that the components act on each other components. MBS is widely used in many areas. The railway system is an area where MBS has been used to analyze dynamic behaviours such as the interaction between rolling stocks and rails or the interactions between components of rolling stocks. The behaviours of each body in MBS are defined by using equations based on Newton's laws of motion. Some advantages of MBS are simulations can simplify the complex system while the accuracy is satisfying and MBS is flexible that can be modeled for different kinds of systems.

There are different software that can be used to apply MBS such as ADAMS (Automated Dynamic Analysis of Mechanical Systems), SIMPACK, LMS Virtual.Lab Motion, RecurDyn, MSC Nastran Multi-Body Dynamics, and Visual-Multi-Body-Simulation (Visual-MBS). However, the MBS simulations used in this thesis are D-Track. D-Track was developed in 1994 [94] to study the dynamic behaviour of rolling stocks and tracks. Steffens [95] further developed the GUI for facilitating users and validated the accuracy of the developed MBS simulations. In addition, he included the Dynamic Analysis of Rail Track Structure (DARTS) to further develop the simulation. However, the accuracy issue had not been solved. In 2007, Leong [96] found the sources of the accuracy issue. He included the calculation method based on Australian Standard code AS1085.14 in the simulation and also include the field data for comparison and accuracy improvement. The field data he used were collected from the Melbourne to Geelong route in

Australia. He compared different parameters from the simulation with field data such as average rail acceleration, average wheel-rail contact force, shear force, and bending moment. In addition, he considered results from other simulations such as DARTS, NUCARS (New and Untried Car Analytic Regime Simulation), SUBTTI (Subgrade-Train-track Interaction), DIFF (Vehicle-Track Dynamic Analysis Model), and VIA (Vehicle Interaction with Track Analysis Model). He found that the accuracy from D-Track was satisfying and results were close to field data and other numerical simulations. Examples of error causes were miscalculation of wheel-rail force calculation, assumptions, sleeper pad reactions, and sleeper bending moment. After the improvement, he could fix the issues error issue and improve the accuracy of the simulations. From the improvement, the accuracy was more than 90% when compared to field data. Then, many studies were conducted to validate the accuracy of D-Track and they indicated that the results from D-Track could be used as benchmarks and representatives of field data [97]. One of the main advantages of D-Track is the simulation is flexible and easy to use. Different parameters can be adjusted with ease such as track profile that default profiles are available or can be imported via comma-separated values (CSV) files, track properties that can be adjusted to suit the investigated track, rolling stock properties to represent both passenger and freight rolling stocks, analysis method, or irregularities that both rail and wheel irregularities are available and can be used to represent defects. in addition, rail and substructure can be modified such as rail types, sleeper types and spacing, and track bed properties. Speeds and weights of rolling stock are also available.

Different outputs can be exported from D-Track depending on the requirement of the application. Examples of outputs are wheel-rail contact force, ABA, or sleeper displacement. Examples of output from D-Track are demonstrated in Figure 3-1 and Figure 3-2 respectively. The first one is an example of wheel-rail contact force. The second figure presents an example of ABAs. It is worth noting that those two figures present only one set of data from a wheel. In machine learning model training, two sets of data from two wheels can be used to train the models.

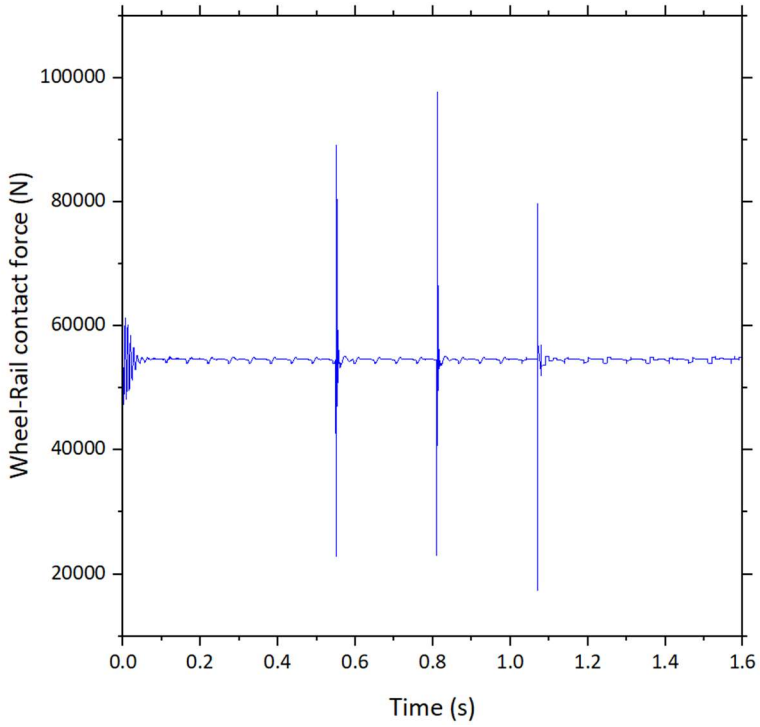


Figure 3-1 Example of wheel-rail contact force from D-Track

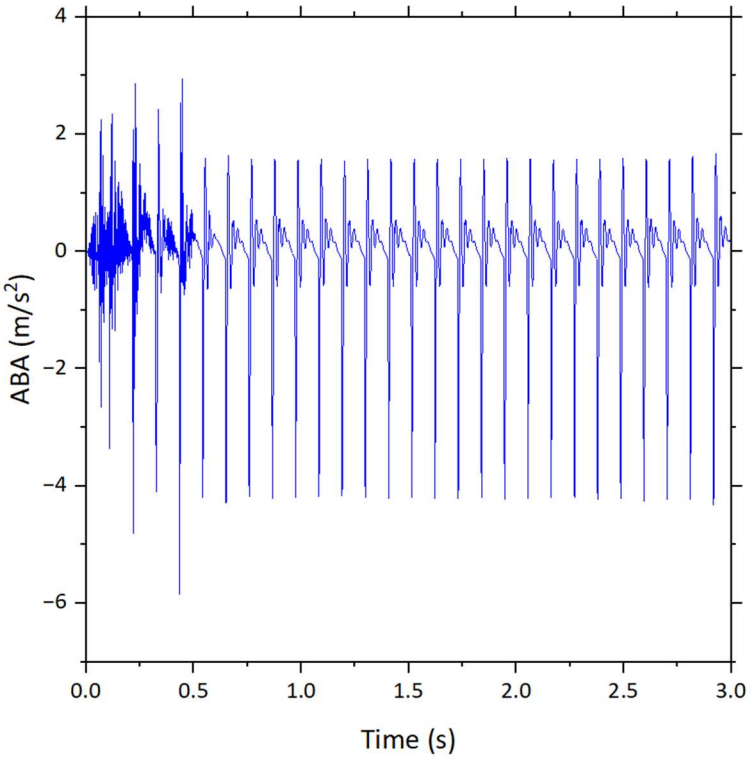


Figure 3-2 Example of ABA from D-Track

Other examples of inputs and outputs of D-Track are shown in Table 3-1.

Table 3-1 Examples of D-Track inputs

Input				Output
Track	**Vehicle**	**Irregularity**	**Variety**	
• Rail type • Gauge • Axial force • Pad stiffness • Pad damping • Sleeper type • Spacing • Track bed Stiffness • Track bed Damping	• Speed • Tare mass • Carry mass • Primary suspension stiffness • Primary suspension damping • Wheel dimension • Hertzian contact coefficient	• Corrugation • Dipped joint • Dipped weld • Peaked weld • Arbitrary profile • Wheel flat	• Speed • Irregularity length • Irregularity depth • Centre of irregularity • Rail analysis position • Sleeper analysis position	• Acceleration • Force and pressure • Moment and shear • Bending moment • Displacement

Irregularities or defects that are available in D-Track can be shown as equations in Figure 3-3. In addition, customed irregularities can also be inputted to D-Track in the form of CSV files. Therefore, many other irregularities and defects can be used for analysis.

Type	Equation	Shape
No Irregularity	Nil	
Corrugation	$f(x)=\frac{d}{2}\times sin\left(\frac{2\pi x}{L}\right)$	
Dipped Joint	$f(x)=d\times\left(1\pm cos\left(\frac{\pi x}{L}\right)\right)$	
Dipped Weld	$f(x)=\frac{d}{2}\times\left(1-cos\left(\frac{2\pi x}{L}\right)\right)$	
Peaked Weld	$f(x)=-\frac{d}{2}\times\left(1-cos\left(\frac{2\pi x}{L}\right)\right)$	
Arbitrary Rail Surface Profile	x and y coordinates must be defined	
Wheel Flat (worn)	$f(x)=\frac{d}{2}\times\left(1-cos\left(\frac{2\pi x}{L}\right)\right)$	
Arbitrary Wheel Surface Profile	x and y coordinates must be defined	

Figure 3-3 irregularities available in D-Track

3.2.2 Finite element method (FEM) model development and validation

The finite element method (FEM) is developed as a mathematical model. The purpose of the model is to mimic the system behaviour by discretizing a continuous system. For example, an object under compression is divided into smaller elements to perform analysis and study its behaviour. Partial differential equations are used to solve problems of system behaviour. Small elements interact with other elements via nodes. The reasons for dividing an object into smaller elements are the calculation can be done more efficiently and simply. Then, the analysis results of small elements will be combined to represent the result of the whole object.

FEM is widely used in many fields such as physics, engineering, systems, structures, fluids, or electromagnetic fields. Some advantages of FEM are it can improve the accuracy of calculation, it has high flexibility due to variety of the setting, it can save the cost because real physical objects do not need to be built, and some complicated simulations can be tested while they may not be able to be conducted in a real-world situation.

Examples of FEM software are ANSYS, Abaqus, COMSOL Multiphysics, SolidWorks Simulation, and MSC Nastran. In this thesis, LS-DYNA will be used to create FEM models to od analysis and generate numerical data in some cases when field data are not available or data diversity is required. An example of the FEM model used in this thesis is shown in Figure 3-4. From the figure, tracks are modelled as beams and supported by sleepers. The rolling stock is modelled with other related components such as the primary suspension system, secondary suspension system, and wheels. Outputs from FEM models can be varied depending on the application purposes. Examples of outputs are ABA, sleeper displacement, or wheel-rail contact force.

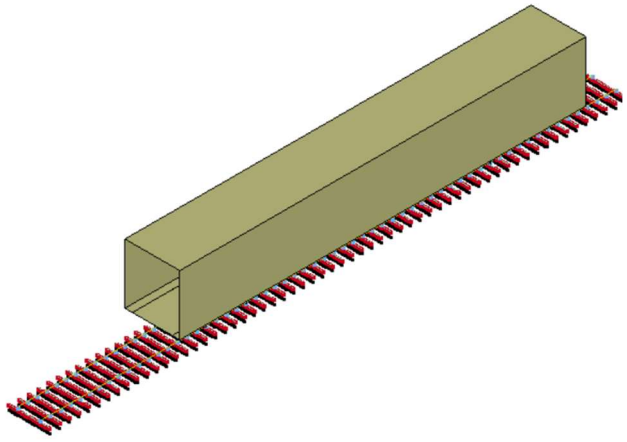


Figure 3-4 Example of FEM model

Because results from FEM are numerical data, model validation is required to ensure that created models are reliable, accurate, and can be used as representatives of the real-world system. In this thesis, model validation will be discussed in detail when numerical data are used and generated by simulations. Further information will be discussed in the machine learning model development chapters which are Chapters 5 to 7.

3.3 Data processing

Data preprocessing is a crucial process in machine learning model development. The process takes the majority of the time of machine learning model development. Time spent for data preparation and processing is estimated at 60-70% of the total time for machine learning model development [98]. In addition, this step might be the most difficult in machine learning model development because it requires a lot of processes and steps to prepare data and make data in the form that machine learning can learn. Obtained data are varied and different projects require different forms of data due to the uniqueness of each project. Sometimes, raw data can be used directly in machine learning models. For example, raw data which are in the form of text have to be processed to number. Data with errors or missing values have to be corrected. Some data require further calculation before being used in machine learning models. Last but not least, some data require feature engineering for selecting relevant features from raw data to train machine learning models. That is why data processing is required. Data processing may refer to data cleaning, data pre-processing, or data preparation.

Some steps may be standard and can be applied to different machine learning projects. However, there are no exactly defined steps for data processing. Some examples are shown as follows:

- Data cleaning: this step is used to remove errors in data.
- Outlier removal: in some cases, data might have an abnormal distribution. For example, data might have some samples with extremely high or low values. This might be caused by errors from data collection or data in extreme events that do not normally occur. Therefore, removing this kind of data might improve predictive performance. An example of data with outliers is shown in Figure 3-5. The figure demonstrates the distribution of superelevation which is one of the track geometry parameters. It can be seen that the majority of data is close to zero. When the values are more different from zero, the number of samples tends to decrease. However, there are some samples with superelevation higher than 80 mm. These samples can be considered outliers because the values are extremely high which not suppose to take place in a real situation. They might be errors from data collection and should be removed from the dataset to train machine learning models.
- Feature selection or feature engineering: this step is used to select relevant features as input for machine learning models. Relevant features are based on machine learning projects. Assumed irrelevant features will be removed in this step to make the training shorter due to fewer features and improve the model performance. Some machine learning models can cope with feature selection itself and feature selection does not need to be done manually such as CNN.

- Data transform: this step is used to reshape data. It is worth noting that the shape of data refers to data dimensions and sizes. For example, data in the form of images are 2D data but their sizes can be varied based on the pixels in both axes (horizontal and vertical axis). Most machine learning models require features or inputs to be consistent. That means the shape of the data has to be fixed. However, in reality, the shape of data can be varied. Therefore, data transform is used to reshape data to be able to be fed into machine learning models. Data transform can be used to make data shape bigger (e.g., padding) as shown in Figure 3-6, smaller (e.g., trimming or pooling) as shown in Figure 3-7 and Figure 3-8 respectively, or the same (e.g., distributing). Some machine learning techniques can handle different shapes of data such as fully convolutional networks (FCN) that can handle data with different shapes.

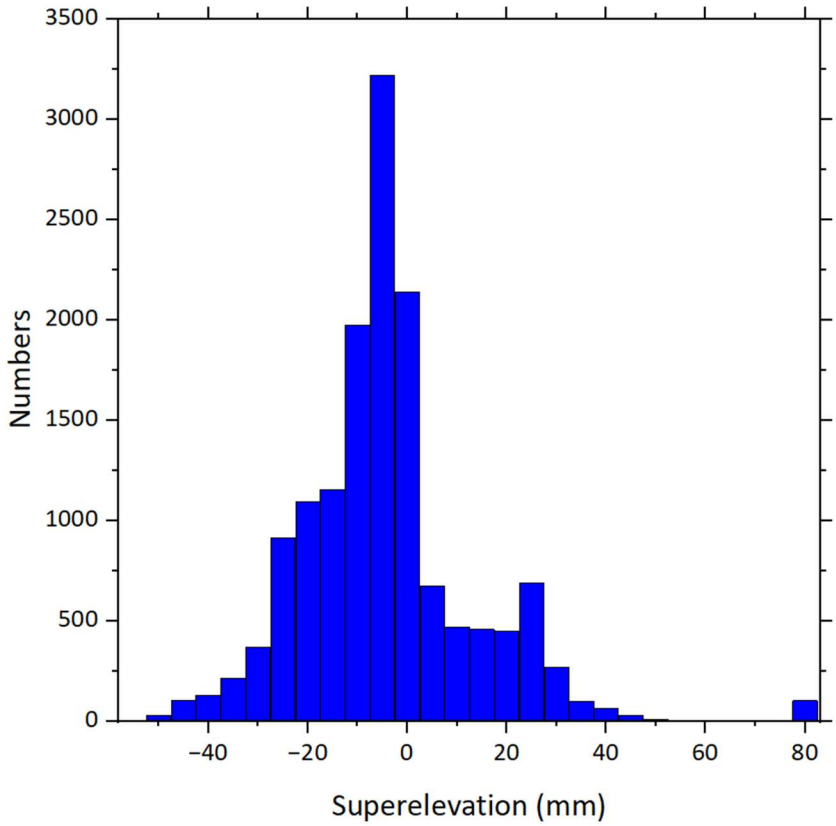


Figure 3-5 An example of data with outliers

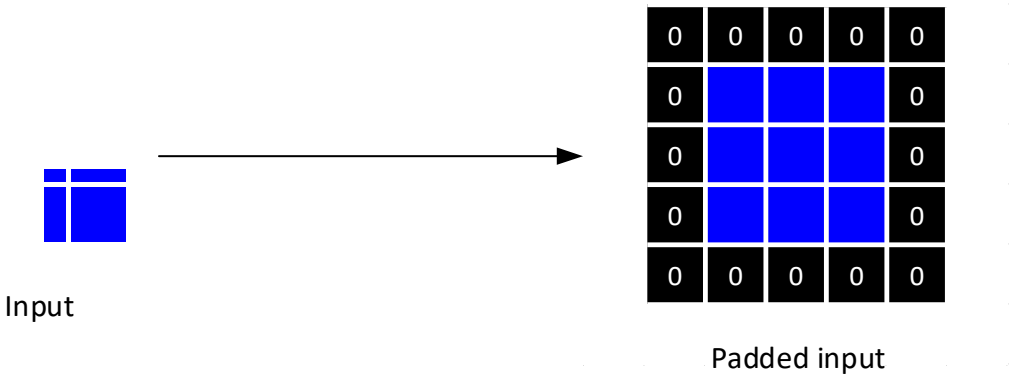


Figure 3-6 Example of data transform using padding

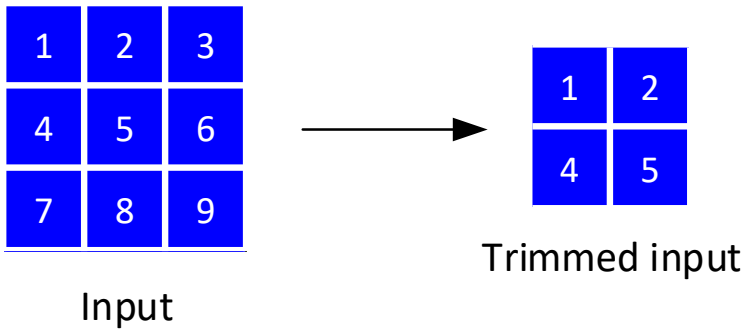


Figure 3-7 Example of data transform using trimming

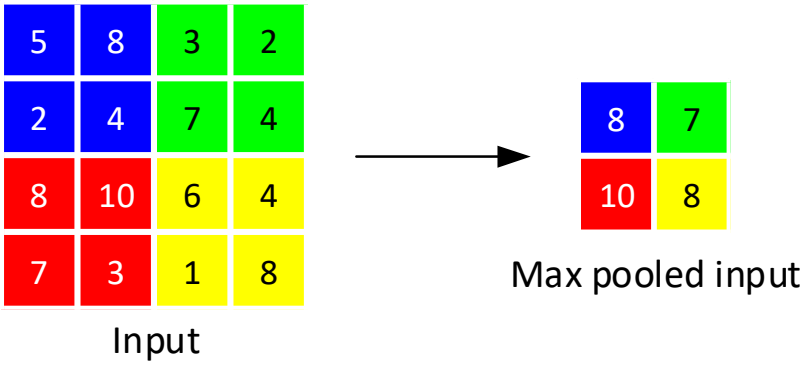


Figure 3-8 Example of data transform using max pooling

- Dimensionality adjustment: this step can be considered a part of data transform. The main difference between data transform and dimensionality adjustment is that dimensionality adjustment focuses on data reshaping which changes the number of data's dimensions. For example, the number of data dimensions can be increased by stacking more than one set of data together or using the padding technique to increase the number of dimensions as shown in Figure 3-9. Data dimension can be decreased by

trimming that removes some part of data completely to make data in the desired shape or flattening for a flatten layer that has a function to make multidimensional data to be 1D data while no data are removed as shown in Figure 3-10. Flattening is done by stacking data in dimensions apart from the first dimension to the first dimension. As a result, the number of data dimensions will be one but the size will be larger. Flattening is required when multidimensional data are fed into dense layers or fully connected layers to do classification or regression.

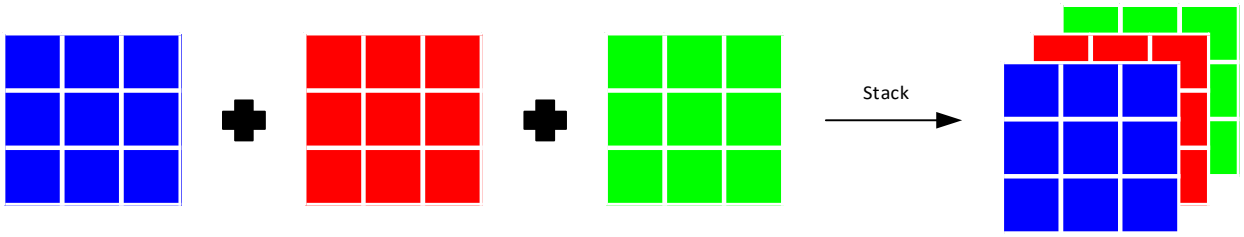


Figure 3-9 Example of data dimensionality adjustment using stacking

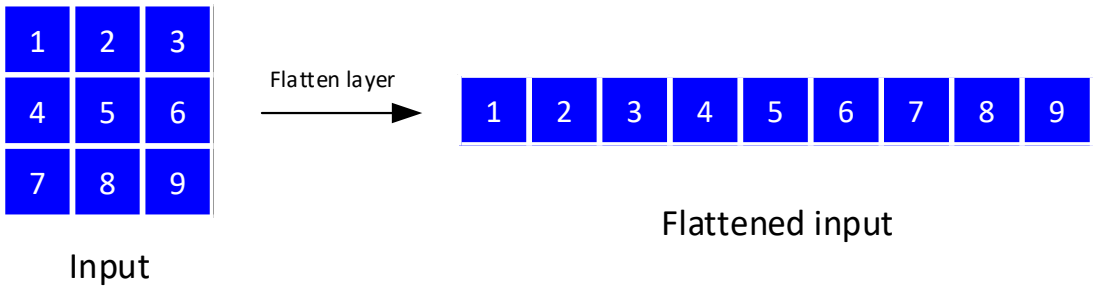


Figure 3-10 Example of data dimensionality adjustment using a flatten layer

3.3.1 Data frequency

Data frequency refers to how often the data are measured or collected over time. Data frequency can be varied based on the characteristics of the data or the purpose of the data application. In general, data frequency can be ranged from low frequency to high frequency depending on the mentioned reasons. For example, data of ABA may have a data frequency of 1,000 Hz because the purpose of data application is used to detect irregularities or detects on rails or rolling stocks. On the other hand, demographic data about a population may have a data frequency of one year or ten years because the population change is much less than the change in ABA data. This can indicate that data frequency depends on the requirements of use. As mentioned, data frequency can be varied depending on the application of data. High-frequency data is suitable for time-series analysis while low-frequency data is suitable for discrete analysis.

There is no fixed standard or requirement for data frequency to be used in different applications. Some applications may have a standard identifying the required frequency. For example, ISO 15186-3 recommends a data frequency between 50 Hz to 80 Hz [99]. It does not mean that the data with higher frequency have a better quality than the data with lower frequency. It is important to select the most suitable data frequency based on the purpose of the data use. Data with too high a frequency may have a too high cost to collect and more storage for data. In addition, higher frequency data require a longer time to process in the further stages. In the case of BIM, if data or information has to be stored in BIM models, the BIM models' size will be bigger when the data frequency is higher. Therefore, data frequency has to be carefully selected otherwise the BIM models will require a lot of resources for implementation. However, too low

data frequency may result in the loss of some parts of data which is undesirable in some situations. Therefore, data frequency has to be carefully selected to suit the application.

In the aspect of machine learning, data frequency can affect model performance. Machine learning models detect patterns in data. If data have too low data frequency, some patterns may be lost. On the other hand, the bigger size of data also requires more time for training machine learning models when the performance of the models may not be improved significantly. In general, machine learning models require quality data with a large amount. This is to ensure that machine learning models can detect hidden patterns and comprehensively cover every label or case that possibly occurs.

Examples of the difference in data frequency requirement in machine learning are natural language processing or speech recognition requires high-frequency data because the purpose of the training is to ensure that the machine communication is natural as human communication. However, image classification or image processing requires data with a lower frequency compared to natural language processing because the classification can be done although the data frequency is lower such as the number of pixels in images.

Moreover, the data frequency affects the temporal nature of the data which is important to machine learning applications. For example, for time-series analysis or prediction, high-frequency data are used to capture short-term trends, and hidden patterns in lower-frequency data are used to capture long-term trends. An example of this application is stock forecasting.

It can be seen that data frequency plays an important role in the quality and quantity of data for machine learning. In addition, many machine learning techniques require the consistency of features or input. However, in practice, features can have different shapes such as images with different sizes that have different numbers of pixels or ABA with different shapes because the data frequency may be fixed but the shapes of data are varied because the speeds of rolling stock are varied. For more detail, when a track section is considered, the length of the section is constant. Normally, the data frequency is also constant because it is based on sensors or equipment used to measure. If the rolling stock runs faster, the shape of the data will be smaller because the time required for the rolling stock to run on the track section is shorter so the shape of data will be smaller. The data shape can be remained the same by adjusting the data frequency. However, those approaches may not be efficient for the machine learning concept. Therefore, data processing can compensate for the effect of data frequency and differences in data shape.

In this thesis, there are two main techniques used to keep the data shape consistent when the data frequency is the same but the rolling stocks' speed is varied. They are fast Fourier transform (FFT) and padding which will be further described in the following sections.

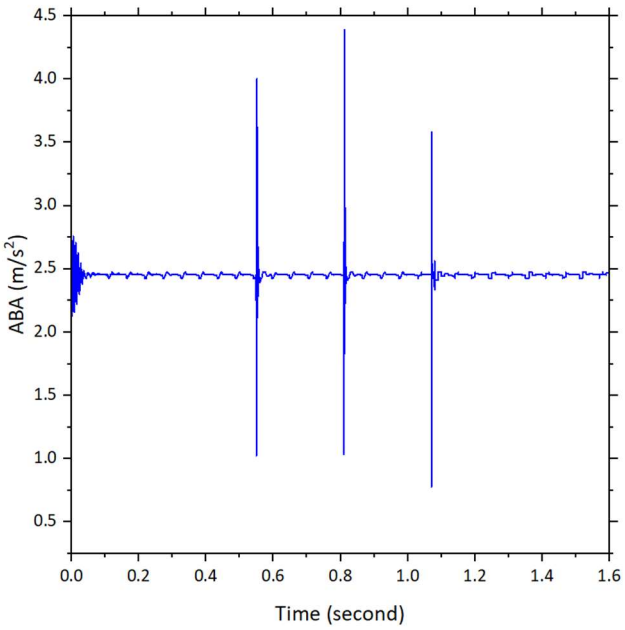
3.3.2 FFT

FFT or fast Fourier transform is a widely used processing the Discrete Fourier Transform (DFT) FFT is used to convert time-domain data into frequency-domain data. The most outstanding characteristic of FFT is FFT can be used to process data and provided processed data with the

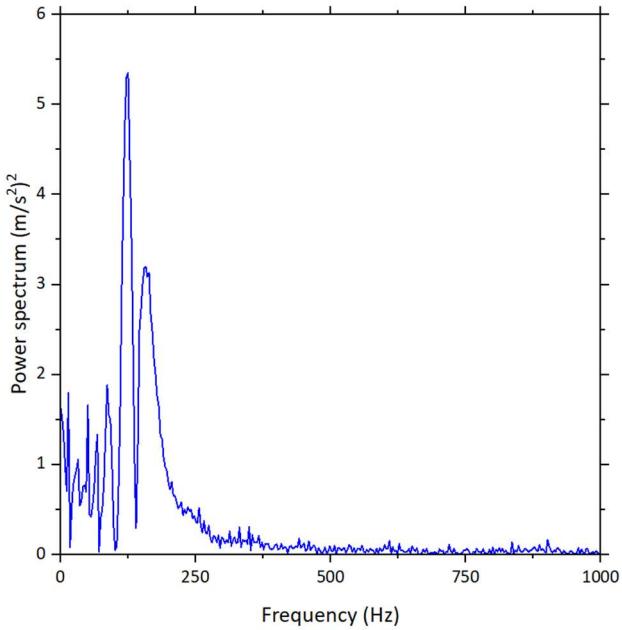
same shape for machine learning models. The shape of processed data can be bigger or smaller than the shape of the original data. In addition, it may be another aspect for machine learning to explore the pattern hidden in the data apart from data in the form of a time domain.

The application of FFT is not limited to signal processing only. FFT can be used to speed up the training process of machine learning training because data can be processed to have a smaller size. In addition, FFT is much faster than DFT when they are used to process data. That is why FFT is popular and widely used in different areas.

FFT can be implemented using Python libraries such as NumPy, SciPy, PyTorch, and TensorFlow or some software such as MATLAB, Mathematica, Octave, and GNU. An example of processed data using FFT is shown in Figure 3-11. FFT used in this thesis will be further described in the machine learning model development namely Chapters 5 to 7.



(a)



(b)

Figure 3-11 Example of processed data using FFT; (a) original data and (b) processed data using FFT

3.3.3 Padding

Data padding is a technique used in the machine learning field to deal with variable-shape features such as sequences, sentences, or images that have different shapes. Padding can be conducted by adding extra elements (normally zeros) to the end or start of smaller shape data so that all data will have the same shape.

As mentioned, many machine learning techniques require features to have a fixed shape or size. Therefore, padding is important because it is used to make the shape of the feature the same. Without padding, it would be difficult to feed variable-shape features into machine learning models if those machine learning algorithm does not support variable-shape features.

Normally, padding can be done in two ways which are pre-padding and post-padding. Pre-padding is done by adding zeros to the features. The number of zeros added to the features depends on the biggest and smallest shapes of the samples. For example, if the smallest of samples is 1*500 while the biggest shape of samples is 1*1,000, 500 zeros will be added to the end of the samples with the smallest shape. Samples with bigger shapes will require a smaller number of zeros to be added to make the processed features have 1*1,000 shapes. Post-padding is similar to pre-padding except zeros will be added to the end of features. It is worth noting that any number can be added to features not limited to zero only.

As mentioned, all features will have the same length after padding. Then, features can be organized into a matrix consisting of rows and columns. The resulting matrix can be fed into machine learning models that require features to have a fixed size before training. Padding can be implemented using different programming languages and machine learning libraries. Examples of Python libraries that can implement padding are NumPy, TensorFlow, PyTorch, Pandas, and SciPy. An example of processed data using the padding technique is shown in Figure 3-12.

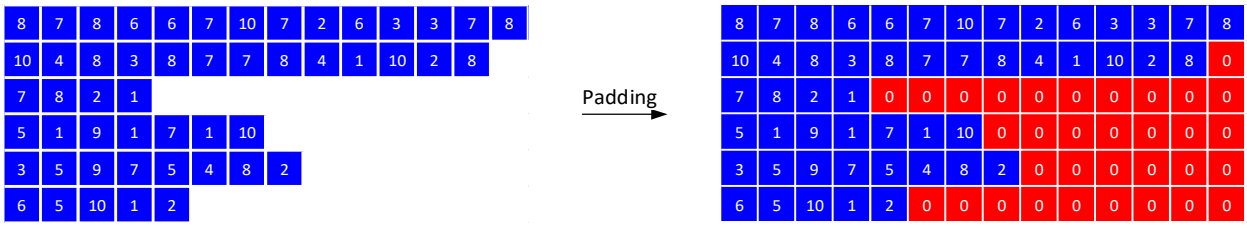


Figure 3-12 Example of data processing using padding

4 Machine learning and BIM applications

4.1 Machine learning

Machine learning attracts more attention nowadays. Machine learning can be considered a branch of AI. Machine learning focuses on developing computer algorithms to learn some things from data and improve performance over time when additional programming is not needed. Machine learning is aimed to create algorithms that can automatically detect patterns and do predictions based on the patterns.

Machine learning is widely used in various fields such as image processing, speech recognition, natural language processing, fraud detection, recommendation systems, and predictive maintenance which is the aim of this thesis. Machine learning can be mainly categorized into three types which are supervised learning, unsupervised learning, and reinforcement learning.

4.1.1 Types of machine learning and their characteristics

Supervised learning is used to generate the concept of function. Algorithms are trained to give the best estimation from input data. It is called “Supervised” because algorithms need labelled data for training. In this case, humans act as teachers to train the computer with features and predictions (labels). The machine will learn the pattern from it and try to predict the output. The main types of supervised learning problems are regression and classification. The common algorithms are Nearest Neighbor, Naive Bayes, DT, Linear Regression, SVM, CNN, Long Short-Term Memory (LSTM), RNN, Gated Recurrent Unit (GRU), Residual Network (ResNet), and ANN.

For unsupervised learning, the computer can train itself without help from humans. Moreover, the machine can also provide some new knowledge or insights for humans from what the machine discovers from the data. This will be beneficial for humans when insights are not known. It is used for pattern detection and descriptive modelling by mining for rules, detecting patterns, and summarizing and grouping data points. The main types of unsupervised learning problems are clustering and association rules. The common algorithms are k-means clustering and association rules. In conclusion, unsupervised learning does not need labelled data for training and desired labels are unknown.

Reinforcement learning aims to use observations to take actions that maximize the reward. Reinforcement learning will keep learning from the environment until it finds all possibilities to maximize performance. The reward will be set to learn to improve its performance. The decision-making function will be set for the machine to take action. Any actions will result in different rewards for the learning or reinforcement of the machine. There are many applications of reinforcement learning such as board games (Chess, Go) or self-driving cars. The common algorithms are Q-Learning, DQN, Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), Asynchronous Advantage Actor-Critic (A3C), Trust Region Policy Optimization (TRPO), Deep Deterministic Policy Gradient (DDPG), Temporal Difference (TD), Monte Carlo Tree Search (MCTS), and Deep Adversarial Networks.

Some machine learning architectures are shown in Figure 4-1 to Figure 4-4. Figure 4-1 presents the architecture of DNN when features are fed into the architecture through nodes in each layer. There are one input layer and one output layer. The layers between them are hidden layers that

the number can be adjusted to suit the problem. Weights are used to multiply features from the previous layer. Activations functions are also used to process features to mimic the nonlinearity behaviour. Figure 4-2 presents the architecture of CNN when features are not considered as an individual values but as a group of values. This concept is developed based on a human vision that multiple points of data are considered together to classify objects. In CNN models, feature complexity can be reduced using the pooing concept that also mimics human vision. The concept is humans can recognize objects although the resolution of the picture is reduced. The pooling concept is done by selecting a number or value to represent the group of values so the complexity of the feature is reduced. Figure 4-3 demonstrates the architecture of CNN when applied to time-series data. the concept is the same as image processing but the dimension of features is reduced from two (picture) to one (time-series data). Figure 4-4 demonstrates the LSTM architecture that might look like the DNN architecture. However, the main characteristic of LSTM is there is a loop weight training function used to consider the historical data inside the architecture. Therefore, it can be said that LSTM models have a memory which is good for time-series forecasting.

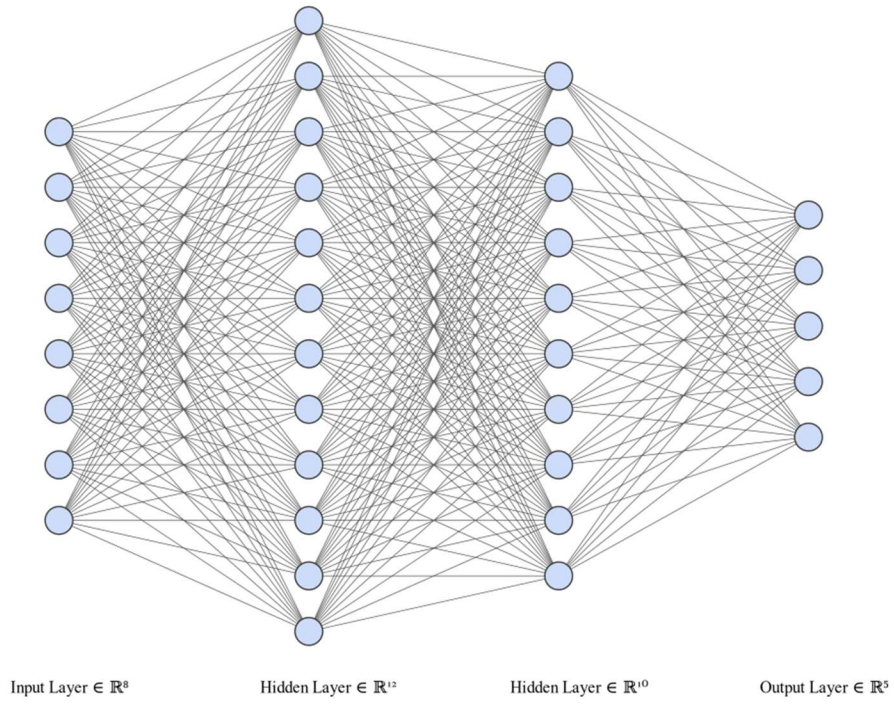


Figure 4-1 An example of DNN architecture

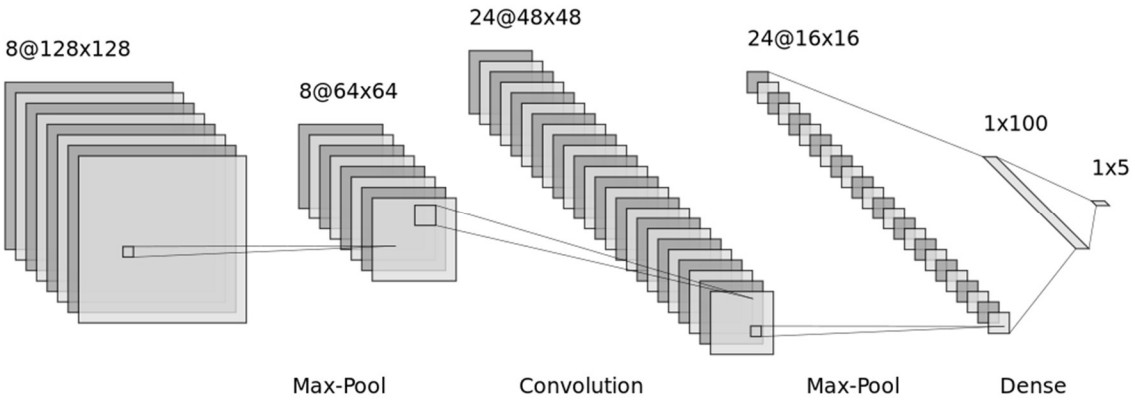


Figure 4-2 An example of CNN architecture for image processing

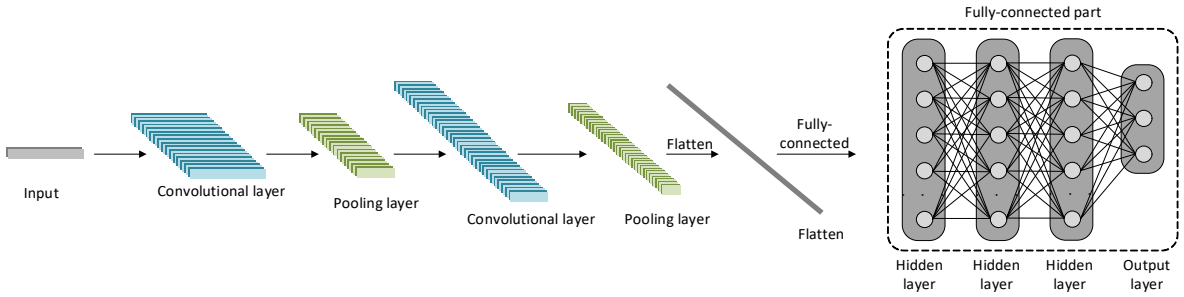


Figure 4-3 An example of CNN architecture for time-series data

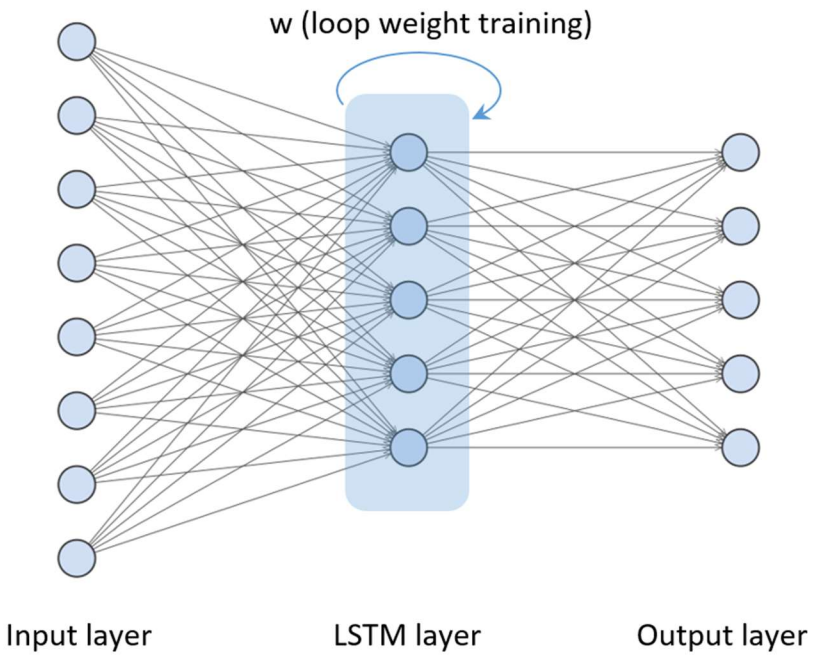


Figure 4-4 An example of LSTM architecture

From previous paragraphs, there are many types of AI that can be used for this thesis. In this thesis, all three main categories of machine learning will be deployed for different purposes of railway predictive maintenance. More details will be described in the following chapters.

4.1.2 A general framework of machine learning

The overall framework or process of machine learning starts with raw data obtained from data collection. The data is collected by different sensors and measurement approaches. Collected data will be processed based on machine learning algorithms. Each set of data is called a sample. Then, the data is normally split into two groups which are training data and testing data. Training data is used to train models and develop predictive models while testing data is used to evaluate the performance of developed models. It is worth noting that this applies to supervised learning only. These two groups of data will be utilized separately to make sure that there is no over-fitting issue or data leaking issue from using all data to develop models. The proportion of the splitting can be varied depending on many factors such as the sample size, model design, etc. The proportion might be 0.7, which means that 70% of the data is used to train models while another 30% of the data is used to test models. A technique used to split data is called sampling. Sampling can be done using different ways such as linear sampling, shuffled sampling, or stratified sampling.

Linear sampling splits samples with a specific number without changing the order of samples. For example, if the total number of samples is 500 and the proportion of the splitting is 0.7, the first 350 samples are used as the training data while another 150 samples are used as the testing data. Different from linear sampling, shuffled sampling determines the number of samples from the proportion of splitting. Then, certain numbers of samples are selected randomly by ignoring the order. Last, stratified sampling shares the same characteristic as shuffled sampling. However, this approach includes class distribution in the sampling process. The proportion of each category is

maintained the same in training and testing data. Therefore, the proportion of each label category will always remain the same. To demonstrate a clear view of this approach, a binomial classification of class A and class B is used to demonstrate. From the original data, if the percentage of data with classes A and B is 60:40, samples will be shuffled and selected by maintaining the percentage of 60:40 for both training data and testing data. This approach seems to be the most reasonable approach to split samples because this approach can make sure that every class is included in the model development process. This is the main advantage of stratified sampling which linear and shuffled do not have. For those two sampling techniques, some classes are possibly ignored during the model development. Then, developed models cannot predict ignored classes effectively because they are unknown or totally new classes for developed models.

After the data splitting process, training data is used to train predictive models while testing data is used to test the accuracy of models. In this thesis, the models are developed using Python language. Model tuning is used to ensure that the combination of hyperparameters is proper, which can be conducted in different ways. For example, increasing or decreasing the number of layers, adjusting related parameters, and changing loss functions, activation functions, and optimizers. From this process, a number of models are tested. For performance evaluation, different criteria are possibly used depending on the nature of the problems such as accuracy, R^2, root mean square error (RMSE), mean square error (MSE), mean absolute error (MAE), and mean percentage error (MPE).

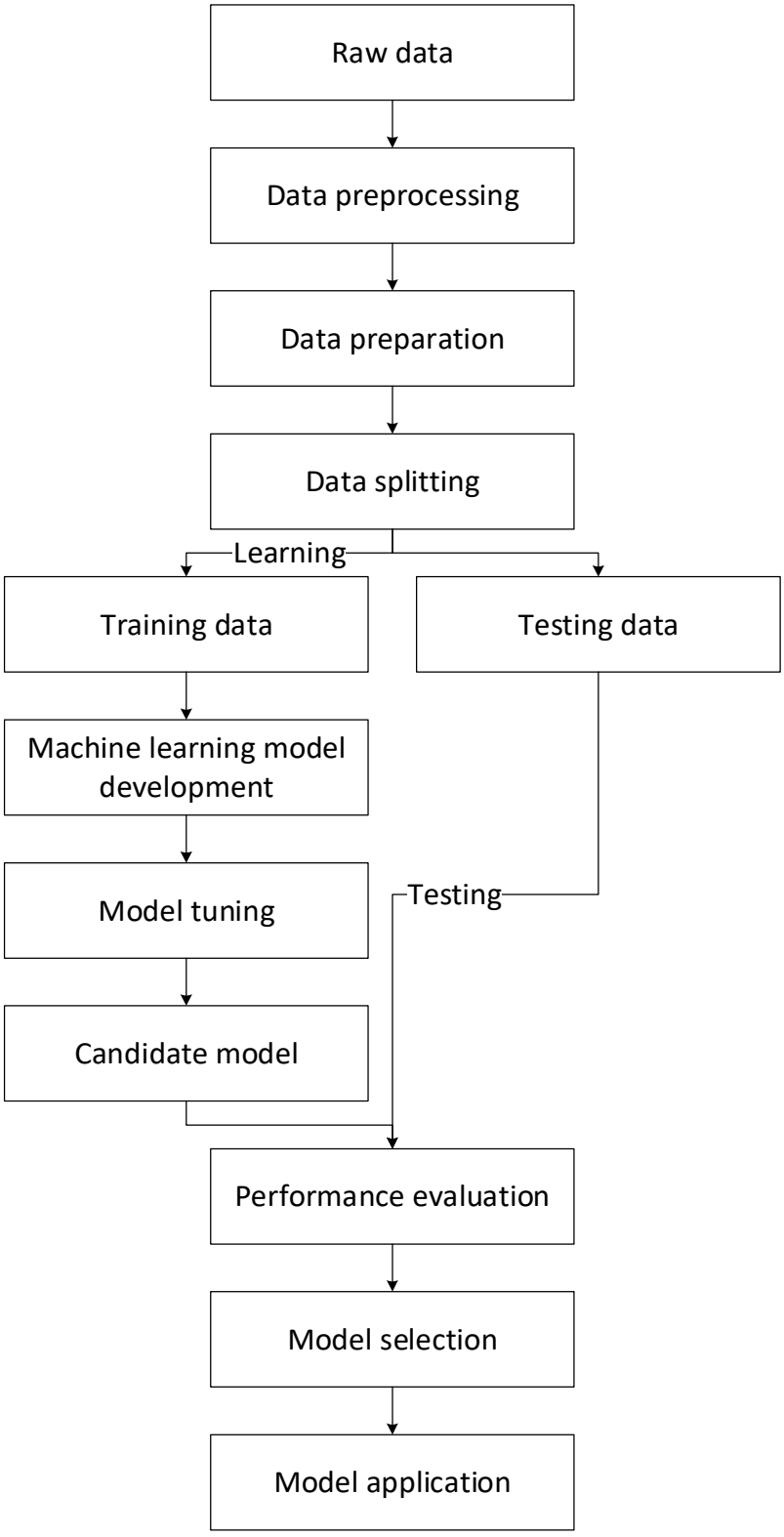


Figure 4-5 General framework of machine learning

4.1.3 Machine learning development platform

To develop machine learning models, there are different approaches that can be done such as the use of software platforms or coding using computer languages. Examples of software platforms are RapidMiner, Weka, IBM Watson Studio, Alteryx, IBM SPSS Modeler, KNIME Analytics Platform, Posit, Tableau, Microsoft Azure, and MATLAB. Examples of computer languages used for developing machine learning models are Python and R. For the software platform, one of the easiest and simplest software is RapidMiner.

RapidMiner is a powerful platform for data preparation and machine learning model development. It provides a user-friendly interface as a drag-and-drop visual workflow designer that allows users to create complex data analysis workflows and machine learning model development without any programming experience and coding. Therefore, it can be used easily which requires a short time for learning. RapidMiner also provides a range of machine learning algorithms and models such as DT, RF, and DNN. It also supports unsupervised learning such as K-mean clustering and association rules. In addition, this software also supports Python script in case there are no operations required to be used for the data analysis or predictive model development. The main benefit of using RapidMiner is it does not require coding to develop machine learning models as mentioned. Another benefit of RapidMiner is it provides a free license for students. Similar software is Microsoft Azure which applies the drag-and-drop approach to develop machine learning models as well. However, the limitation of Microsoft Azure is it is not free software. Therefore, using RapidMiner is a better option. An example of the operation window of RapidMiner is shown in Figure 4-6.

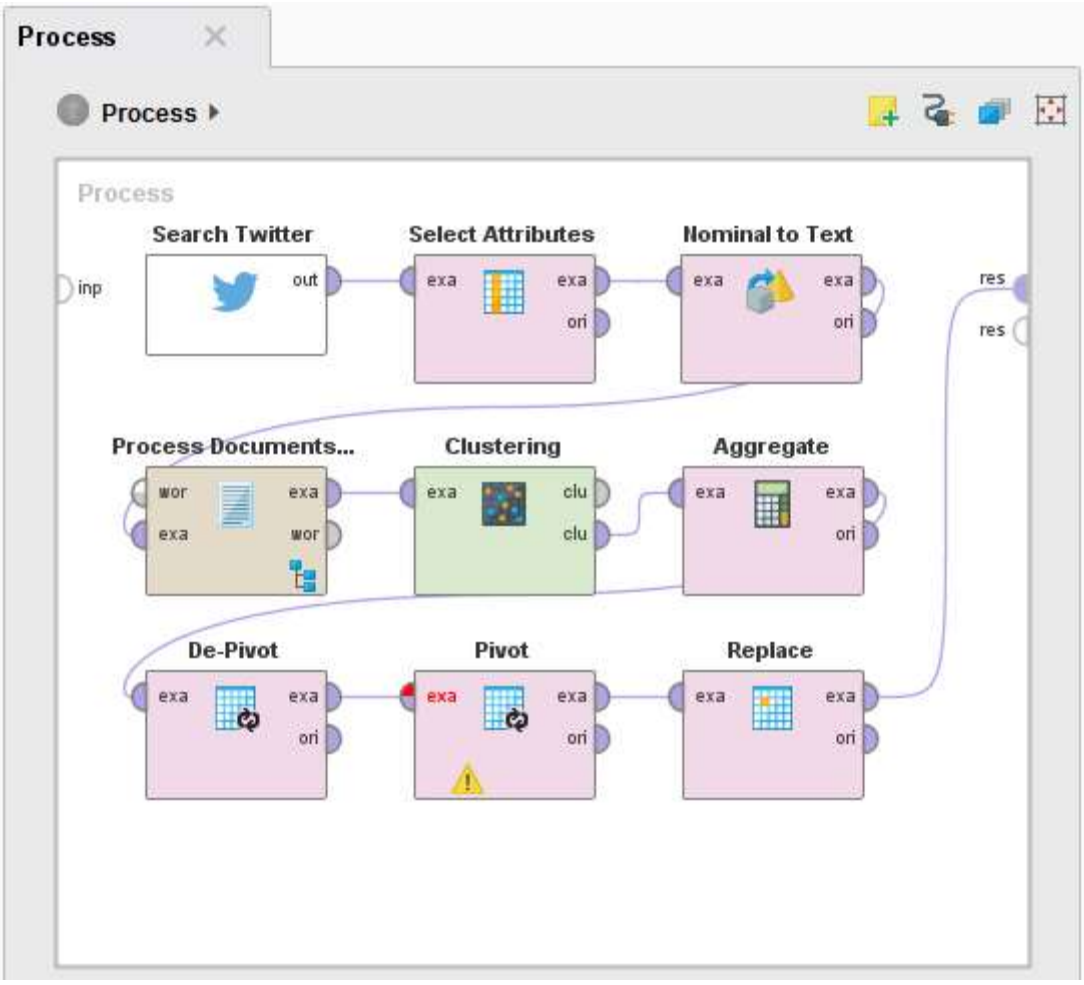


Figure 4-6 Example of RapidMiner operation

However, there are limitations of software platforms such as RapidMiner. That is the software is not flexible enough in some cases to create complicated machine learning models such as reinforcement learning. For flexibility and more complicated models, coding using computer languages is required and unavoidable. For coding, this thesis focuses on Python because it is the most popular language for machine learning development. The community is big and solutions for technical issues can be found easily compared to other languages. The platform used in this thesis is Anaconda. This platform supports data science operations and other languages besides

Python such as R. Anaconda includes many packages required for data science and machine learning by facilitating the development such as NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow. Anaconda supports different source-code editors such as JupyterLab, JupyterNotebook, Spyder, Microsoft Visual Studio Code (VS Code), and RStudio. Users can easily manage packages using the included package manager called Conda. Anaconda also provides a GUI which is Anaconda Navigator. It allows users to manage environments and packages visually which is more accessible for everyone. The Anaconda platform is shown in Figure 4-7.

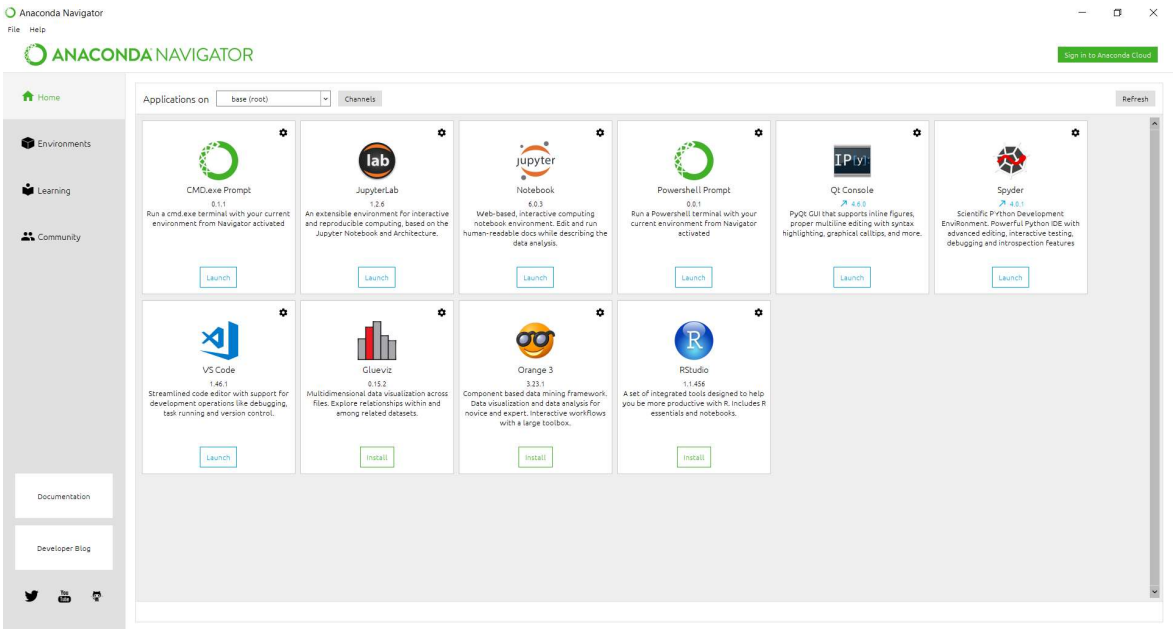


Figure 4-7 Anaconda platform

For a source-code editor, this thesis uses VS code as shown in Figure 4-8. The benefits of this editor are simple to use, an offline platform, and provides a free license for students. A benefit of the offline platform is it does not need to rely on the service providers' server. The training process of machine learning models does not need to wait for other users. In addition, using the offline platform can be integrated with Graphic Processing Unit (GPU) to train machine learning

models which are significantly faster than using only the Central processing unit (CPU). For deep learning, complex model structure, or models concerning a big amount of data, the training time is long because of the time-consuming computations. With GPU, these computations can be performed significantly faster. From a study, GPU is 4-5 times faster than CPU [100]. In the market, GPU is available from many manufacturers such as NVIDIA, AMD, Intel, Google, and Amazon. However, the popular one is NVIDIA GPU because of different advantages such as CUDA is a standard library that is powerful, easy to use, and famous while GPUs from other manufacturers are not as popular as NVIDIA. In addition, the size of the CUDA community is the biggest so the information and the support are the strength of NVIDIA GPU.

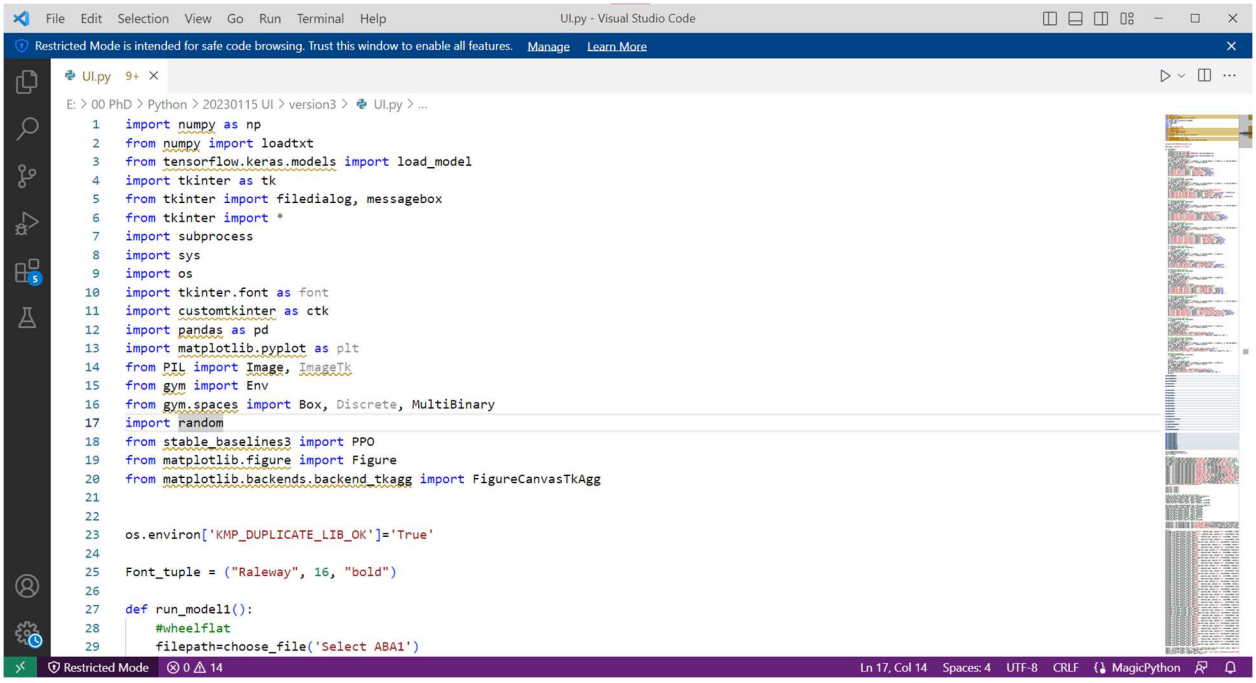


Figure 4-8 VS Code

Machine learning platforms are not limited to what is mentioned in the previous paragraph, Google Colab and Jupyter Notebook are cloud-based platforms that provide users with a free

environment for running Python code. Users can do coding in a browser-based environment in which GPU hardware is available. The free version is available and the paid version is optional for more powerful hardware. Google Colab can be fully integrated with Google Drive so users can store notebooks and data in the cloud and easily share them with others. Google Colab provides built-in support for popular machine learning frameworks such as TensorFlow, Keras, PyTorch, Pandas, NumPy, and Matplotlib. Moreover, additional libraries can also be installed in the environment. It can be seen that the main advantage of Google Colab is it does not require any installation or setup.

For Jupyter Notebook, the main character is the same as Google Colab which is a cloud-based platform. However, one of the key features of Jupyter Notebook is the code can be organized as "cells" which allows users to organize their code into executable blocks. This makes it easy to experiment with the code by sections separately and users do not need to run the whole code to test. Therefore, it can save a lot of time for code testing and debugging.

To summarize, developing a machine learning model can be done using different ways which have different advantages and disadvantages. This thesis will mainly use the coding via Anaconda with VS Code approach which is more flexible although it requires coding. In addition, it does not require the internet for machine learning model development as cloud-based platforms. Therefore, the workflow will be relatively more flexible.

4.1.4 Python libraries

Python libraries are pre-written modules or packages of code that can be imported into a Python machine learning program to provide additional functionality. Libraries contain functions, classes, and variables that can be used to perform specific tasks or solve specific problems such as data analysis and machine learning. Python libraries are developed and maintained by the Python community or third-party developers.

Libraries can be installed using the pip or Conda package manager which is included with Python and Anaconda respectively. Libraries can be applied when they are installed in virtual environments and imported into a Python program using the import statement.

Many Python libraries are available for different applications and functions. Some popular Python libraries are NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, and Keras.

one thing to consider is Python library compatibility. The compatibility of Python libraries is very important and critical when developing machine learning models. It is crucial that all the libraries used in the machine learning models are compatible with each other and with the version of Python being used. If libraries are not compatible, it can lead to errors, crashes, and unexpected behaviour in the application. This can make it difficult to debug and fix issues. To ensure compatibility, it is important to check the version compatibility of each library based on each

library's documentation in the compatibility part. The use of a virtual environment is good practice. Anaconda also provides this function via Conda.

Examples of Python libraries that will be used in this thesis, their functions, and their versions are shown in Table 4-1.

Table 4-1 Examples of Python libraries

Library	Function	Version
Auto-py-to-exe	Convert a Python script into an executable file	2.17.0
Customtkinter	GUI development	5.1.2
Gym	Reinforcement learning algorithm	0.21.0
H5py	Export developed machine learning models	2.10.0
Keras	Machine learning model development	2.3.0
Matplotlib	Data visualization	3.3.2
Numpy	Mathematical operations and data management & analysis	1.19.2
Pandas	Mathematical operations and data management & analysis	1.1.3
Stable_baselines3	Reinforcement learning model development	1.5.0
Tensorflow	Machine learning model development	2.2.0

4.2 BIM

Building Information Modelling (BIM) is a process or digital representation of characteristics of a facility to share resources and information throughout the project life cycle [101]. BIM is used in different industries such as architecture, civil, construction, building system, plant, and structure. The process of BIM application starts with the plan, design, build, and operate stages. The main benefits of BIM are supporting collaboration, better visualization, and improved project outcomes. From its benefits, many governments around the world try to push the application of BIM through regulations and conditions for the overall investment.

By applying BIM, it is expected that the relevant information can be collected and stored comprehensively and properly. Information from the design stage and construction stage of the project will not be ignored but it will be continually used in the maintenance stage as well. This will be beneficial for project management throughout the project life cycle and asset management because the information from the beginning of the project can be used for the benefit of maintenance. At the same time, information in the operation and maintenance stage is also required for the prediction and planning of the maintenance. BIM will act as a platform to share and store this information along with the project life and that will improve the overall project outcome.

This thesis will use BIM as a platform for information management to collect information from the beginning of the project and use that information for the benefit of the maintenance stage.

4.2.1 BIM software

There are different software supporting BIM. Examples of software supporting BIM are Autodesk Revit, ArchiCAD, Trimble Tekla Structures, Vectorworks Architect, SketchUp, Allplan, Rhino 3D, Autodesk Civil 3D, and CGS Ferrovia. From the review, the software that is most suitable for this thesis is Autodesk Civil 3D together with Ferrovia because they are designed to support railway infrastructure. Different from Civil 3D and Ferrovia, other software is designed and normally used for buildings rather than railway infrastructure. An example of a BIM model of a railway track developed using Revit is shown in Figure 4-9 and Figure 4-10. However, it is not very flexible and effective for railway infrastructure.

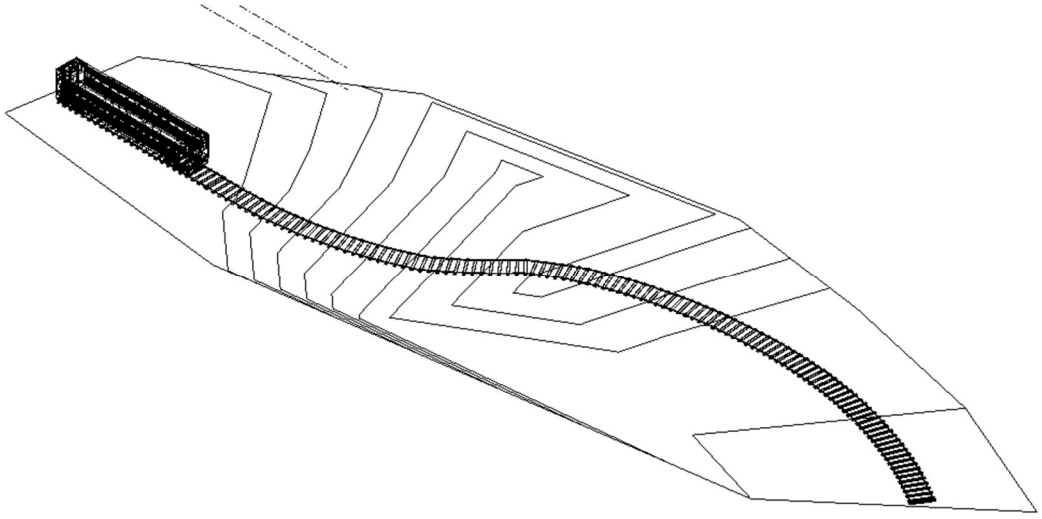


Figure 4-9 Example of railway track BIM model developed using Revit



Figure 4-10 Side views of railway BIM model developed using Revit

Autodesk Civil 3D is software that is widely used in the civil engineering field. It is specifically designed to support civil engineers and designers in designing transportation systems, drainage systems, and construction. Autodesk Civil 3D has functions to create and modify complex design objects such as corridors, pipes, and other objects dynamically. Survey data can be integrated to create topographic models of the existing terrain. The software also has functions to generate documentation automatically which facilitates the tasks such as construction plan generation, profile generation, and cross sections generation. Civil 3D also provides collaborative design tools by allowing multiple designers to work on the same project simultaneously which is the main

characteristic and benefit of BIM. It can be seen that Autodesk Civil 3D is a powerful tool for design and documentation processes.

Ferrovia is a software application developed by CGS Labs, a company that specializes in developing software for civil engineering and infrastructure. It is used for railway design and analysis. CGS Ferrovia is used by railway designers and engineers to design and analyze railway infrastructure, including tracks, bridges, tunnels, and stations. The software offers a range of advanced features for railway design such as 3D modelling, track alignment design, curve design, spiral design, infrastructure design, analysis tools, and documentation.

4.2.2 Dimensions of BIM

With Civil 3D and Ferrovia, models are developed as 3D models. To develop to the higher dimension, the fourth dimension or 4D can be time or schedule [78]. However, only Civil 3D does support this function. Therefore, Navisworks is needed to add time data for additional data. A 5D model is developed from the 4D model by adding cost data. For the 5D model, Civil 3D has a function to store cost data and calculate the total cost. For the sixth dimension or 6D, it can be anything such as project information to evaluate carbon footprint [78], energy consumption, etc. This thesis will use this function to manage the information or significant parameters integrated with machine learning for railway infrastructure predictive maintenance.

5 Supervised learning for railway infrastructure maintenance

Supervised learning is the most popular type of machine learning. The main function of the algorithm is to label samples or datasets for prediction, classification, or regression. The overall process of machine learning model training of this type is the models will be trained to find the optimum weight and biases. Then, when all parameters are optimized, it can be said that the models are fitted and trained. Then, the models will have the capability to predict labels of new unseen samples.

During the training process, the models will get the features or inputs and correct labels or outputs. The models will use them to learn. For each training time, epoch, or iteration, the model will evaluate their performance by using the loss function which there is a variety of them that can be used. Loss functions can be selected to use depending on the nature of the problems. For example, cross-entropy loss is popularly used in multi-class classification problems, and RMSE loss is popularly used in regression problems. When the model is being trained but the accuracy or the performance of the models does not improve, it is a sign that can be used to conclude that the training is done and the models are optimized.

Applications of supervised learning are wide and popular. Some examples of supervised learning applications are image recognition, speech recognition, natural language, and detection. In this thesis, the applications of supervised learning will be described in this chapter.

Classification and regression are two main problems in supervised learning. A classification algorithm is used to assign samples to specific categories. In other words, the labels are discrete. During the model training, the models are trained to learn from hidden patterns in features to predict correct labels. Some examples of classification algorithms are linear classifiers, SVM, DT, RF, ANN, DNN, and CNN. The regression algorithm is used to predict the labels when the labels are continuous. In other words, the labels of regression problems are real numbers. Examples of regression algorithms are linear regression, ANN, DT, RF, and SVR. It can be seen that the main characteristic difference between classification and regression problems is the continuity of labels. For example, a machine learning model used to detect railway defects is a classification model because there are two classes for prediction, “defect” or “no defect”. However, a machine learning model used to evaluate the size of defects or the location of defects is a regression model because labels can be any number from zero to any positive number. Therefore, the labels are continuous.

In this thesis, supervised machine learning will be used to detect defects, evaluate defect severity, and evaluate the track infrastructure conditions in different aspects such as defect locations, types of combined defects, and severities of each combined defect. The supervised machine learning models will be developed in both forms of classification and regression depending on the problems aimed to be solved. The detail of each development will be demonstrated in the following sections of this chapter.

5.1 Classification problems in railway maintenance

5.1.1 Wheelflat detection and severity classification

5.1.1.1 Introduction and backgrounds

Wheelflat is a form of damage that takes place on a train wheel's surface because of excessive, prolonged braking, or sliding & slipping [102]. When a train wheel undergoes excessive braking, it generates a flat area on the wheel's surface, where the wheel contacts the rail. In severe cases, this can lead to a thumping sound, vibration, and even derailment if the severity is high. It can also be described as an out-of-round wheel because the wheel is no longer circular in shape due to wear or other damage. The wheelflat's sharp edge becomes eroded from the impact, making it challenging for humans to detect. Wheelflat can cause extra wear and tear on the rail and other parts of the train, necessitating more maintenance and repair work. Wheelflat can be a safety concern for the train and its passengers. Therefore, it is crucial to frequently monitor and maintain the condition of train wheels to avoid such problems from occurring.

Wheelflats are caused when a train wheel slips or glides along the track while braking. This can be due to a variety of reasons such as slippery rails, excessive velocity, improper brake application, uneven wear, fatigue, material defects in the wheel, or poor condition of brake pads. When a wheel slips, it produces a flat area on the wheel's surface that can differ in size and depth based on the severity of the slip.

To avoid wheelflats, trains should be operated at proper speeds, brake systems must be maintained and tuned frequently, and the wear on wheel profiles should be monitored to keep

the wheel in good condition. If a wheelflat is detected, it needs to be replaced or maintained immediately to avoid further damage to the train or the track. If a wheelflat is detected promptly when its size is small, it can be eliminated by wheel grinding. Even though smaller wheelflats are not as severe as larger ones, they can still cause significant damage although the speed of rolling stock is low [103]. There are various techniques and parameters for detecting wheelflats, such as ultrasonic testing, impact force, wheel radius, and defect detection [104].

This section demonstrates the development of supervised machine learning to detect and classify the wheelflat severity. The techniques used in this section are DNN, 1D-CNN, and LSTM. The machine learning models developed using these techniques are compared for evaluating their performance. The scope of this section focuses on wheelflat only. Other defects, both wheel and rail defects, have not been considered and included in this stage. Data used in this case are numerical data generated from D-Track which is a validated software for MBS. The detail of validation is explained in 3.2.1.

Wheelflat is one of the most common defects found in railway infrastructure. Therefore, there have been many attempts to develop approaches to detect and predict wheelflat. Dong et al. [105] applied the FEM method to model the train track interaction. In that study, they used the developed FEM model to study the dynamic behaviour and they included wheelflat as a defect in the FEM model. They mentioned that the impact force was significantly affected by the speed and axle load of rolling stocks. As mentioned, when the demand for railway transportation is increasing, railway operators have to operate more trains to meet the increasing demand by increasing the speed and road of rolling stocks. Therefore, railway infrastructure and rolling stock

tend to deteriorate faster and defects tend to emerge more frequently. Wheelflat will also take place more frequently. As mentioned, wheelflat can lead to serious incidences such as derailment if the severity is high. In addition, defects with light severity are easier to control and require less maintenance cost to repair. Therefore, predictive maintenance is a useful concept that can be used in this situation. To perform predictive maintenance efficiently, a tool with the sufficient capability to detect wheelflat and evaluate the severity plays an important role.

There have been many attempts to develop different techniques to detect wheelflat. One parameter that is popular to use is axle box acceleration (ABA). In the beginning, ABA was not used to detect wheelflat directly but it was used to further calculate for other parameters such as energy. Bracciali and Cascini [106] applied FFT to process ABA and used the processed data to predict the occurrence of wheelflat. Yue et al. [107] used the same concept to process ABA and used the processed values to detect wheelflat. However, the main difference between this study and the previous study is they applied wavelet transform to process data instead of FFT. However, they found the same conclusion that ABA could be used to indicate wheelflat. They also added that the use of frequency could provide a more accurate prediction because time-domain data could be misleading sometimes. They stated that the processed data by using FFT could be used to predict wheelflat and yielded more satisfying outcomes than the time-domain data. It is worth noting that time-domain data mentioned in that study referred to peak values of ABA only, not the whole raw ABA data. This finding was also confirmed by Belotti et al. [108] who applied a wavelet signal to predict wheelflat. In the study, they set the speed of rolling stock to 10 to 100 km/h. They claimed that they could achieve the accuracy of the prediction by up to

94%. Although frequency-domain data can be obtained by processing time-domain data, some equipment can be used to collect frequency-domain data directly. Bragg Gratings is an example of equipment that can be installed along the track and used to collect data in the form of frequency domain [109]. Ultrasound technology is another approach that can be used to detect wheelflat but the main disadvantages are time-consuming installation and the high cost [104]. Another traditional method is visualization. However, it requires manpower and time. In addition, it can be affected by human error. The machine learning model is an alternative that requires relatively low cost and the time to do prediction is short because of the more computational power nowadays.

Although there are different approaches proposed to detect wheelflat, there is still an argument about the process of detection from the type of data, how to process, and techniques used to predict [110]. Gao et al. [21] used noise as a feature with GA and ANN to predict wheelflat. For the noise, they calculated the mean value and used it as the main feature for prediction. Compare to the traditional method they used in China, they claimed that their developed method could improve the accuracy of prediction by 10%. Li et al. [111] applied the same concept but they added that the noise could be affected by background noise and the noise of the railway infrastructure deterioration. Therefore, they filtered data using adaptive multiscale morphological filtering (AMMF) before using the processed data to predict wheelflat. Krummenacher et al. [112] applied vertical force as the main feature to detect wheelflat using machine learning. They deployed SVM, ANN, and CNN. The accuracy that they obtained was 90% for the wheelflat detection.

From the literature review, deep learning techniques are not popular in detecting wheelflat. In addition, the wheelflat severity classification has not been well explored. Frequency-domain data have not been applied to machine learning. This section will find the potential of combining these concepts together and expect to fill some research gaps in this area.

5.1.1.2 Data and processing

Data for training machine learning models are numerical data generated using D-Track. The detail and validation of the software were described in Section 3.2.1. The frequency of analysis can be adjusted as a function of the software. In this section, the frequency of analysis is defined to be every 0.0005 seconds or 2,000 Hz. From Figure 3-3, wheelflat is a type of defect that can be mimicked in D-Track by customizing the geometry of the wheels. However, the software requires inputting the length and depth of the wheelflat to generate the geometry of the wheelflat. ABAs were exported from D-Track to train machine learning models is the ABAs.

To create the diversity of data and mimic the real situation, different parameters as inputs of D-Track are varied. On the other hand, some parameters will be used for training machine learning models. The first parameter that is varied is sleeper spacing. In this section, the spacings of sleepers are varied from 600 to 750 mm which is the spacing typically used in the railway industry. Speeds of rolling stock are 20 to 200 km/h that suppose to cover the majority of the rolling stock speed used nowadays. The range of the rolling stock weight is 40 to 80 tonnes. The last parameter which is important in this section and will be used as labels of machine learning models is the size of the wheelflat. In this section, machine learning models are developed to detect and classify

the severities of wheelflats. For the length and depth of wheelflats, although the software allows inputting of any numbers as the length and depth of wheelflats, these two numbers have to be reasonable based on the wheels' geometry shown in Figure 5-1. L is the length of wheelflats, D is the depth of wheelflats, and R is the radius of the wheels. In this section, the radius of the wheels is assumed to be 460 mm. The relationship between the length and depth of the wheelflat can be calculated using Equation 1. Then, the function of the wheel related to the rail used in D-Track can be computed using Equation 2.

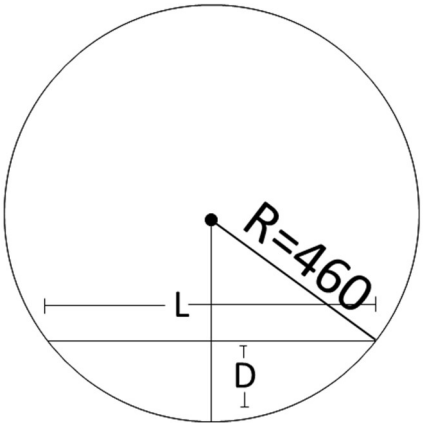


Figure 5-1 Relationship between length and depth of wheelflats

$$R^2 = (R - d)^2 + \left(\frac{L}{2}\right)^2 \quad \text{Equation 1}$$

$$f(x) = \frac{d}{2} * \left[1 - cos\left[\frac{2\pi x}{L}\right]\right] \quad \text{Equation 2}$$

As mentioned, machine learning models in this section are aimed to detect and predict severities of wheelflat, therefore, there are two functions or two machine learning models. Different sizes of wheelflats will be categorized into four levels which are no wheelflat, light defect, intermediate wheelflat, and sever wheelflat. For wheelflat detection, the model will predict two classes which

are no wheelflat and defect when light, intermediate, and severe wheelflats are combined together. For severity classification, no wheelflat class will be removed from the samples. The depth and length of wheelflats based on each level are shown in Table 5-1. The sizes of wheelflats are referred to in the studies done by Vyas and Gupta [113] and Zhai [114] who said the majority of wheelflat has a length of 52 to 60 mm while the depth is ranged from 0.9 to 1.0 mm. Any other parameters that are not mentioned will be fixed. The total number of MBS simulations from D-Track is 1,608. The number of samples in each class can be shown in Table 5-2.

Table 5-1 Depth and length of wheelflats based on levels

Class/Level	Depth of wheelflat (mm)	Length of wheelflat (mm)
0 (no defect)	0.0	0.00
1 (light wheelflat)	0.1	19.18
2 (intermediate wheelflat)	0.5	44.98
3 (severe wheelflat)	1.0	60.63

Table 5-2 Number of samples in each class

Class/Level	Number
0	888
1	240
2	240
3	240

In this section, the main features used in the machine learning models are ABAs. Examples of ABAs from D-Track simulations are shown in Figure 5-2. The figure presents the differences of ABAs when the sizes of wheelflat are the same but the weights of rolling stock are different. Two top plots present the ABAs with wheelflat while two bottom plots present the ABAs when there is no wheelflat. It can be seen that in the case of wheelflat, there are six peak values on both positive and negative sides. These values are used for simplified features for the DNN model

when raw data will be used as features for CNN and LSTM models. For simplified features, nine features will be used which are sleeping spacing, the weight of rolling stock, the speed of rolling stock, and six peak values. To extract simplified features from raw data, data processing is done using Visual Basic for Application (VBA). For raw data, the outcomes from D-Track are ABAs which are time-domain data. To explore the potential of frequency-domain data as mentioned in previous studies, FFT is used to process data. Examples of processed ABAs using FFT are shown in Figure 5-3. One benefit of processing data using FFT is it can process data and provide data with a similar shape. It is worth noting that when the frequency of analysis is the same while the speed of rolling stock changes, the shape of the output will be different. The data with higher speed will have a shorter length of data.

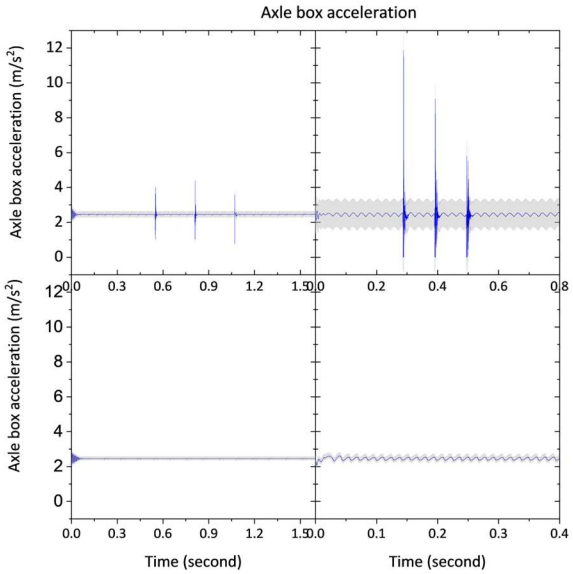


Figure 5-2 Examples of ABAs from D-Track

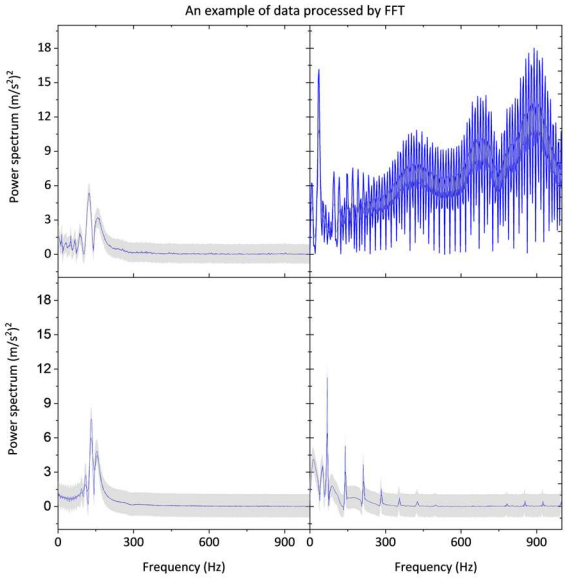


Figure 5-3 Examples of ABAs processed using FFT

Although the above figures demonstrate only one set of ABA per each case, machine learning model training in this section uses two sets of ABAs from both wheels. Samples will be separated into two groups with the proportion of 70% and 30% to be training data and testing data respectively. The sampling method is stratified sampling which the proportion between each class is still the same in both training data and testing data.

5.1.1.3 Methodology

This section uses a deep neural network (DNN), CNN, and LSTM to develop predictive models. Each machine learning model requires different forms of features. For DNN, the architecture of the model is quite similar to ANN but the number of hidden layers is more than one. The features have to be selected and fed into the DNN model by a human. Therefore, it requires feature engineering which is similar to many machine learning models. However, CNN has an advantage

beyond other machine learning techniques because it contains a feature extraction part [115]. This part has a function to select significant features or patterns hidden in data. This process occurs automatically by machine. It can be partially called unsupervised for this part of CNN. From this, feature extraction does not require the knowledge or experience of humans. A benefit is feature extraction will not be affected by human bias. To perform this function, different components in CNN architecture have their own function.

The results of the feature extraction part in CNN are feature mapping. In the Python language, feature mapping is called filters. Similar to other neural networks, features' significances are demonstrated through weights and biases which will be tuned and adjusted during the training process. The number of filters can be defined in the architecture of CNN. More filters refer to the increasing nonlinearity of the model. However, it cannot guarantee that the higher number of filters will result in better accuracy. Therefore, model tuning is a required process for any machine learning model development. The number of filters can be defined in convolutional layers. The feature mapping can be computed by using a moving window called a kernel. The kernel will move along the whole data from the start until the end of the data. The data will be processed using activation functions which can be different depending on the problems aimed to solve.

A pooling layer is another component that is popularly used in CNN. The pooling layers concept is developed by observing that humans can recognize objects or pictures no matter whether those objects or pictures are big or small. CNN was developed based on the concept that humans can recognize things based on areas we see, not only one location. From this concept, it means

that the machine learning model does not require all data to make the prediction. Only some parts of the data can be used to make the prediction. Pooling layers apply this concept by selecting only some significant features for models. The benefits of the pooling layer are it can reduce the sensitivity of the model by reducing unnecessary features from the prediction. In addition, the model can process faster because of the smaller number of features to process and train. Two popular pooling techniques used nowadays are average pooling and max pooling. As their names, average pooling does the calculation by calculating the average value of all values while max pooling will select the maximum value of all values as a representative. A general outcome of the pooling layer is shown in Figure 5-4 and the result from max pooling is shown in Figure 3-8.

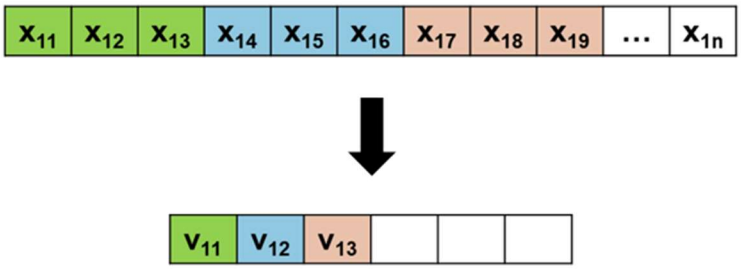


Figure 5-4 Result from pooling layer

The main component of the CNN model is a convolutional layer. Pooling layers are optional. They can be added after convolutional layers or they can be ignored depending on which way provides better model performance. Convolutional layers and pooling layers are combined into the feature extraction part. Another main part of CNN is the classification part. It is worth noting that although it is called the classification part, it can be used for regression problems as well. To connect these two parts, feature extraction and classification parts, a flatten layer is deployed.

The flatten layer will connect results from the feature extraction part to dense layers or fully-connected layers in the classification part. The concept of the flatten layer is simple. The layer will rearrange nD data to 1D data by stacking data in a dimension larger than one to the data in the first dimension. Therefore, it can be imagined that data will turn to linear data when they are processed through the flatten layer. An example of data processed by the flatten layer is shown in Figure 3-10. The concept of flatten layer is shown in Figure 5-5. After that, the data is processed in the classification part. The function of this part is quite similar to the traditional ANN model. There are an input layer, hidden layers, and an output layer. The number of nodes in the input layer is referred to the number of features or feature mapping in this case. The number of nodes in hidden layers and the number of hidden layers can be any number. The higher number refers to the high nonlinearity. The number of nodes in the output layer depends on the characteristics of the problems. For example, if the problem is regression, the number of nodes in the output layer is likely to be one. If the problem is classification, the number of nodes in the output layers is equal to the number of classes. The number of nodes and hidden layers has to be defined at the beginning of the architecture creation. These numbers will not be changed during the training process, different from weights and biases. For parameters that have not changed during the training process, they are called hyperparameters. To maximize the performance of the machine learning model, hyperparameters are also needed to be tuned through the process called hyperparameter tuning which will be described in detail later.

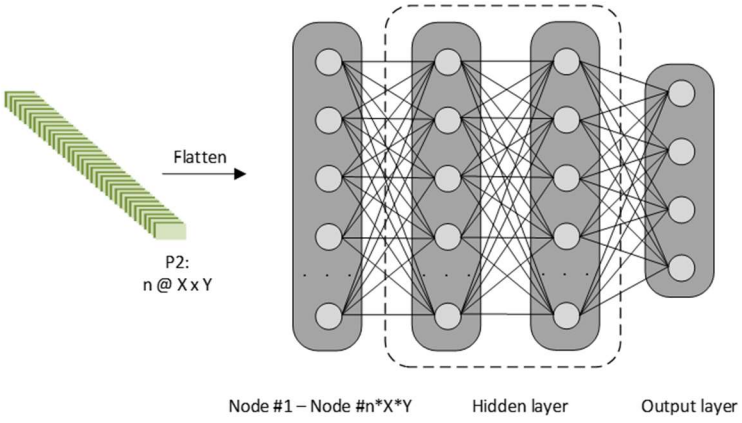


Figure 5-5 The process of the flatten layer

One problem that can be found in machine learning is overfitting. Overfitting happens when the model tends to memorize labels instead of doing predictions. Therefore, the model will not be able to perform well when dealing with unseen or new data. A method that can be used to deal with this problem is developing many machine learning models for prediction and using them to predict labels. Then, the final labels are finalized using the voting system from those developed models. However, this approach is costly and time-consuming. Therefore, the concept of dropout is developed. The dropout concept is some nodes in specific layers will be disenabled during the training process. The disenabling will be random. Therefore, the model will not rely on specific features too much. The model will learn to use every feature to predict labels. This is another approach that can be used to prevent overfitting.

LSTM is another technique used to develop a machine learning model in this section. LSTM is a type of RNN-based model. The characteristic that makes LSTM outstanding from other RNN-based models is it can memorize previous data fed into nodes. Normally, data are independent when they are fed into nodes in neural networks. However, features in LSTM are not

independent. Each feature affects other features and the model considers this relationship to do a prediction. Therefore, the nature of LSTM is beneficial for predicting data in the form of sequences such as speech recognition or natural language. The architecture of LSTM consists of three components which are the input gate, forget gate, and output gate. The function of each gate is represented by its own name. The input gate is used to receive data from the previous node. Forget gate is used to select which data to send to the following node and which data to be forgotten. This is to prevent the model from memorizing everything and the model will need a long time to process the whole set of data. The output gate is used to send data to the next node. A main advantage of LSTM is it does not require feature engineering as CNN because the model can extract the hidden pattern in the data and use it to predict the target labels.

From machine learning model development, DNN, CNN, and LSTM are used. Each model is tuned to optimize performance. For the DNN model, it is found that the architecture that provides the best performance consists of one hidden layer. The number of hidden nodes is 14. Multiple-hidden layers are tried in this case but one hidden layer provides the best performance. Because the problems in this section are classification, the main criterion used to evaluate the performance is accuracy. Other indicators are also demonstrated such as precision and recall. The numbers of nodes in the input layer and output layer of the DNN model are nine and four respectively according to the number of features and classes. An activation function used in the hidden layer is Rectified Linear Unit (ReLu) and the activation function used in the output layer is softmax which is popular in multi-class classification. The architecture of the DNN model is shown in Figure 5-6.

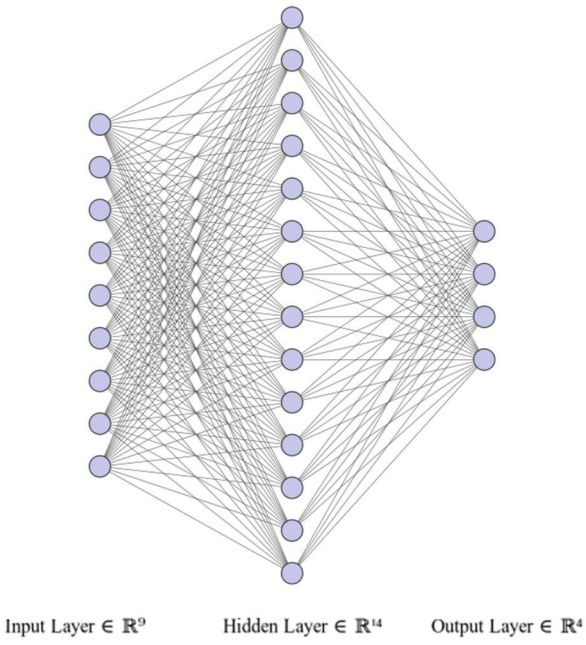


Figure 5-6 DNN architecture for wheelflat detection and severity classification

The architecture of the CNN model is shown in Figure 5-7. The shape of the feature is 2*1*670 which demonstrates there are two sets of 1D data stacked with each other. Then, data are fed into a convolutional layer and the number of filters is 32. Max pooling is used to reduce the sensitivity of the model. Then, the data are processed in another pair of the convolutional layer and max pooling layer. In this stage, dropout is used between the second convolutional layer and the max pooling layer. The percentage of dropout is set to 50% to prevent overfitting. Then, data are flattened and fed into the classification part. For the classification part, there are two hidden layers with 100 hidden nodes before the output layer. Similar to the DNN model, the activation function used in the output layer is softmax when ReLu is used in other layers.

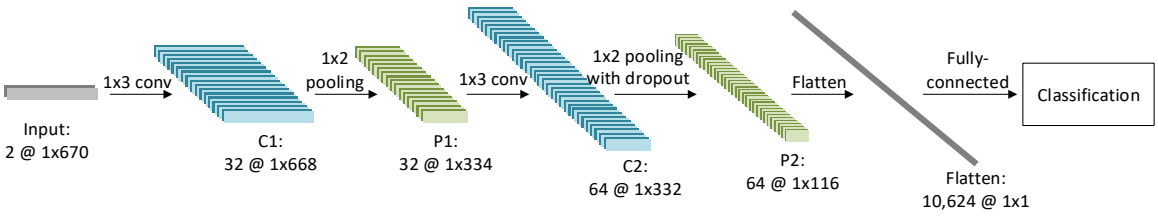


Figure 5-7 CNN architecture for wheelflat detection and severity classification

The LSTM architecture is shown in Figure 5-8. The feature used in the input layer is similar to the CNN model. The LSTM layer consists of 200 LSTM cells. Then, the data are processed through two hidden layers when the numbers of hidden nodes are 100. The activations functions used in different layers are the same as the DNN and CNN models.

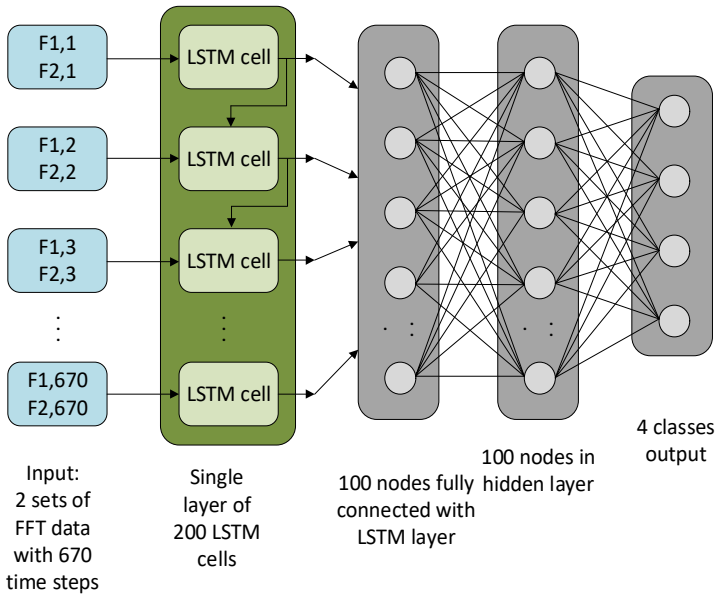


Figure 5-8 LSTM architecture for wheelflat detection and severity classification

5.1.1.4 Results and discussion

To evaluate the model performance, accuracy, precision, and recall are presented. They can be calculated using Equation 3 to Equation 5 where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. True and false do mean their meaning. Positive means the model captures samples as the target while negative means that the model predicts that the samples do not belong to the target. For accuracy, it is used to evaluate the overall performance

of the model while precision and recall are mainly used to evaluate the performance of the model based on each class. Therefore, every class will have its own precision and recall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Equation 3}$$

$$Precision = \frac{TP}{TP + FP} \quad \text{Equation 4}$$

$$Recall = \frac{TP + TN}{TP + FN} \quad \text{Equation 5}$$

For the performance evaluation of the developed machine learning model, three approaches are used which are overall accuracy, accuracy on wheelflat detection, and accuracy on wheelflat severity classification. For overall accuracy, four classes of labels as shown in Table 5-1 are considered. For wheelflat detection, the consideration will be binary classification when levels 1 to 3 are combined to be wheelflat class and level is no wheelflat class. For wheelflat severity classification, levels 1 to 3 are considered. From the model training, the accuracies and epoch or training time can be shown in Figure 5-9. It can be seen that when the models are trained for a while, the accuracies cannot be improved which means that the models are optimized. In other words, the training is finished. From the figure, when the models are trained for 30 epochs, the accuracies are not significantly improved. As mentioned, the testing data are 30% of the samples. The average overall accuracy of the DNN model is the highest followed by the CNN and LSTM models respectively. The overall accuracy of the DNN, CNN, and LSTM models are 0.90,0.80, and 0.77 respectively. Therefore, the use of simplified data is sufficient to provide satisfying outcomes in this case. Besides the highest accuracy, the DNN model required the lowest training time. To

consider the overall accuracy in detail, precision and recall are considered and demonstrated in Table 5-3. The results from the table also confirm that the DNN model performs the best.

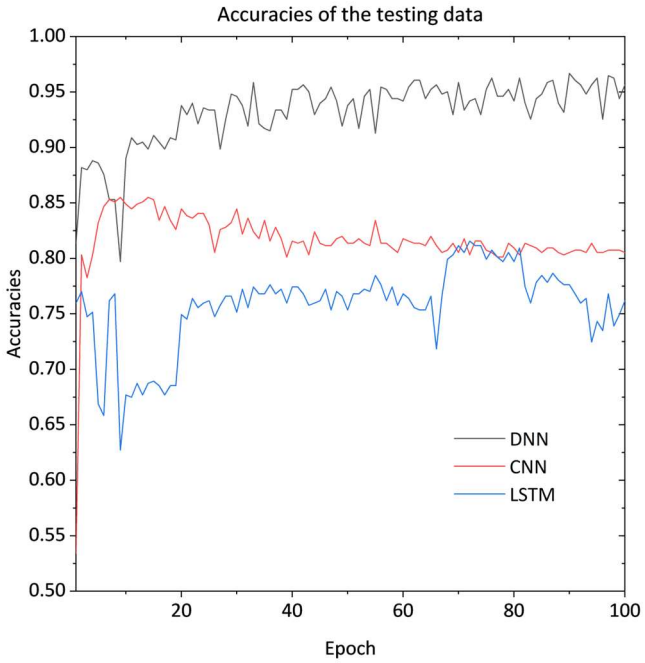


Figure 5-9 The relationship between epoch and accuracies

Table 5-3 Precision and recall of wheelflat predictive model

Class	Accuracy			Precision			Recall		
	DNN	CNN	LSTM	DNN	CNN	LSTM	DNN	CNN	LSTM
Level 0	0.96	0.80	0.77	1.00	0.94	0.73	0.99	0.91	1.00
Level 1				0.97	0.57	1.00	0.99	0.70	0.59
Level 2				0.88	0.77	0.80	0.84	0.55	0.53
Level 3				0.86	0.54	0.76	0.91	0.69	0.45

For the accuracy of wheelflat detection, there are two classes to consider which are wheelflat and no wheelflat classes. Therefore, levels 1 to 3 will be combined together. The accuracy, precision, and recall are shown in Table 5-4. The DNN model also has the highest accuracy followed by the CNN and LSTM respectively. The DNN model has accuracy, precision, and recall close to one in every case. The CNN and LSTM models perform better than each other in different cases.

Table 5-4 Precision and recall of wheelflat detection

Class	Accuracy			Precision			Recall		
	DNN	CNN	LSTM	DNN	CNN	LSTM	DNN	CNN	LSTM
No wheelflat (level 0)	1.00	0.91	0.81	1.00	0.94	0.73	0.99	0.91	1.00
Wheelflat (levels 1-3)				0.99	0.88	1.00	1.00	0.91	0.62

For the wheelflat severity classification, levels 1 to 3 are considered. They are called light wheelflat, intermediate wheelflat, and severe wheelflat according to the sizes of wheelflat. The accuracy, precision, and recall of each predictive model are shown in Table 5-5. The DNN also has the best performance based on these indicators. However, the LSTM model seems to perform better than the CNN model in this case. The LSTM model has a higher accuracy and many cases of precision and recall.

Table 5-5 Precision and recall of wheelflat severity classification

Class	Accuracy			Precision			Recall		
	DNN	CNN	LSTM	DNN	CNN	LSTM	DNN	CNN	LSTM
Light wheelflat	0.91	0.70	0.84	1.00	0.67	1.00	0.99	0.75	1.00
Intermediate wheelflat				0.88	0.84	0.80	0.84	0.60	0.81
Severe wheelflat				0.86	0.63	0.76	0.91	0.78	0.74

In terms of training time, the DNN and CNN models require a small amount of time to train the models. Both of them require a millisecond to train the model for one epoch. However, the training time of the LSTM model is significantly higher than the other two models. It takes 11 seconds to train one epoch. The long training time of LSTM seems to be an issue in this case as found in many previous studies.

5.1.2 Railway combined defect detection and severity classification

5.1.2.1 Introduction and backgrounds

In this previous section, the application of machine learning to predict wheel defects was demonstrated. In this section, the application of machine learning to predict rail defects is explored. In addition, real practice is considered. Defects do not occur independently. Multiple defects can take place at the same time and they will be more challenging for railway operators to detect and evaluate. In this section, machine learning potential will be explored to see whether it can be used to predict combined defects. Dipped joint and settlement are used as case studies.

Dipped joints are defects occurring at rail joints as their name. Physically, dipped joints are the lowering of rail surface level at rail joint locations. Dipped joints emerge during the operation that the shape of the rail deteriorates. For the settlements, they are phenomena that track section sinks below the design level. Settlements can take place because of many reasons such as poor construction, deterioration of foundation, poor drainage system, or settlement because of normal operation. Settlements cannot be avoided because tracks will settle over time although other factors are perfect. As with other rail defects, dipped joints and settlements negatively

affect passenger comfort, induct other defects, and cause derailment when the defects are severe. Dipped joints and settlements can be inspected using different approaches such as visualization [116], ultrasonic [117], magnetic induction [118], or eddy current [119]. These approaches are called non-destructive testing (NDT). A main benefit of NDT is the testing will not affect the railway infrastructure and does not create any damage to the railway infrastructure. In addition, NDT is cost-effective and requires a small amount of time to perform. Machine learning can be considered a type of NDT as well because the testing does not create any damage to the railway infrastructure.

In this section, different machine learning techniques are used to develop predictive models. The deployed techniques are DNN, CNN, and LSTM. The aims of the machine learning model application are to detect dipped joints and settlements including evaluate their severity. The features used to train the machine learning models are ABAs. As mentioned, the use of ABAs has many advantages. First, it is fast to collect data because data collection can be done during regular operation by installing accelerometers at axle boxes. In addition, data collection will not interrupt the operation because any rolling stocks running on the target tracks can be used as data collection equipment. The cost of data collection is not expensive because accelerometers are relatively cheap compared to other advanced equipment. Last, the conditions of tracks can be monitored regularly because the data collection does not interrupt the regular operation. Predictions can be done in a short time because machine learning can do predictions due to the increasing computational power nowadays. As in the previous section, this section uses numerical data generated by D-Track. Differences between this section and the previous section

are that combined defects are considered, rail defects are explored instead of wheel defects, and combined defects are considered as rail irregularities which have to be customized because there is no available function in D-Track to input combined defects. Therefore, customized geometries of rail will be inputted into the software in the form of a CSV file.

Núñez et al. [70] applied ABA to detect rail defects. In that study, exampled defects were squats and corrugations. In their study, they could achieve an accuracy of 85%. Li et al. [71] also applied ABA to defect rail defects. Their targeted defects were light squats and their findings confirmed the same results as the previous studies that ABA can be used to detect and predict different defects in the railway industry.

For machine learning, another technique that is popular is computer vision or image processing. Many studies performed predictions using this technique [14, 112, 120-126]. However, the main drawbacks of image processing are the accuracy of the prediction is based on the quality of the images. When the speed of data collection is higher, the quality of collected images tends to drop significantly. Another drawback is image capturing has a limitation when the light is limited because it significantly affects the quality of images. Although there have been studies focusing on more than one defect, the studies focusing on combined defects when multiple defects take place at the same time are limited. This research gap is aimed to be fulfilled by the study presented in this section.

5.1.2.2 Data and processing

As in the previous section, data used in this section are numerical data generated by using D-Track. Different parameters are varied to create data diversity and mimic the real situation. Parameters that are varied can be summarized in Table 5-6. It is worth noting that settlements in this section can be categorized into two groups which are short settlements and long settlements. The lengths of short and long settlements are 3,000 and 10,000 mm respectively. Features used in this section are ABAs generated by using D-Track.

Table 5-6 Summary of varied parameters for studying dipped joints and settlements

Parameters	Value
Sizes of dipped joint	0-10 mm. (The length of the dipped joint is 1,000 mm.)
Sizes of settlement	0-100 mm. (The lengths of the settlement are 3,000 and 10,000 mm for short and long settlements respectively)
Speeds of vehicle	20-200 km/h
Weights of vehicle	40-80 tonnes
Rail type	AS60
Gauge	Standard gauge
Pad type	HDPE
Sleeper type	Prestressed
Sleeper spacing	600-750 mm.
Track bed stiffness	50.1 MN/m
Track bed damping	159 kNs/m
Primary suspension stiffness	1.22 MN/m
Primary suspension damping	4 kNs/m
Wheel radius	460 mm.
Hertzian contact coefficient	7.25E+10 $N/M^{3/2}$

From the list of simulations, the total number of samples generated by D-Track is 1,605. Examples of ABAs exported from D-Track are shown in Figure 5-10 and Figure 5-11. Both show the ABAs when the speed of rolling stock is 20 km/h and the weight of rolling stock is 40 tonnes. The difference between these two cases is the first figure presents the ABA when the track is defect-free and the second figure presents the ABA when the sizes of the dipped joint and settlement are 2.5 mm and 20 mm respectively and the settlement is a short settlement as shown in Figure 5-12. From the figures, it is easy to recognize when the track is free from defects. However, it will be more complicated when the track has combined defects. From Figure 5-11, there are peak values that are caused by dipped joints and settlement. These peak values will be used as simplified features from the DNN model as Section 5.1.1 when raw data will be used as raw data for the CNN and LSTM models. As in the previous section, there are two sets of raw data from two wheels. Those two sets of data will be used to train machine learning models.

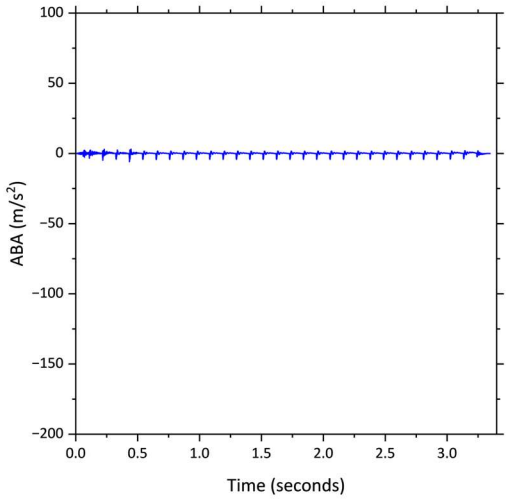


Figure 5-10 Example of ABA when there is no defect

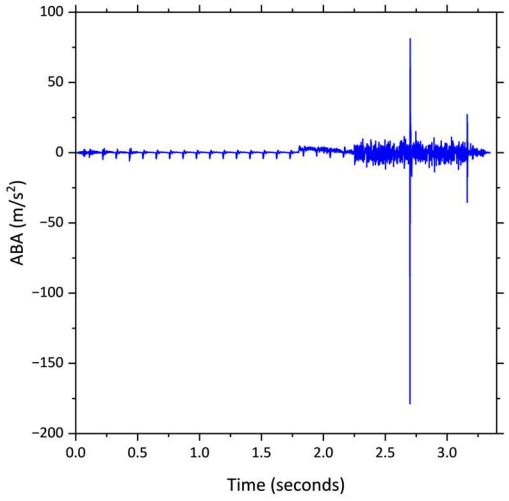


Figure 5-11 Example of ABA when a track is defective

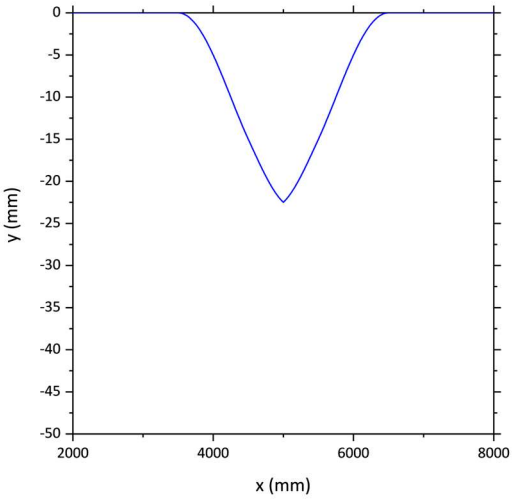


Figure 5-12 Geometry of the track with dipped joint (2.5 mm) and short settlement (20 mm)

For raw data, ABAs will be used directly for the CNN and LSTM models. Different from the previous section, ABAs are not processed to be frequency-domain using FFT but will be used in the form of time-domain data. For simplified data for the DNN models, 14 features are used consisting of the weight of rolling stock, speed of rolling stock, and six peak values of ABAs in

positive and negative sides from both wheels. To process simplified data, VBA code is used as in the previous section to extract the simplified features from raw data. It is worth noting that the speed and weight of rolling stock are known in the beginning from sensors installed on rolling stock. Labels of each sample are done based on the sizes of dipped joints and settlements. Labels are shown in Table 5-7. As in the previous sections, machine learning models are used to detect and evaluate the severity of dipped joints and settlements. However, combined defects are considered in this case. The development approach will be slightly different from the previous section. In brief, a hypothesis will be tested whether two models developed for detecting each defect or a model developed to detect both defects will have better performance. Then, two individual models are developed to predict the severity of dipped joints and settlements separately.

Table 5-7 Labels of dipped joints and settlements based on sizes

Purpose	Label	Size of defect
Dipped joint severity classification	0	2.5 mm or smaller
	1	2.5-5.0 mm.
	2	Bigger than 5.0 mm.
Settlement severity classification	0	20 mm or smaller
	1	20-60 mm.
	2	Bigger than 60 mm.

For a single model to detect dipped joints and settlements together, there are four classes of labels which are no defect, dipped joint, settlement, and dipped joint and settlement. For two models to detect dipped joints and settlements separately, each model will classify two classes which are no defect and defect. For the models to evaluate the severity of dipped joints and settlements, the labels are based on the sizes of defects as shown in Table 5-7.

5.1.2.3 Methodology

The workflow of the machine learning models is shown in Figure 5-13. In this case, a regression part to evaluate the sizes of defects is also added for the completion of the study. For the classification problems which are detection and severity evaluation, the same indicators used in 5.1.1 are used. For the regression problem which is the size of defect prediction, MAE is used to evaluate the model performance.

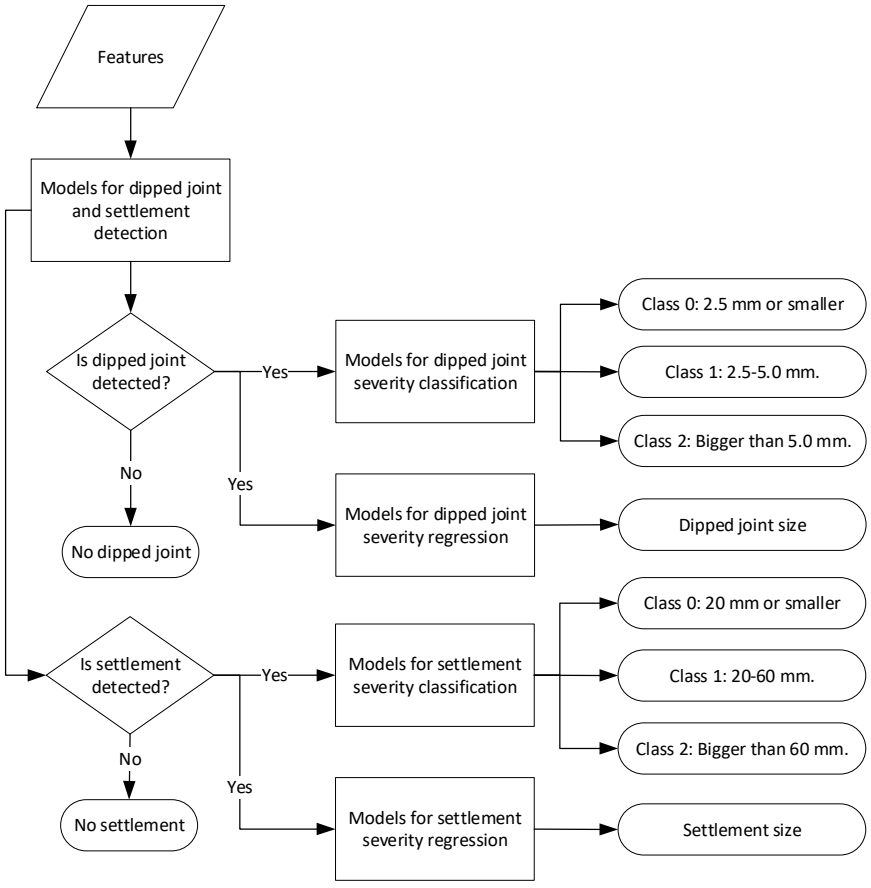


Figure 5-13 Workflow of a machine learning model for dipped joint and settlement detection and severity evaluation

Stratified sampling is used to separate the training data and testing data for training and evaluating models' performance. They will be separated with the proportion of 70% and 30% respectively. To ensure the performance of the developed machine learning models, hyperparameter tuning is also performed. The list of hyperparameters tuned is shown in Table 5-8.

Table 5-8 List of hyperparameters tuned for combined defect detection and severity classification

Model	Tuned hyperparameter	
DNN	The number of hidden layers The number of hidden nodes Activation function Batch size	Learning rate Momentum Optimizer Dropout
CNN	The number of convolutional layers Filter Kernel The number of max pooling layers Pool size The number of hidden layers The number of hidden nodes	Activation function Batch size Learning rate Momentum Optimizer Dropout
LSTM	The number of LSTM cells The number of hidden layers The number of hidden nodes Activation function Batch size	Learning rate Momentum Optimizer Dropout

5.1.2.4 Results and discussion

For the models that are used to detect dipped joints and settlements, the performances can be shown in Table 5-9. From the table, the CNN model has the highest accuracy. The value of almost 1 indicates that the model can capture all defects and be reliable when used for prediction. The DNN model performs better than the LSTM model. Therefore, the use of raw data and simplified

data does not guarantee which approach provides better performance but they need to be tested. Another conclusion that can be drawn is the use of data in the form of time-domain data without processing to be frequency-domain also has the potential to provide a satisfying outcome. As the finding found in the previous section, the LSTM model has the lowest accuracy. At the same time, the processing time for training is the longest.

Table 5-9 Performance on dipped joint and settlement detection

Model	Accuracy
DNN	0.86
CNN	0.99
LSTM	0.79

The list of hyperparameters of the CNN model that are tuned to provide the best performance of each model is shown in Table 5-10.

Table 5-10 Tuned hyperparameters for the CNN model to detect dipped joint and settlement

Model	Tuned hyperparameter	Value
CNN	The number of convolutional layers	2
	Filter	80 (conv1) and 64 (conv2)
	Kernel	5
	The number of max pooling layers	2
	Pool size	3
	The number of hidden layers	2
	The number of hidden nodes	100
	Activation function	ReLu (except dense3 which uses Softmax)
	Batch size	64
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	N/A

The second approach uses two models to detect dipped joints and settlements. This approach is used because of the assumption that the machine learning model with a specific purpose should perform better than the model that has many functions. In this case, machine learning models have two classes to predict instead of four classes. Therefore, they should perform better. The performances of the machine learning models developed using this approach are shown in Table 5-11. It can be seen that the CNN model has the highest accuracy on both dipped joint and settlement detection. The DNN and LSTM models perform better than each other in different cases. The DNN model performs better on dipped joint detection while the LSTM model performs better on settlement detection. The CNN model has an accuracy on dipped joint detection equal to 1.00 which is the perfect number. However, when the overall accuracy is calculated by multiplying two accuracies, the overall accuracy is 0.99 which is similar to the first approach. This phenomenon also occurs in the case of the DNN and LSTM models. Therefore, it can be concluded that the use of a model to detect both dipped joint and settlement at the same time and the use of separated models to detect dipped joint and settlement independently do not provide a different performance.

Table 5-11 Performance on dipped joint and settlement detection using separate predictive models

Model	Accuracy
Dipped joint detection	
DNN	0.94
CNN	1.00
LSTM	0.86
Settlement detection	
DNN	0.92
CNN	0.99
LSTM	0.94
Total accuracy	0.99

The tuned hyperparameters of the CNN models to detect dipped joints and settlements separately are shown in Table 5-12.

Table 5-12 Tuned hyperparameters for the CNN model to detect dipped joint and settlement separately

Model	Tuned hyperparameter	Value
CNN for detecting dipped joint	The number of convolutional layers	2
	Filter	64 (conv1) and 32 (conv2)
	Kernel	7
	The number of max pooling layers	2
	Pool size	2
	The number of hidden layers	2
	The number of hidden nodes	100
	Activation function	ReLu (except dense3 which uses Softmax)
	Batch size	64
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	0.25 (before maxpooling2)
CNN for detecting settlement	The number of convolutional layers	2
	Filter	80 (conv1) and 64 (conv2)
	Kernel	6
	The number of max pooling layers	2
	Pool size	4
	The number of hidden layers	2
	The number of hidden nodes	100
	Activation function	ReLu (except dense3 which uses Softmax)
	Batch size	64
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	N/A

When samples are detected that the track sections have dipped joints or settlements, samples will be further fed into the machine learning models for severity classification. In this case, there are many classes to detect. Therefore, severity classification for dipped joints and settlements will be developed separately to ensure the performance of the predictive models. As shown in

Table 5-7, there are three classes in both dipped joint and settlement severity classification. They can be called light, intermediate, and severe defects. The performances of the predictive models are shown in Table 5-13. From the table, the CNN model is the best model for classifying the dipped joint severity when the accuracy is 0.84. However, the LSTM model is the best in classifying the severity of settlements. Therefore, the overall accuracy is 0.83. It can be seen that the accuracy of the CNN model on classifying the severity of settlement is in the same range as the LSTM model which is 0.95 which is acceptable as well. Another way to present the performance of the machine learning model on the classification is the confusion matrix. A confusion matrix presents the relationship between the predicted labels and true labels of samples. Normally, a confusion matrix is present in a tabular form when rows represent true labels and columns represent predicted labels. The confusion matrix of dipped joint and settlement severity classification is shown in

Table 5-14 and Table 5-15 respectively. If the developed predictive models perform well, it is expected that all numbers should be in the diagonal cells.

Table 5-13 Performance of the predictive models on severity classification

Model	Accuracy
Dipped joint severity classification	
DNN	0.61
CNN	0.84
LSTM	0.51
Settlement severity classification	
DNN	0.51
CNN	0.95
LSTM	0.99
Total accuracy	0.83

Table 5-14 Confusion matrix of dipped joint severity classification using the CNN model

	Predicted class 0	Predicted class 1	Predicted class 2
Actual class 0	75	8	16
Actual class 1	5	69	24
Actual class 2	1	9	189

Table 5-15 Confusion matrix of settlement severity classification using the LSTM model

	Predicted class 0	Predicted class 1	Predicted class 2
Actual class 0	94	1	0
Actual class 1	1	171	0
Actual class 2	0	1	182

From hyperparameter tuning, the tuned parameters of the CNN model for classifying dipped joint severity and the LSTM model for classifying settlement severity are shown in Table 5-16.

Table 5-16 Tuned hyperparameters for the predictive model to classify dipped joint and settlement severity

Model	Tuned hyperparameter	Value
CNN for classifying dipped joint severity	The number of convolutional layers	2
	Filter	32
	Kernel	9
	The number of max pooling layers	2
	Pool size	3 (maxpoo1ing1) and 6 (maxpooling2)
	The number of hidden layers	2
	The number of hidden nodes	80
	Activation function	ReLu (except dense3 which uses Softmax)
	Batch size	8
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	N/A

Model	Tuned hyperparameter	Value
LSTM for classifying settlement severity	The number of LSTM cells	200
	The number of hidden layers	2
	The number of hidden nodes	100
	Activation function	ReLu (except dense3 which uses Softmax)
	Batch size	64
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	N/A

In addition, machine learning models are developed to evaluate the size of dipped joints and settlements which can be considered regression problems. Different from the classification problem, the labels of the regression problem are continuous. In this case, the labels are the sizes of defects so they can be any numbers from zero to infinity. The performance of the machine learning models for predicting the sizes of dipped joints and settlements is shown in Table 5-17. MAE is used as an indicator for the defect size evaluation. Different from accuracy, precision, and recall, the lower MAE indicates better performance. From the table, it can be seen that the CNN model performs the best when used to evaluate the size of the dipped joint while the LSTM model is the best model for evaluating the size of the settlement. In both cases, MAEs are about 1 mm which is desirable when evaluating defect size. The DNN model for evaluating the size of the settlement does not perform well as can be seen from the significantly high MAE.

Table 5-17 Performance of the predictive models on defect size prediction

Model	MAE (mm)
Dipped joint severity regression	
DNN	2.01
CNN	1.25
LSTM	2.54

Model	MAE (mm)
Settlement severity regression	
DNN	21.14
CNN	3.13
LSTM	1.58

The relationships between the predicted values and true values of the size of dipped joints and settlements are shown in Figure 5-14 and Figure 5-15 respectively. R^2 or the coefficient of determination is also presented to evaluate the performances of the models. R^2 presents how predicted values fit with the true values. The possible range of the R^2 is zero to one. The higher R^2 represents the better performance of the model. However, it is worth noting that only R^2 is not sufficient to indicate the performance of the models. Normally, it has to be used with other indicators to measure the performance of the model such as MAE which is conducted in this section. From Figure 5-14, the R^2 of the CNN model is the highest which is 0.98 while the other two models have low R^2. From Figure 5-15, the R^2 of the LSTM model is the highest and the R^2 of the CNN model is in the same range which is 1.00 and 0.99. These results conform to the outcome in Table 5-17. However, the use of R^2 presents that CNN performs quite well in both cases.

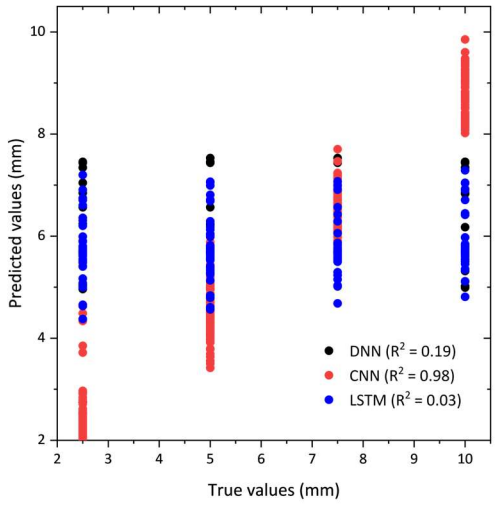


Figure 5-14 True values and predicted values of dipped joints' size from developed machine learning models

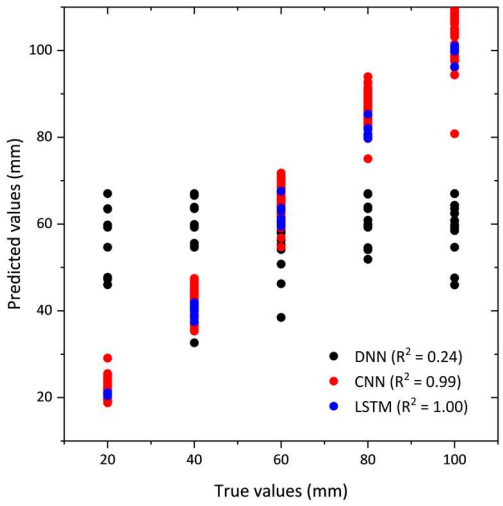


Figure 5-15 True values and predicted values of settlements' size from developed machine learning models

From hyperparameter tuning, the tuned hyperparameters of the CNN model for predicting the size of dipped joints and the LSTM for predicting the size of settlements are shown in Table 5-18.

Table 5-18 Tuned hyperparameters for predicting the size of dipped joints and settlements

Model	Tuned hyperparameter	Value
CNN for predicting the dipped joint size	The number of convolutional layers	2
	Filter	32 (conv1) and 64 (conv2)
	Kernel	5 (conv1) and 4 (conv2)
	The number of max pooling layers	2
	Pool size	3 (maxpoo1ing1) and 2 (maxpooling2)
	The number of hidden layers	2
	The number of hidden nodes	100 (dense1) and 50 (dense2)
	Activation function	ReLu (except dense3 which uses Linear)
	Batch size	16
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	N/A
LSTM for predicting settlement size	The number of LSTM cells	200
	The number of hidden layers	2
	The number of hidden nodes	200
	Activation function	ReLu (except dense3 which uses Linear)
	Batch size	8
	Learning rate	0.001
	Momentum	0.9
	Optimizer	Adam
	Dropout	N/A

5.1.3 Track component defect prediction based on track geometry

5.1.3.1 Introduction and backgrounds

The indicators that are used to evaluate the track condition are track geometry parameters and track component inspection. Normally, track geometry parameters consist of superelevation, alignment, longitudinal level, gauge, and twist. The track geometry parameter diagram can be shown in Figure 5-16. As with other inspections in the railway industry, the inspection can be done using different approaches. One of the popular approaches used to measure the track

geometry parameters is TGC. TGCs are normally used to measure track geometry parameters regularly to inspect the condition of tracks. Therefore, almost all railway operators suppose to have this information. If railway operators can use track geometry parameters to detect track component defects, it will be beneficial to railway operators and the railway industry because track geometry parameter information tends to be available while track component defects require additional inspection. Therefore, the inspection of track component defects is relatively expensive and time-consuming. In general, the speed of TGCs is 70 mph so data collection can be done quickly. This section aims to develop machine learning models to predict track component defects using track geometry parameters as features.

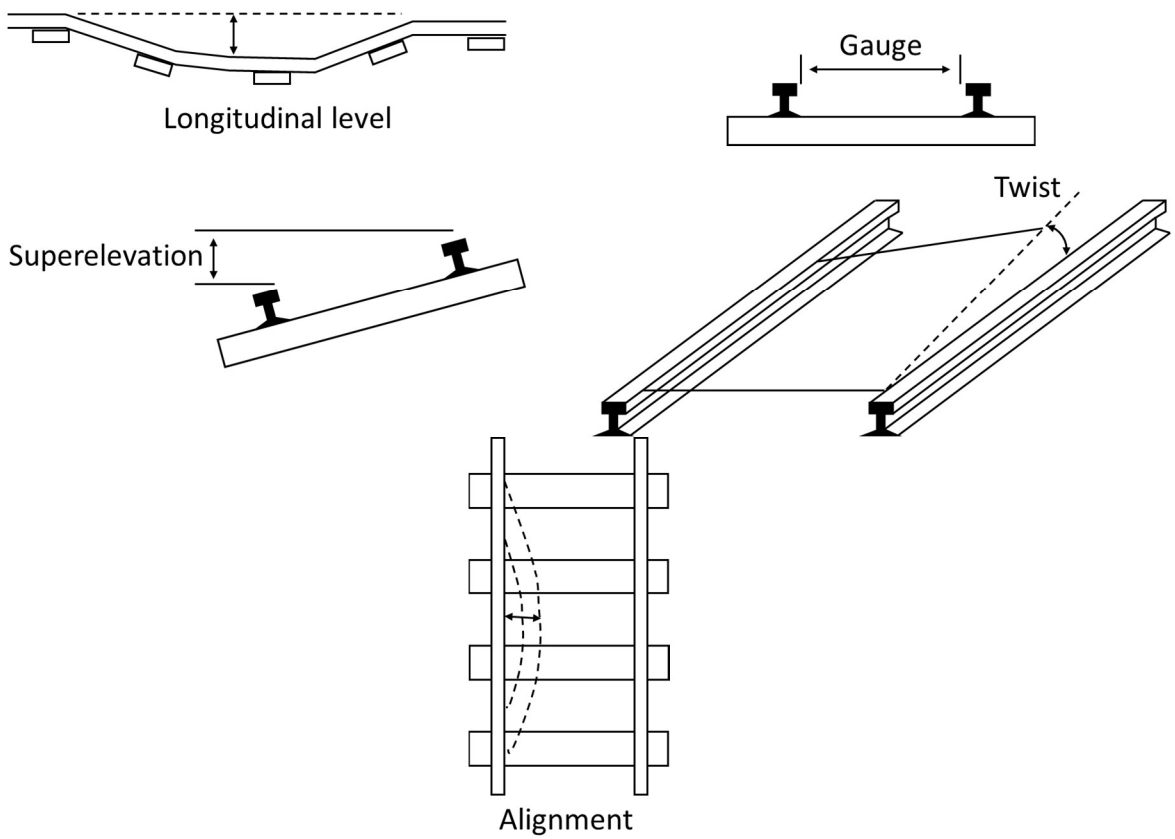


Figure 5-16 Track geometry parameters

Different from Sections 5.1.1 and 5.1.2 which use simulated data to develop machine learning models, this section will use field data to develop machine learning models. Data are provided by MRS Logística S.A. in Brazil. More detail about the field data will be described in 5.1.3.2. In brief, track geometry parameters are collected using TGCs that collect data every foot. The total length of the studied track section is 30 km. The duration of the available data is four years from 2016-2019. Track component defects are collected from defect inspection reports in the related period. From the report, there are more than 81 types of defects. However, to simplify the data, different types of track component defects will be grouped according to track components. Interesting track components in this section are rail, switch and crossing, fastener, and rail joint. Machine learning techniques that will be used in the predictive models are DNN, CNN, multiple regression (MR), SVM, gradient boosting (GB), DT, and RF. It is believed that the outcome of this machine learning model development will improve the efficiency in railway inspection and maintenance especially predictive maintenance of the railway infrastructure because the monitoring can be done more regularly and severe damages can be avoided.

Lasisi and Attoh-Okine [65] applied track geometry parameters to calculate TQIs which is a traditional method. Then, they developed a machine learning model to predict track defects. In their study, binary classification was used for labelling. Two classes of labels were defect and no defect. From their study, SVM was the best model for predicting when the accuracy could be achieved up to 90%.

Martey et al. [127] focused on the substructure of the railway system. They explored the effect of geocell and geogrids that were installed in the track substructure. They applied many machine

learning techniques to study the change in the installation of geocells and geogrids. They found that the RF model was the best model to capture the change of the improvement.

Sadeghi and Askarinejad [128] tried to detect track component defects based on track geometry parameters. They focused on rail, sleeper, fastener, and ballast. They stated that only track geometry parameters were not sufficient to evaluate the track condition. They added that the track component defects tend to occur more frequently when the standard deviation (SD) that is popularly used to evaluate the track conditions via track geometry parameters is higher. From their study, they concluded that different track component defects were affected by track geometry parameters at different levels. For example, gauge significantly affected the defect of fasteners while profile significantly affected the ballast defects. Therefore, there was no clear conclusion about the relationship between track geometry parameters and track component defects.

Zarembski et al. [129] tried to find the relationship between track geometry parameters and rail defects. They applied MR as the machine learning technique to develop a predictive model. However, the finding from their study did not provide a clear relationship between the track geometry parameters and rail defects. From their study, the correlation between the track geometry parameters and rail defects was only 11%.

Soleimanmeigouni et al. [67] also applied the data from TGCs to develop a data-driven approach for predicting track geometry defects. the data used in their study were collected from 2015 to 2018 when the total distance of the interesting section was 84 km. However, the main limitation

of their study was they focused on only one track geometry parameter which was the longitudinal level of the tracks. They developed the predictive model using logistic regression. Features used in their study to develop the predictive model were SD and kurtosis of longitudinal level. Because they used logistic regression to develop the predictive model, the problem they tried to solve was binary classification when the classes were defect and no defect. It can be seen that there are many limitations in their study such as the number of parameters they tried to predict and the outcome that they could predict.

Mohammadi et al. [23] studied the relationship between track geometry parameters and rail defects using machine learning. The machine learning techniques used in their study were RF, logistic regression, XGBoost, and SVM. Their problem was similar to many studies done in the past which was binary classification when they tried to detect defects. They could achieve an accuracy of 80% when using XGBoost. The limitation of this study is it focused on only rail defects. Therefore, the comprehensiveness of the prediction is still limited.

It can be seen that there have been many studies focused on the relationship between track geometry parameters and rail defects because they were directly related. However, the study about the relationship between track geometry parameters and comprehensive track component defects is very limited. This section will explore the potential of machine learning in predicting track component defects based on track geometry parameters.

5.1.3.2 Data and processing

In this section, data are from three sources which are track geometry parameters collected using TGCs, track component defects from defect inspection reports, and track design from design reports. The length of the section is 30 km. The duration of data collection is from 2016 to 2019.

For track geometry parameters, TGCs are used to collect foot-by-foot data. This data measurement is done annually. In total, there are more than 95,000 samples of data from this part. From the measurement, track geometry parameters consist of 12 parameters which are superelevation, longitudinal level (10m and 20m chord) from both wheels, alignment (10m and 20m chord) from both wheels, gauge (with and without load), and twist. It can be that some parameters have two sets of data from the right and left wheels. They will be used as features to develop predictive models.

For the track component defect inspection report, many defects are recorded as mentioned. The reports identify the type of defect, location of defects, and data of finding. From the reports, there are more than 81 types of track component defects recorded. The total number of samples is more than 1,900. However, a sample in track component defect inspection reports may be related to more than one section of track geometry parameters because a track component defect can cover multiple sections of tracks. Some examples of track component defects are damaged frogs, broken welds, boltless joints, and broken rails. These defects will be grouped to train the machine learning models otherwise the samples are too scattered and machine learning

models will not be able to learn. The group of track component defects is based on track components consisting of rail, switch and crossing, fastener, and rail joint.

For the track design, two parameters are obtained which are the radius of curvature and the gradient of the tracks. These data can be extracted from the design and data extraction is based on foot-by-foot data from track geometer parameter data.

Three sets of data will be combined together and processed as features for machine learning models. For track geometry parameters, one set of track geometry parameters of each section may not be sufficient for predicting track component defects. Therefore, two sets of data from two ends of each section are used as representatives of the track geometry parameters of each section. In conclusion, 24 track geometry parameters are used per section. In addition, differences between the two ends are also used as additional 12 features. Another additional feature is a binary feature indicating that the track section is a tangent or curve track. Therefore, the total number of features for training machine learning models is 37. Data from defect inspection reports are used as labels after being grouped as mentioned. To summarize, data from track geometry parameters and track design are used as features while data from defect inspection reports are used as labels. In total, there are almost 200,000 samples for training machine learning models. Samples are split using the same proportion as the previous section that is 70% are used as training data and 30% are used as testing data. It is worth noting that although the number of groups of component defect is four, an additional group is added which is no defect group.

5.1.3.3 Methodology

In this section, machine learning techniques that are used are MR, DNN, CNN, SVM, DT, RF, and GB. Hyperparameter tuning is used to find the best combination of hyperparameters for each model to ensure the model's performance. Two popular hyperparameter tuning techniques are grid search and random search. Grid search creates the possible combination of hyperparameters according to the range of each hyperparameter defined by humans. Therefore, the range of searching is pre-defined and constant. Random search is different. Random search is done by machine by randomly selecting combinations of hyperparameters within the ranges defined by humans. Therefore, grid search is more predictable and required less time to tune hyperparameters. However, random search has the benefit of exploring feasibilities that may not be considered by humans. In this section, grid search is used for hyperparameter tuning. The list of hyperparameters that will be tuned in this section is shown in Table 5-19.

Table 5-19 List of hyperparameters tuned for predicting track component defects

Model	Tunes hyperparameters	
DNN	Number of hidden layers Number of hidden nodes Activation functions Optimizer	Learning rate Momentum Batch size
CNN	Number of convolutional layers Filter Kernel Number of max pooling layers Pool size	Activation functions Number of hidden layers Number of hidden nodes Optimizer Batch size
MR	Feature selection Min tolerance	Ridge
SVM	Kernel type Kernel cache	Convergence epsilon
GB	Number of trees	Min split improvement

Model	Tunes hyperparameters	
	Maximum depth	
DT	Maximal depth Confidence	Minimal leaf size Minimal size for split
RF	Number of trees Maximal depth	Voting strategy

To evaluate the performances of the developed machine learning models, accuracy, precision, and recall are used. The calculation can be done using Equation 3 to Equation 5.

5.1.3.4 Results and discussion

From the model training, the accuracies of the developed machine learning are shown in Figure 5-17. From the figure, the DNN and CNN models have similar performance while the DNN model performs slightly better. The accuracies of both models are 0.943 and 0.938 respectively. in contrast, the accuracies of other models are about 0.5 or lower. The SVM model has the lowest accuracy. It can be inferred that the SVM technique has a concept to create hyperplanes in samples. Then, the samples are classified using these hyperplanes. The number of hyperplanes is equal to the number of features. In this case, the number of features is 37 so the number of hyperplanes is this value as well. When the number of features is increasing, SVM tends to be suffered from this increasing number because the model will be complicated and the performance will drop. At the same time, the SVM model will also suffer from the long training time. In this case, it is not surprising that the SVM model has the lowest accuracy. For the MR, DT, RF, and GB, it can be inferred that they are likely linear models. However, the problem in this section is non-linear and complicated. Therefore, these models do not perform well in detecting

hidden patterns in non-linear data. Differently, the DNN and CNN models can better handle the non-linearity and that is a reason why their performances are better than other models.

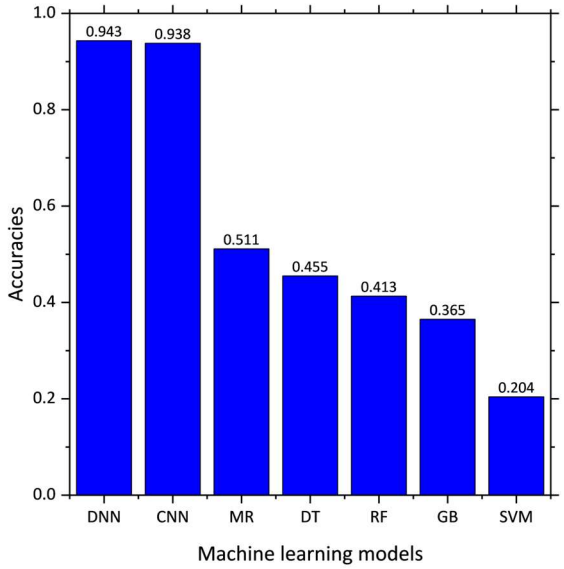


Figure 5-17 Performances of machine learning models on predicting track component defects

For the precisions and recalls, the outcomes of the DNN and CNN models are presented in Table 5-20. Other model values are not shown because the performances are relatively low compared to the DNN and CNN models. It can be seen that both models have high values of precisions and recalls which demonstrate the high performances of the models in terms of the ability to predict and the reliability of the prediction. Most values are about 0.90 or higher which is satisfying.

Table 5-20 Precisions and recalls of the DNN and CNN models on predicting track component defects

Class	Precision		Recall	
	DNN	CNN	DNN	CNN
Switch and crossing	0.94	0.91	0.90	0.87
Fastener	0.86	0.91	0.90	0.86
Rail joint	0.96	0.93	0.93	0.93
Rail	0.96	0.93	0.94	0.97
None (Defect-free)	0.93	0.96	0.96	0.93

From the model tuning, the optimized sets of hyperparameters of each machine learning model are shown in Table 5-21. Tuned hyperparameters for the predictive model to classify dipped joint and settlement severity

Table 5-21 Tuned hyperparameters for the predictive model to detect track component defects

Model	Tuned hyperparameter	Value
DNN	Number of hidden layers	2
	Number of hidden nodes	750 (dense1), 1,750 (dense2)
	Activation functions	ReLu
	Optimizer	Adam
	Learning rate	0.001
	Momentum	0.8
	Batch size	128
CNN	Number of convolutional layers	2
	Filter	128 (conv1), 256 (conv2)
	Kernel	4 (conv1), 6 (conv2)
	Number of max pooling layers	2 (after each conv)
	Pool size	1
	Activation functions	ReLu
	Number of hidden layers	2
	Number of hidden nodes	400 (dense1), 200 (dense2)
	Optimizer	Adam
	Batch size	128
MR	Feature selection	M5 prime
	Min tolerance	0.05
	Ridge	1E-8
SVM	Kernel type	Dot
	Kernel cache	200
	Convergence epsilon	0.001
GB	Number of trees	50
	Maximum depth	5
	Min split improvement	1E-5
DT	Maximal depth	10
	Confidence	0.1
	Minimal leaf size	2
	Minimal size for split	4

Model	Tuned hyperparameter	Value
RF	Number of trees Maximal depth Voting strategy	100 10 Confidence vote

The results show that the track geometry parameters can be used to predict and detect track component defects. This finding will help railway operators save a lot of time from inspection and improve the efficiency of railway predictive maintenance. In addition, they will gain the benefit from improved safety because their staff does not need to be on-site frequently. The data used to predict track component defects are track geometry parameters that should be available for most railway operators. Therefore, it can save the cost as well and the data collection can be done at a higher speed when visual inspection.

5.1.4 Unsupported sleepers and the number of unsupported sleeper prediction

5.1.4.1 Introduction and backgrounds

In the railway industry, two main railway track infrastructures are ballast track and ballastless track. Ballast track is a traditional track infrastructure that has been used for a long time. Railway sleepers lay on the ballast that is used as a substructure and used to distribute loads from rolling stock to the ground. The advantages of a ballast track are low construction cost, ease of construction, good drainage performance, low noise and vibration, and ease of maintenance. However, the ballast track deteriorates faster than the ballastless track so it requires more maintenance and has a shorter service life. In addition, it usually creates pollution from the dust occurring from ballast deterioration and the speed of rolling stock is more limited. Ballastless track, as its name, does not have the ballast that supports sleepers but sleepers are cast using

concrete. Compared to a ballast track, a ballastless track is more advanced. The advantages of a ballastless track are it requires less maintenance cost, causes less dust, supports higher speed than a ballast track, has a longer service life, and provides more passenger comfort if the condition is good. However, the disadvantages of a ballastless track are higher construction cost, it is suitable for some site conditions such as soft soil or earthquake zones, and the maintenance can be done with difficulty.

As mentioned, the ballast track tends to deteriorate faster and easier. The deterioration results in poor track geometry, poor passenger comfort and safety, and higher maintenance costs. The mechanism of ballast deterioration starts from the occurrence of voids under sleepers [29]. These voids result in higher vibration and impact loads. Then, the settlement of the track structure takes place because of these voids. However, sleepers will not settle or move down along with the substructure because they are fastened with tracks. Therefore, sleepers are not supported by ballast. This phenomenon is called an unsupported sleeper or hanging sleeper. From this, the whole railway infrastructure will deteriorate even faster because the load applied to the railway infrastructure is higher due to lower load distribution. If incidences are severe, they can even cause a derailment.

Unsupported sleepers can be detected using onboard methods or track-side methods. Examples of onboard methods are the Swiss track stiffness measurement vehicle and the Swedish railway rolling stiffness measurement vehicle [130]. It can be seen that the onboard methods are done using specific vehicles. This method is widely used in different countries such as China and USA. A disadvantage of this technique is the cost of the vehicle is high. Some railway operators choose

to not purchase these vehicles due to the cost but they rent the measurement vehicles instead. Therefore, the inspection of unsupported sleepers cannot be conducted frequently. In addition, onboard methods also suffer from limited speed and accuracy [29]. Examples of track-side methods are accelerometers, cameras, track deflection techniques [131], geophone, and laser arrays [132]. A disadvantage of track-side methods is the costs of equipment, installation, and software are expensive.

This section aims to combine the benefit of both on-board and track-side methods by using accelerometers installed at axle boxes to measure ABAs. ABAs can be collected on board with the speed of regular operations and the cost of equipment is not high. Once sensors are installed, data measurement can be done more frequently because there are no additional costs. Condition monitoring can be done regularly so that severe defects can be avoided. The overall maintenance cost and efficiency can be improved. ABAs will be used to train machine learning models to detect supported sleepers in this section. At the same time, the severity of unsupported sleepers will be classified using machine learning as well.

The machine learning techniques that will be used to develop predictive models in this section are CNN, LSTM, ResNet, and FCN. Data used to train the machine learning models are simulated using FEM models which are validated with field data. In this section, the frequency of analysis from FEM models is constant. Therefore, the shapes of the data will be varied depending on the speed of the rolling stock. The faster-rolling stocks will generate a shorter shape of data because the time used to operate the rolling stocks on the track section is shorter. Therefore, data reshaping will be done for CNN, LSTM, and ResNet. However, data reshaping is not required for

the FCN model because the architecture of the model support varied shape of data. Data processing techniques used for reshaping are FFT and padding.

The development of unsupported sleepers is caused by track substructure or ballast that deteriorates due to the operation. Then the gaps or voids occur under sleepers but sleepers will not settle down with ballast because they are fixed with tracks. Therefore, some sleepers are unsupported or hung [133]. Augustin et al. [134] did a study and stated that about 50% of sleepers were not fully supported. Zhu et al. [133] focused on the dynamic behaviour of the track under the unsupported sleeper condition. In their study, they modelled a physical model with a scale of 1:5. They varied the speed of rolling stocks from 160 to 320 km/h and the number of unsupported sleepers from 0 to 3. They found that unsupported sleepers created an additional impact on the railway infrastructure.

Mosayebi et al. [135] used FEM models to simulate the unsupported sleeper phenomena. They found that unsupported sleepers created additional force and displacement to the railway infrastructure. They tried to find the relationship between the track velocity and stiffness of the track and they found that the relationship was non-linear. They found that when a sleeper was unsupported, the displacement increased by 90% and 190% when more than two sleepers were unsupported. At the same time, the tracks were applied by forces increasing more than 1.4 and 1.78 times when one and two sleepers were unsupported. Moreover, the vibration was increased more than by 40% when there were some unsupported sleepers. Many studies drew the same conclusion as this study [136-138].

Because sleepers are common components in the railway infrastructure, they were interested in many academics. Besides unsupported sleepers, sleeper defects were aimed to be detected. Different techniques were used such as image processing [25], modeling [26], and computer vision [27, 28].

To detect unsupported sleepers, many approaches have been used. The simplest approach was measuring the gap between sleepers and ballast done by Kim et al. [139]. It was straightforward and could identify unsupported sleepers immediately. In addition, they found that unsupported sleepers resulted in lower track stiffness. They found that the stiffness of the track when unsupported sleepers occurred was only 11.6% compared to fully-supported sleepers. Displacements were also used by Sysyn et al. [29] to detect unsupported sleepers. In their study, they recorded video to collect data and applied machine learning models to detect unsupported sleepers. They applied SVM and KNN to develop predictive models. However, the performances of their model were not very desirable because the accuracies of those two models were 58% and 65% respectively. They clearly stated that the performance of the models could be improved and one technique they suggested was using more advanced machine learning techniques.

It can be seen that unsupported sleepers were normally detected using the additional installation of sensors or devices. This section aims to develop machine learning models to detect unsupported sleepers which use data that is easy to measure. In this case, ABAs will be used as features. Although machine learning was tried to detect unsupported sleepers, the performances were not satisfying, and modern machine learning techniques were not tried. In addition, the severity has not been predicted using machine learning. This section will explore the potential of

machine learning in this task. It is expected that the developed machine learning models can detect and classify the severity of unsupported sleepers. Railway operators will be able to diagnose unsupported sleepers better using the developed approach.

5.1.4.2 Methodology

This section will use data from FEM models. The FEM models are developed based on the FEM model developed by Li et al. [140]. The FEM model was developed to study the interaction between rolling stocks and railway infrastructure. The rolling stock model was developed using the concept of MBS. LS-DYNA was used to develop the FEM model. The FEM model is shown in Figure 5-18. The rolling stock was modelled consisting of a car body supported by two bogies. Each bogy was supported by two wheelsets and each wheelset contained primary and secondary suspension systems. The track was modelled using the FEM concept. On the top, two rails were modelled using Euler beams. Rails were supported by rail pads at the location where contact with sleepers. Rail pads were modelled as a series of springs and dampers. Sleepers were modelled as Euler beams. Last, ballast supported sleepers when springs and dampers were used to mimic the behaviour of ballast. The details are shown in Figure 5-19.

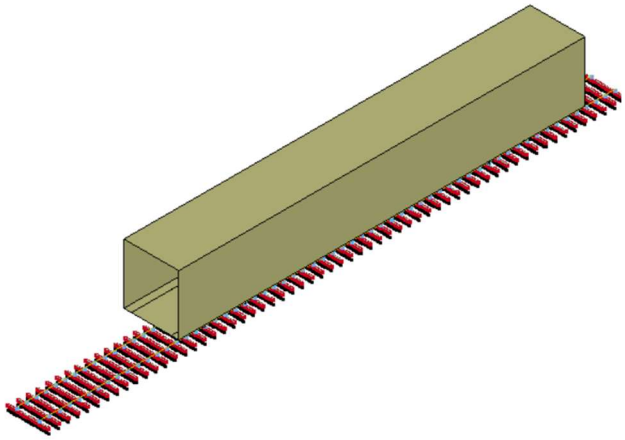


Figure 5-18 FEM model for studying the interaction between rolling stocks and railway infrastructure

The interaction between wheel and rail was modelled using the keyword "*Rail_Track and *Rail_Train" in LS-DYNA. The interaction was simulated using Equation 6 where F was wheel-rail contact force, K was contact stiffness, Z_w was wheel vertical displacement, Z_r was rail vertical displacement, and δ was track irregularity.

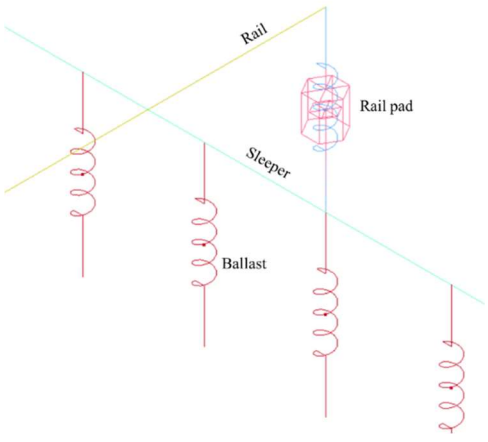


Figure 5-19 Detailed models of railway infrastructure

$$F = K \times (Z_w - Z_r - \delta)$$ Equation 6

The model was modelled based on the real situation and field data. Therefore, the irregularity used in this model was based on field data as shown in Figure 5-20. Springs and dampers used in the model were simulated using keywords "S01-SPRING_ELASTIC" and "S02-DAMPER_VISCOUS" respectively. To mimic the behaviour of unsupported sleepers, springs related to ballast are removed from the model. The number of unsupported sleepers used in this section is ranged from one to four. The range of rolling stock speed is 100 to 200 km/h. Wheel-rail contact stiffness is varied to simulate the real situation.

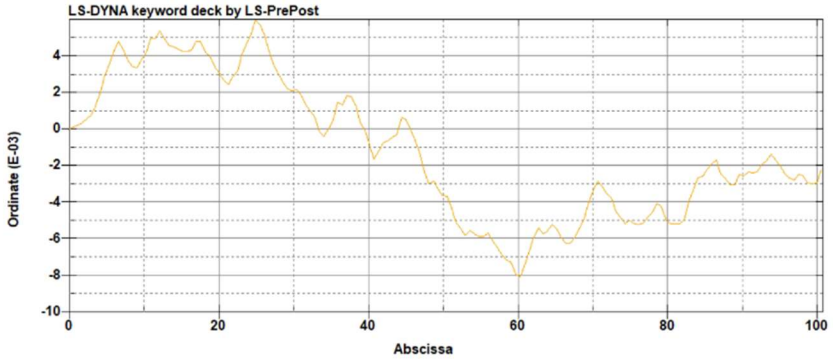


Figure 5-20 Track irregularity from field data

Field data collected by Cai et al. [141] and Ngamkhanong et al. [142] were used to verify the FEM model. The field data were collected from the Sui-Yu Railway. The rolling stocks' speed in that section is 160 to 220 km/h. The FEM model validation was done by comparing wheel-rail contact force, displacements of rails, and displacement of sleepers. The comparison of different parameters from field data and the FEM model are shown in Table 5-22. From the comparison,

the differences of parameters from field data and the FEM model are close. The differences are less than 10%. These results validate that the FEM model can be used as a representative of the field data.

Table 5-22 The comparison of parameters from field data and the FEM model for unsupported sleepers

Parameters	Field data [141, 142]	FEM model	Differences
Wheel-rail contact force (kN)	100	98.4	1.60%
Rail displacement at rail seat (mm)	2.606	2.596	0.38%
Rail displacement at mid span (mm)	2.604	2.415	7.26%
Sleeper displacement at rail seat (mm)	2.576	2.522	2.10%
Sleeper displacement at mid span (mm)	2.511	2.352	6.33%

The machine learning models in this section are developed using CNN, LSTM, ResNet, and FCN. As mentioned, FCN is the only technique in this section that the data shapes can be varied. This is because the FCN architecture has a component called a global pooling layer that supports different shapes of input data. In addition, dense layers are not used but a 1*1 convolutional layer is used instead for classification. ResNet is an advanced machine learning model developed based on CNN as well as FCN. The main outstanding characteristic of ResNet is it contains skip connections to allow data to be passed through some convolutional layers when the model detects those layers are not useful. This is because the nature of the ResNet model contains many convolution and pooling layers. If data are processed through every layer, the performance of the model might not be satisfying or the training time is too long. Skip connections allow the model to use different layers as it is necessary.

In this section, machine learning models are developed for two purposes which are unsupported sleeper detection and unsupported sleeper severity classification. 70% of samples are used to train the models while the rest 30% are used to test the models. Grid search is used for hyperparameter tuning. The list of tuned hyperparameters is shown in Table 5-23. The performances of the machine learning models are evaluated using accuracies.

Table 5-23 List of hyperparameters for predicting unsupported sleepers

Model	Tuned hyperparameters	
CNN	Number of convolutional layers Filter Kernel size Number of pooling layers Pool size	Activation function Batch size Optimizer Number of hidden layers Number of hidden nodes
LSTM	Number of LSTM cells Window size Activation function Batch size	Optimizer Number of hidden layers Number of hidden nodes
ResNet	Number of residual blocks Number of convolutional layers Filter Kernel size Number of pooling layers Pool size	Activation function Batch size Optimizer Number of hidden layers Number of hidden nodes
FCN	Number of convolutional layers Filter Kernel size Number of pooling layers Pool size	Activation function Batch size Optimizer Number of hidden layers Number of hidden nodes

5.1.4.3 Data and processing

Data used to train machine learning models are ABAs obtained from FEM models. ABAs are from two front wheels of rolling stocks. Different parameters are varied to mimic real situations. The

frequency of the output from FEM models is set to 1,000 Hz. A variety of parameters is shown in Table 5-24. The simulated track section has a length of 35 m and the spacing of sleepers is 0.6 m. The total number of sleepers in the section is 60. Unsupported sleepers are randomly located from the first location to the last location. However, the unsupported sleepers will be located adjacently if there is more than one unsupported sleeper. Other varied parameters are shown in the table. The total number of simulations is 2,016 which will be used to train machine learning models.

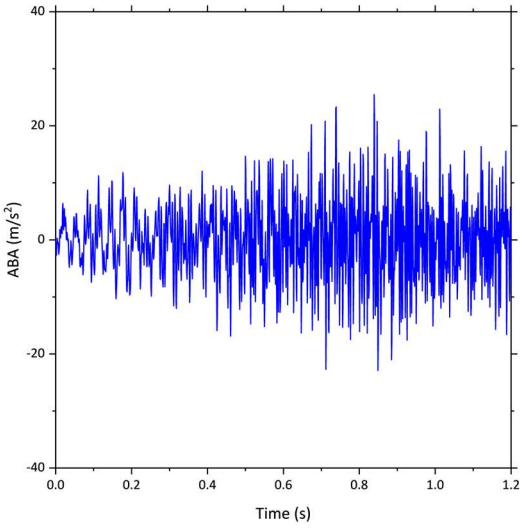
Table 5-24 Parameter variation for unsupported sleeper conditions

Parameters	Range
Number of unsupported sleepers	1-4
Location of unsupported sleepers	1-60
Speed of the rolling stock (km/h)	100-200
Size of irregularity (%)	98-101
Vertical stiffness of wheel-rail contact (N/m)	1.06×10^{9}-1.325×10^{9}

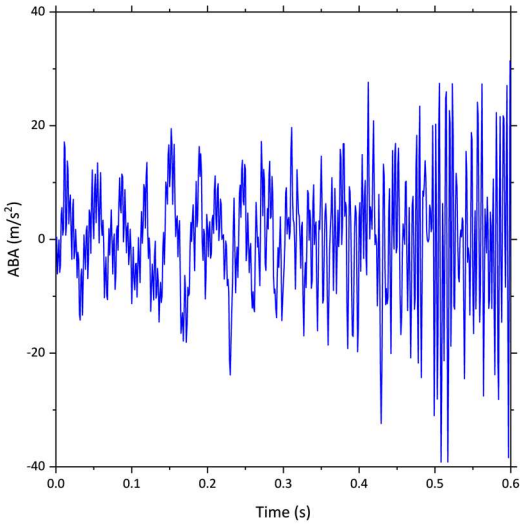
As mentioned, the length of the track section and the frequency of output are fixed. The shapes of data will vary based on the speed of rolling stocks. The data shapes will be shorter when rolling stocks run faster because they take less time to run on the track. Examples are shown in Figure 5-21. It can be seen that the length of data is twice shorter when the speed of rolling stocks is twice faster.

In this section, CNN, LSTM, ResNet, and FCN will be used to develop machine learning models. Except for the FCN model, other models require data with the same shape to train and deploy. Therefore, data processing is required to make the data shape consistent. Two techniques that

will be used are padding and FFT. Padding is a simple technique that is shown in Figure 3-6 and Figure 3-12. In this case, ABAs are one-dimension data so zeros will be added in a row only. The number of added zeros depends on the longest data length or the data from the slowest rolling stocks (100 km/h). FFT is a method to process time-domain data to frequency-domain data as presented in 5.1.1 and Figure 5-3. Processed data from Figure 5-21 using FFT are shown in Figure 5-22.

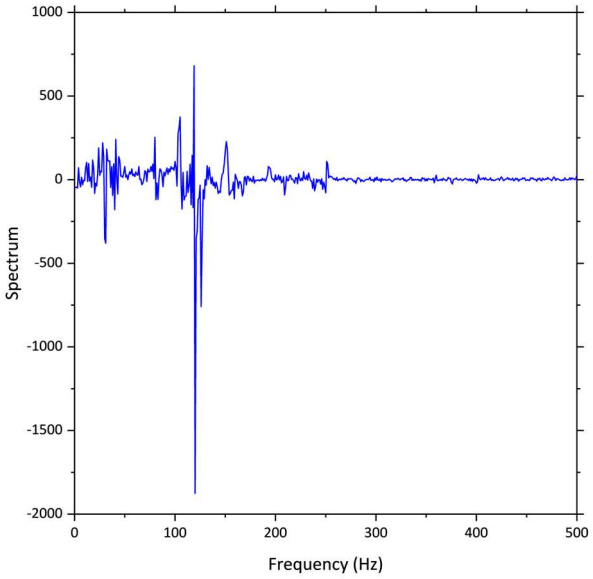


(a)

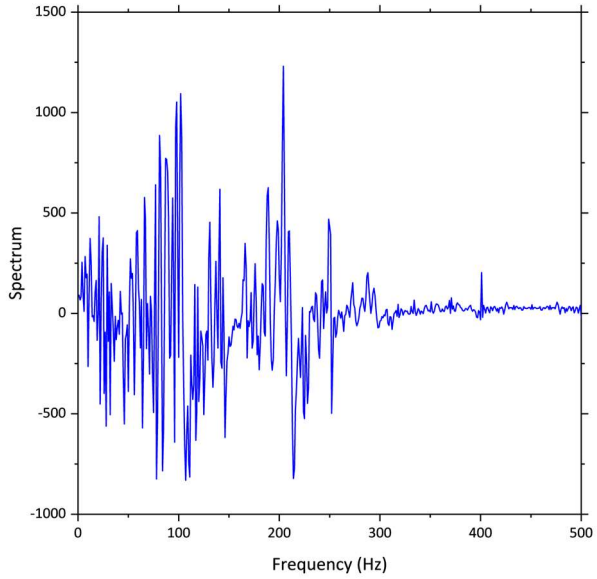


(b)

Figure 5-21 Examples of ABAs for unsupported sleeper analysis (a) the speed is 100 km/h and (b) the speed is 200 km/h



(a)



(b)

Figure 5-22 Examples of data processed using FFT (a) the speed is 100 km/h and (b) the speed is 200 km/h

5.1.4.4 Results and discussion

For the unsupported sleeper detection, the accuracies of machine learning models are shown in Table 5-25.

Table 5-25 Accuracies on unsupported sleeper detection

Models	Accuracies (%)		
	FFT	Padding	Raw data
CNN	99.17	100.00	-
LSTM	98.18	80.17	-
ResNet	83.64	93.39	-
FCN	-	-	95.21

From the table, every model performs well in terms of detection. Every accuracy is higher than 80%. The CNN model performs better than any other model when the accuracies are the highest. For the CNN model, data processing using the padding technique provides better performance. However, it cannot be concluded that padding is better than FFT in every case. For the LSTM model, FFT processing provides better performance than padding. For the FCN model that used the raw data to predict, the performance is also satisfying which is higher than 95%. In this case, it can be said that the ResNet model is the most complicated one and the training time is the longest. However, it cannot guarantee that its performance will be the best based on the results shown in the table. For the training time, the training times from the longest are 46, 6, 2, and 1 seconds/epoch for ResNet, LSTM, FCN, and CNN respectively. Compared to the previous study that the accuracy was 65%, and the performances of the machine learning models developed in this section are significantly better. The tuned hyperparameters of the CNN model are shown in Table 5-26.

Table 5-26 Tuned parameters for unsupported sleeper detection

Tuned hyperparameters	Values
Number of convolutional layers	2
Filter	64 (conv1), 32 (conv2)
Kernel size	7
Number of pooling layers	2
Pool size	2
Activation function	ReLu
Batch size	64
Optimizer	Adam
Number of hidden layers	2
Number of hidden nodes	100

For the unsupported sleeper severity classification, the number of unsupported sleepers is ranged from one to four. Therefore, there are four classes of labels. The accuracies of the models are shown in Table 5-27.

Table 5-27 Accuracies on unsupported sleeper severity classification

Models	Accuracies (%)		
	FFT	Padding	Raw data
CNN	90.28	92.89	-
LSTM	71.56	33.89	-
ResNet	51.18	92.42	-
FCN	-	-	81.28

Compared to the detection, the accuracies are lower which can be inferred that the number of classes to predict is higher. Therefore, the accuracies are lower. However, the overall performance is satisfying. The highest accuracy is higher than 90%. The CNN model also performs the best in classification. In this case, ResNet with padding data has the second-best performance while the LSTM has the lowest accuracy when used with padding data. The FCN model's performance is acceptable even when used with raw data. It is to remark that the performances of the LSTM and ResNet models drop significantly when data are processed unsuitably. It can be concluded that these two models are sensitive to data. As the detection, the ResNet model requires the longest training time. The tuned hyperparameters of the CNN model are shown in Table 5-28.

Table 5-28 Tuned parameters for unsupported sleeper severity classification

Tuned hyperparameters	Values
Number of convolutional layers	2
Filter	64 (conv1), 32 (conv2)
Kernel size	5 (conv1), 7 (conv2)
Number of pooling layers	3
Pool size	3
Activation function	ReLu
Batch size	8
Optimizer	Adam
Number of hidden layers	2
Number of hidden nodes	100 (dense1), 50 (dense2)

From the machine learning model development, the developed models have the potential to detect and classify the severity of unsupported sleepers. Railway operators will get benefit from the models or the development concepts to apply to their responsible railway system. The data collection can be done with ease and small time consumption because ABAs are used to predict. This will improve the efficiency of predictive maintenance of the railway infrastructure because the monitoring can be done regularly. The severity of defects will not be too high and defects will require low maintenance costs.

5.1.5 Turnout support condition evaluation under flooding

5.1.5.1 Introduction and backgrounds

Railway infrastructure consists of many components to allow the railway system can function as it is designed. This section will focus on turnout support. Turnouts are important railway infrastructure that is used to guide rolling stocks when they need to be steered from one track

to another track. Normally, turnouts are installed at railway junctions, siding tracks, or any locations where rolling stocks need to change to run on another track. Turnouts are the components that usually move laterally because of their function. There are many components in a turnout such as switch rails, stock rails, point blades, crossing noses, guard rails, and frogs. To steer rolling stocks, turnouts have to lock before allowing rolling stocks to change directions. To perform this function, turnouts are applied by high loads and impacts. In addition, they are movable parts so the deterioration is high in terms of turnouts themselves and their underneath supports [143-145]. Therefore, turnouts can be considered a weak component of the railway infrastructure. Without other incidences, turnouts and their supports rapidly deteriorate but some extreme events can accelerate the deterioration of turnouts and their support. An example of these extreme events is flooding.

This section will focus on the deterioration of turnout support under flooding conditions. Floodings are one of the causes that damage the railway system around the world and there are more than 100 incidences that damage to the railway system caused by flooding [146]. As mentioned, turnouts are important components of the railway infrastructure, if they are damaged and cannot function, it will cause delays because of the safety issue and maintenance [147]. In addition, if severe deterioration is not detected, it can cause a derailment. Xin et al. [148] collected data in the Netherlands and found that railway operators needed to repair turnouts in more than 400 locations every year. Some turnouts needed to be repaired immediately within a week because of the function and safety issues. The maintenance cost related to turnouts was more than 6.4 million euros per year. These damages have not been

caused by the turnouts only, but also by the turnout support. This section will try to predict the deterioration of turnout supports in the event of flooding. Machine learning capability will be used to develop a predictive model when features that are used to train the machine learning model are ABAs and crossing nose displacements. The benefits of using ABAs were mentioned in previous sections. This section additionally includes displacements of crossing noses because they might be directly related to the deterioration of turnout supports. Different from previous sections, the machine learning model in this section is not used to detect defects but to evaluate the current conditions of railway infrastructure. This will be beneficial in terms of conditional monitoring for railway operators to plan maintenance. Severe damage can be avoided because monitoring can be done regularly.

Floodings are one of the common incidences in the railway industry however they are not prioritized as much as they should be [146]. Floodings normally happen and the railway system cannot avoid handling them. Railway operators can use different approaches to deal with flooding according to the risk management concept. One approach that can be used in evaluating the effect of flooding on the railway infrastructure. In many cases, flooding events are not severe and the railway infrastructure is not damaged by them. railway operators might not need to take any action. However, the capability to evaluate the condition of the railway infrastructure is required. In this section, turnout supports are focused on. Machine learning is used to develop the predictive model for evaluating the condition of turnout supports.

Hasnayn et al. [149] studied the effect of flooding on subgrade conditions in the events of flooding and after the events. They used the track stiffness as a key parameter. They found that

track stiffness was affected and could be used as an indicator to monitor the track condition. They stated that the track stiffness was affected because of the water content and settlement due to flooding events. They suggested that railway support should be investigated further. Hamarat et al. [150] conducted a further study by focusing on turnout support under flooding conditions. They found that flooding could negatively affect track geometry and track defects. In their study, they developed a FEM model to mimic the dynamic behaviour of rolling stock and railway infrastructure. Examples of their findings were superelevation and twist were suffering from floodings severely.

To monitor and evaluate the condition of turnout supports, many studies were conducted. Li and Berggren [151] tried to find the relationship between track stiffness and track performance. They found that the track stiffness could be used to evaluate the track performance in terms of track component accelerations, moments, and wheel-rail contact forces. From their study, different parameters can be used to indicate the condition of turnout support condition. This section will apply the finding from that study by using track stiffness as an indicator for monitoring and evaluating.

Kaewunruen [152] applied maximum accelerations to evaluate the deterioration of turnouts and found that the results were satisfying. Other studies also confirm their findings [153]. A technique that can be used to evaluate the turnout condition is machine learning. Sysyn et al. [154] applied machine learning to detect turnout defects. They used ABAs as features and got an accuracy of 90%. The potential of machine learning in detecting defects in the railway system is also presented in this thesis.

As mentioned in the previous sections, data measurement in the railway industry can be mainly categorized into two groups which are on-board measurement and track-side measurement. Advantages and disadvantages were described in Section 5.1.4.1. Based on the same reasons, this section will use ABAs as the main features to develop the machine learning model for evaluating turnout support conditions. In addition, crossing nose displacement is also used because it is an important component of turnout that can present in the condition of turnout support.

5.1.5.2 Methodology

FEM models are used to generate numerical data and mimic the dynamic behaviour of turnout supports under flooding conditions. FEM models are developed based on the FEM model developed by Hamarat et al. [155] who applied LS-DYNA to develop the track section containing a turnout. The FEM model is shown in Figure 5-23. The rolling stock is modelled using the MBS concept while railway infrastructure is modelled using the FEM approach which is similar to the FEM model used in Section 5.1.4. The turnout is a 1:9 turnout. In this section, additional boundaries are defined to make the FEM model reliable under flooding events. For example, the bottom of the ballast is defined as not having vertical movement and the ends of the rails do not have movement in any direction.

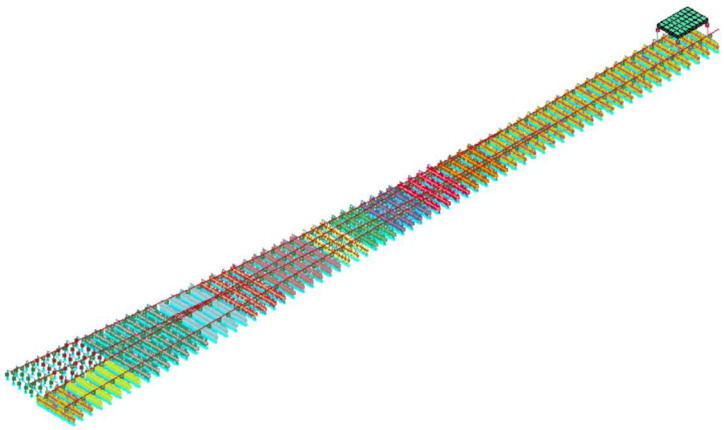


Figure 5-23 FEM model for turnout support under flooding condition study [155]

Model verification is done by comparing different parameters from the outputs of the FEM model to the field data. Field data used in this section are available as open-source data [156, 157]. The main parameters used to compare are accelerations of crossing noses. The accelerations of the crossing nose from the field data and the FEM model are 214.05 m/s^2 and 201.6 m/s^2. The difference between these two values is 5.8% which is acceptable and it can be concluded that the FEM model can be used to represent the field data.

To imitate the flooding conditions, ballast stiffness and damping coefficient are adjusted. These two values will be varied based on the different levels of water under flooding events. In this section, the water level is defined to be 0% and 114% as shown in Table 5-29. At the same time, the classes for the predictive model are also defined based on ballast stiffness and damping coefficient. Weights and speeds of rolling stocks are also varied to represent the situations in

which rolling stocks will be allowed to operate under different speeds and loads under different levels of flooding.

Table 5-29 Different parameters based on different levels of water under flooding events

Parameters	Units	Ranges	Class
Stiffness	MN/m	More than 12.8	1
		10.5-12.8	2
		8.1-10.5	3
		Less than 8.1	4
Damping coefficient	kNs/m	Less than 1.2	1
		1.2-1.5	2
		1.5-2.0	3
		More than 2.0	4
Weights of rolling stocks	Tonnes	32-48	n/a
Speeds of rolling stocks	Km/h	135-225	n/a

Examples of ABAs and crossing nose displacement which will be used to train the machine learning model are shown in Figure 5-24 and Figure 5-25. From the figures, the speed of rolling stock is set to 135 km/h when the weight is 46.4 tonnes. The stiffness and damping coefficient of turnout support is 14.6 MN/m and 1.2 kNs/m which is there is no flooding. It can be seen that when the rolling stock operated through the turnout, the ABAs increase obviously. At the same time, the crossing nose is easier to detect because there is no displacement if the rolling stock is not operated through the turnout. In addition, when the rolling stock is completely through the crossing nose, the displacement will be back to zero or no displacement. From the previous sections, the CNN model performs best in many situations. Therefore, this section will focus on the CNN model to develop the predictive model.

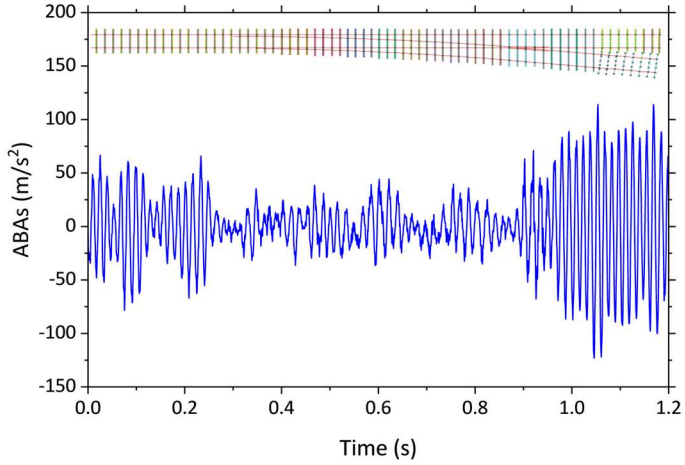


Figure 5-24 Example of ABA under flooding events

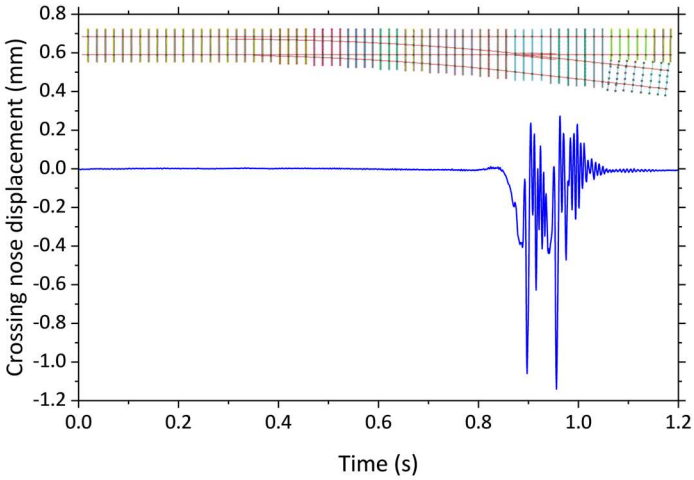


Figure 5-25 Example of crossing nose displacement under flooding events

Grid search is used for hyperparameter tuning. The list of hyperparameters of the CNN model which are tuned is shown in Table 5-30. The indicator used to evaluate the performance of the model is the accuracy. Other indicators are used in corporate with accuracy to demonstrate the performance of the developed machine learning model such as precision and recall.

Table 5-30 List of hyperparameters tuned for classifying the turnout support deterioration under flooding events

Hyperparameters	
Number of convolutional layers	Activation function
Filter	Batch size
Kernel size	Optimizer
Number of pooling layers	Number of hidden layers
Dropout	Number of hidden nodes
Pool size	

5.1.5.3 Data and processing

From the FEM model simulations, the total number of samples is 1,936. There are three sets of data used in a sample. The first two are two sets of ABAs from the left and right wheels of rolling stocks. Another set of data is crossing nose displacement. The machine learning model is trained using three approaches. First, ABAs are used to train the model. Second, crossing nose displacements are used to train the model. Last, both ABAs and crossing nose displacements are used to train the model. Turnout support stiffnesses are the indicators for identifying the turnout support conditions. Track irregularity from the field data is also included in the simulations to imitate the real situation. The track section length is 45 m. The problem in this section is the classification when the number of classes is four as shown in Table 5-29. 70% of the data is used to train the model and another 30% is used to test the model. Data sampling is done using a stratified sampling technique.

5.1.5.4 Results and discussion

The machine learning model is developed to classify the turnout support conditions. The support conditions are defined as classes as shown in Table 5-29. The development approaches can be categorized into three approaches which are the machine learning model using ABAs, crossing nose displacement, and both. The performances of each approach and confusion matrices are shown in Table 5-31 to Table 5-33.

Table 5-31 Performance on turnout support condition classification under flooding events using ABAs

	Pred. class 1	Pred. class 2	Pred. class 3	Pred. class 4	Recall
True class 1	130	0	1	1	0.98
True class 2	8	125	7	0	0.89
True class 3	0	0	160	0	1.00
True class 4	0	0	0	149	1.00
Precision	0.94	1.00	0.95	0.99	0.97

Table 5-32 Performance on turnout support condition classification under flooding events using crossing nose displacement

	Pred. class 1	Pred. class 2	Pred. class 3	Pred. class 4	Recall
True class 1	142	7	0	0	0.95
True class 2	14	133	1	0	0.90
True class 3	0	13	131	2	0.90
True class 4	0	0	2	136	0.99
Precision	0.91	0.87	0.98	0.99	0.93

Table 5-33 Performance on turnout support condition classification under flooding events using ABAs and crossing nose displacement

	Pred. class 1	Pred. class 2	Pred. class 3	Pred. class 4	Recall
True class 1	127	2	1	0	0.98
True class 2	0	143	3	0	0.98
True class 3	1	1	148	0	0.99
True class 4	0	0	1	154	0.99
Precision	0.99	0.98	0.97	1.00	0.98

From the tables, the accuracy of every approach is satisfying when the accuracy is higher than 0.9 in every case. When ABAs and crossing nose displacement are used together, the accuracy is the highest. However, the performance is slightly improved when ABAs are used only. The crossing nose displacement approach provides the lowest accuracy. Although the use of both features provides the highest accuracy, railway operators have to consider the additional cost carefully because measuring crossing nose displacement requires additional installation of track-side sensors. The use of only ABAs may be sufficient because it provides the desirable outcome without additional installation of track-side equipment.

Precisions and recalls conform to the accuracies. The results from the use of both features are the best followed by the use of ABAs. All precisions and recalls are higher than 0.9 which demonstrate the potential of machine learning in evaluating turnout support condition under flooding events. As the accuracies, the use of two features provides the best outcome. However, railway operators have to consider carefully improving performance on prediction and additional costs for installing track-side equipment to measure crossing nose displacement. The cost of each technology varies. Railway operators can consider such laser-based equipment or cameras. Based on the nature of railway projects, they are long, and little additional cost can be enormous because the sensors have to be installed at all locations of the measurement. Therefore, the use of ABAs may be sufficient and efficient in terms of maintenance and cost.

From the hyperparameter tuning, the list of optimized hyperparameters is shown in Table 5-34.

Table 5-34 List of tuned hyperparameters tuned for classifying the turnout support deterioration under flooding events

Tuned hyperparameters	Values
Number of convolutional layers	2
Filter	64 (conv1) and 32 (conv2)
Kernel size	1
Number of pooling layers	0
Pool size	N/A
Dropout	0.5 (conv2)
Activation function	ReLu
Batch size	64
Optimizer	Adam
Number of hidden layers	2
Number of hidden nodes	100

The findings in this section show that the developed model has the potential to evaluate the turnout support conditions under flooding events. This will be beneficial for railway operators to evaluate the current conditions of turnout support under flooding events. The application is not limited to flooding events only, the model can be used in other scenarios to evaluate the turnout support conditions. This will allow railway operators to plan maintenance more efficiently. If the conditions of turnout support are in the acceptable range, maintenance can be postponed until the values reach the trigger level. Therefore, maintenance can be limited as it is necessary and this will save maintenance costs for the railway system.

5.2 Regression problems in railway maintenance

5.2.1 Wheel burn localization

5.2.1.1 Introduction and backgrounds

This section presents the application of machine learning to localize defects. In this section, wheel burns are used as case studies. Wheel burns are one of the common rail defects with a proportion of 13-17% of total railway defects [158]. Wheel burns emerge when there are slips or slides between wheels and rails. These are the results of insufficient adhesion or traction between wheels and rails. The damages mainly take place at rails. In severe cases, these defects can cause derailments. Other causes of wheel burns are too steep grade, improper rolling stock driving, material contamination, and inadequate locomotive power.

The severities of wheel burns vary depending on many factors. Small wheel burns can be maintained by rail grinding while severe wheel burns have to be treated by rail replacement which the maintenance cost is high [159].

Different techniques are used to detect wheel burns such as acoustic [10], laser [11], computer vision [14], and machine learning [160]. In previous sections, machine learning is used to detect and evaluate the severity of defects. However, studies related to defect localization are limited while defect localization is important to the railway industry because the characteristic of projects is longitudinal. A study about defect localization in the railway industry has not been done. Some studies related to defect localization in other industries are a study by Poyhonen et al. [63] and Ferguson et al. [64] who used machine learning to localize insulation defects and

casting defects respectively. Both of them found that the performance of machine learning models was satisfying. Therefore, this section applies machine learning to localize defects when selected defects are wheel burns.

Machine learning techniques that will be used to develop predictive models are DNN, CNN, and LSTM. Data are numerical data from D-Track and the features are ABAs.

5.2.1.2 Data and processing

Data are simulated using D-Track. The sizes and locations of wheel burns are varied to create data diversity to mimic real situations. The speeds and weights of rolling stocks are varied. The data diversity is shown in Table 5-35. The sizes of wheel burns are based on the manual by the Australian Rail Track Corporation [161] which stated that the common sizes of wheel burns are ranged between 25 to 75 mm. In this section, the length of the track section is 20 m so the wheelflats are located within this range. The total number of samples is 1,620. Data splitting for training and testing data is conducted with a proportion of 70/30. Results from D-Track that are used to train the machine learning models are ABAs.

Table 5-35 Parameter variation for wheel burn localization

Parameters	Variation
Sizes of wheel burns	25, 50, and 75 mm
Locations of wheel burns	0-20 m
Weight of rolling stocks	40, 60, and 80 tonnes
Speed of rolling stocks	20-200 km/h

5.2.1.3 Methodology

This section uses DNN, CNN, and LSTM to develop the machine learning model to localize wheel burns. The problems in this section are categorized as regression problems because the labels are continuous range. The features are raw data which are ABAs. For the DNN model, raw data are also used. It can be done by defining the number of nodes in the input layer to the same shape as raw data. The number of nodes in the output layer is one for every model because of the regression problems. This is different from classification problems when the number of nodes in the output layer is equal to the number of classes. The detail of the machine learning techniques was presented in Section 5.1.

Different indicators are used to evaluate the performance of the machine learning models. Examples of indicators for regression are MAE and maximum error (ME). The proportion of data splitting is 70% as training data and 30% as testing data. Grid search is used for tuning hyperparameters. The list of hyperparameters that are going to be tuned is shown in Table 5-36.

Table 5-36 List of hyperparameters for localizing wheel burns

Model	Hyperparameter	
DNN	Number of hidden layers Activation function Optimizer	Number of hidden nodes Batch size
CNN	Number of convolutional layers Kernel size Pool size Batch size Number of hidden layers	Filter Number of pooling layers Activation function Optimizer Number of hidden nodes
LSTM	Number of LSTM cells Activation function Optimizer	Window size Batch size Number of hidden layers

Model	Hyperparameter
	Number of hidden nodes

5.2.1.4 Results and discussion

From the machine learning model training, the optimized hyperparameters are shown in Table 5-37.

Table 5-37 Tuned hyperparameters for localizing wheel burns

Model	Tuned hyperparameter	Value
DNN	Number of hidden layers	2
	Number of hidden nodes	1,100 (Dense1) and 50 (Dense2)
	Activation function	Sigmoid (Dense1) and ReLu (Dense 2)
	Batch size	32
	Optimizer	Adam
CNN	Number of conv layers	2
	Filter	30 (Conv1) and 32 (Conv2)
	Kernel size	2 (Conv1) and 5 (Conv2)
	Number of pooling layers	2
	Pool size	6 (MaxPooling1) and 2 (MaxPooling2)
	Batch size	32
	Optimizer	Adam
	Number of hidden layers	2
	Number of hidden nodes	200 (Dense1) and 5 (Dense2)
	Activation function	ReLu (Conv and Dense2) and Sigmoid (Dense1)
RNN	Number of LSTM cells	200
	Window size	20
	Batch size	64
	Optimizer	Adam
	Number of hidden layers	2
	Number of hidden nodes	200 (Dense1 and Dense2)
	Activation function	ReLu (Dense1 and Dense2)

The performances of the machine learning model are shown in Table 5-38. The CNN model performs the best based on MAE and ME. These two values demonstrate better performance

when the values are low. The MAE of the CNN model is 0.03 m which is very accurate compared to the total length of the track section or 20 m. The percentage of error is only 0.15%. The DNN model performs better than the LSTM model. According to the table, the LSTM model has an MAE of 0.33 m and the ME is 7.10 m which is very high and can be considered unacceptable. Compared to the advanced approach used to localize defects, lasers seem to be the most accurate approach when the error is less than 0.003 m [162]. The accuracy of ultrasonic is about 98% [163]. Compared to the developed approach in this section, the error is low and acceptable. In addition, the cost is less than laser and ultrasonic so the developed approach is practical for railway operators to implement. For the training time, DNN requires the least duration for training while LSTM requires the longest time to train the model. Considering the error and training time, it can be concluded that the CNN model is the most suitable model for localizing wheel burns in this situation.

Table 5-38 Performance on wheel burn localization

Model	MAE (m)	ME (m)	Training time (s)
DNN	0.27	3.54	0.1
CNN	0.03	0.30	1
LSTM	0.33	7.10	23

5.2.2 Turnout support deterioration prediction under flooding

5.2.2.1 Introduction and backgrounds

This section is extended from Section 5.1.5. In this section, the machine learning model is used to evaluate the turnout support condition under flooding events. Different from the previous

section, the machine learning model is used to predict the current support stiffness directly without classifying the classes of turnout support conditions. The machine learning technique and data used to train the model are the same as in Section 5.1.5. The key indicator to evaluate the performance of the developed machine learning model is R^2. However, other indicators are also used to present the performance of the model such as MAE, RMSE, and MPE.

5.2.2.2 Data and processing

Data variation is shown in Table 5-29. The turnout support stiffness is the key parameter to evaluate the turnout support condition instead of the classes. Data are sampled using stratified sampling to maintain the proportion of samples in different classes.

5.2.2.3 Methodology

The problem in this section is the regression when the predictive values are continuous. In this case, the predictive values are turnout support stiffness which can be any positive real number. Data splitting is conducted using the proportion of 70/30. The machine learning technique used to develop the predictive model is CNN. Features are used in three different ways as Section 5.1.5 which are ABAs, crossing nose displacement, and both ABAs and crossing nose displacement. Hyperparameter tuning is conducted using grid search. The list of tuned hyperparameters is shown in Table 5-30.

5.2.2.4 Results and discussion

The performances of the developed machine learning model are shown in Table 5-39. Different indicators are demonstrated to present the performances of different approaches using different features. As mentioned, this section uses three different approaches in terms of features to develop the machine learning model. They are the use of ABAs, crossing nose displacement, and both.

Table 5-39 Performance on turnout support condition prediction under flooding events

Indicators	MAE	RMSE	R^2	MPE
ABAs	0.28 MN/m	0.36 MN/m	0.98	2.54%
Crossing nose displacement	1.08 MN/m	1.45 MN/m	0.70	11.56%
ABAs and Crossing nose displacement	0.21 MN/m	0.30 MN/m	0.99	1.95%

From the table, the results conform with Table 5-31 to Table 5-33. The use of both features provides the best performance based on MAE, RMSE, R^2, and MPE. The second-best approach is the use of ABAs. It can be seen that these two approaches provide similar satisfying outcomes. However, the use of crossing nose displacement only provides obviously lower performance based on every indicator. R^2 is relatively low compared to the other two approaches. Other errors are higher than the errors from another two approaches as well. R^2 from the best approach is very high which demonstrates the high potential of the developed machine learning model in predicting the turnout support conditions in terms of support stiffnesses. The relationship between predicted values and actual values is shown in Figure 5-26.

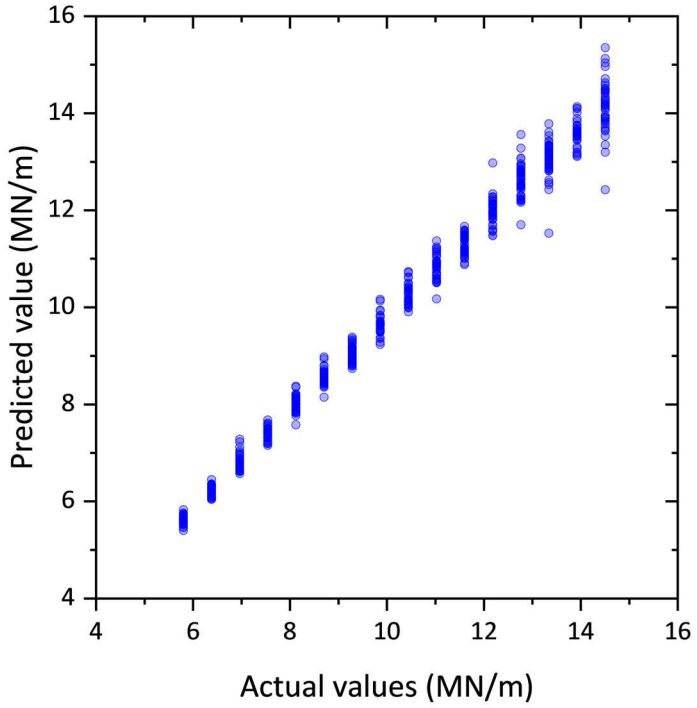


Figure 5-26 Relationship between actual and predicted values of turnout support stiffnesses

The list of tuned hyperparameters is shown in Table 5-34. It can be concluded that the machine learning model has the potential in predicting the turnout support conditions under flooding events. In addition, railway operators can also use the developed model and concept for their railway projects. The railway operators will gain benefits from the precise support condition prediction when they need to make decisions in maintenance. The best practice is not performing the maintenance activities as frequently as possible but performing the maintenance activities as necessary to keep railway infrastructure in good condition enough to operate and avoid severe damages that will cause high maintenance costs and disruptions.

5.2.3 Slab track stiffness prediction

5.2.3.1 Introduction and backgrounds

This section applies the same concept as Section 5.2.2, but the difference is slab tracks or ballastless tracks are used as a case study. Slab tracks are popularly used in the railway industry when speeds of rolling stocks are higher and ballast tracks are not suitable for those higher speeds. Therefore, slab tracks are common in high-speed rail projects. The advantages of slab tracks are they require less maintenance and the structure is more stable compared to ballast tracks. Because slab track structure is concrete, cracks and settlements are common defects in slab tracks [164]. Both of them affect the stiffnesses of slab track which negatively affect the whole stability of track structures. When slab tracks deteriorate, other defects can take place easier, vibrations tend to be higher, and water can penetrate track structures. These all result in poorer quality of track infrastructure. This section aims to develop machine learning models to evaluate the stiffnesses of slab tracks to be beneficial for railway operators to monitor the condition of slab tracks and prepare an appropriate maintenance plan for railway infrastructure.

This section applies validated FEM models to simulate the dynamic behaviour of slab tracks and uses ABAs to evaluate their stiffnesses using machine learning. CNN is used to develop a predictive model. From the literature review, machine learning has not been used to predict slab track stiffnesses. Some related studies were done by Li and Berggren [151] and Guo and Zhai [165] who tried to study slab track stiffness and their deterioration. However, they did not apply machine learning but applied other parameters to predict the stiffnesses such as settlements.

This section aims to fill this gap of research and demonstrate the potential of machine learning to predict slab track stiffnesses.

5.2.3.2 Methodology

FEM models are developed based on the model developed by Li et al. [140]. An example of the FEM models is shown in Figure 5-27. The concept of the development is similar to the models in Sections 5.1.4 and 5.1.5 however the structure is slab track instead. The rolling stocks were modelled using the MBS concept. The track structure consisted of a concrete slab, mortar layer, and concrete base. Each layer was modelled as an elastic layer with different stiffnesses depending on its material properties. The detail of the track structure is shown in Figure 5-28.

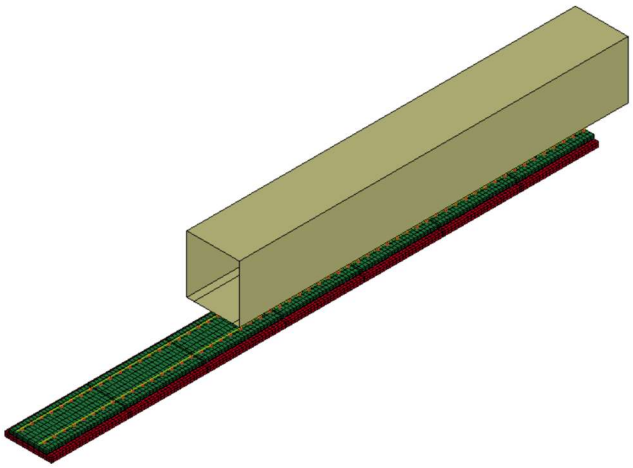


Figure 5-27 FEM model for evaluating slab track stiffnesses

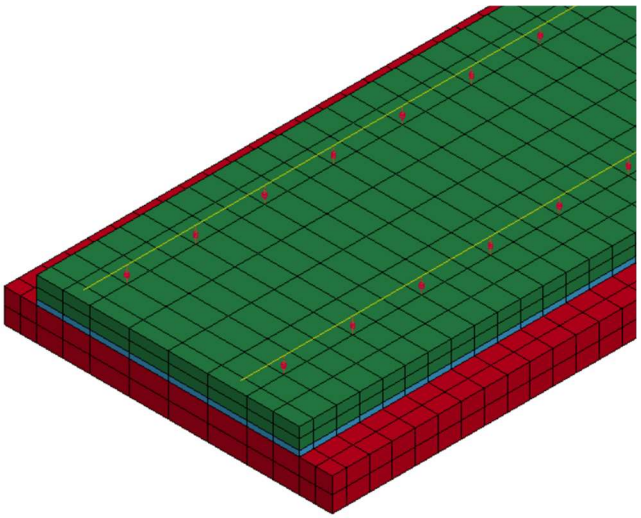


Figure 5-28 Detailed track structure for evaluating slab track stiffness

The FEM model was validated using field data. Field data were collected from the Suining-Chongqing section that was constructed for testing and the rolling stock was Changbai Mountain [166]. To validate the FEM model, wheel-rail contact forces, rail displacement, and sleeper displacement were used to validate the FEM model. The comparison is shown in Table 5-40. It can be seen that the differences between field data and the FEM model are less than 10%. Therefore, the FEM model can be used to imitate the behaviour of the railway infrastructure of slab track.

Table 5-40 The comparison of parameters from field data and the FEM model for slab track stiffness prediction

Parameters	Locations	Field Measurement	FE Model	Difference
Wheel-rail contact force (kN)		100	98.4	1.60%
Rail displacement (mm)	Rail seat	2.606	2.596	0.38%
	Mid span	2.604	2.415	7.26%
Sleeper displacement (mm)	Rail seat	2.576	2.522	2.10%
	Mid span	2.511	2.352	6.33%

CNN is used to develop the machine learning model. It has been proved that it has potential in different areas including the railway system. Grid search is used for hyperparameter tuning to ensure the model performance. The list of hyperparameters is shown in Table 5-41

Table 5-41 List of hyperparameters for predicting slab track stiffness

Model	Tuned hyperparameter
CNN	Number of convolutional layers Filter Kernel size Number of pooling layers Pool size Activation function Batch size Optimizer Number of hidden layers Number of hidden nodes

5.2.3.3 Data and processing

From the FEM models, the data frequency is 1,000 Hz. The total number of samples is 1,458. The total length of the studied track section is 15 m. Different parameters are varied to create data diversity and allow the machine learning model to learn from different situations. The data diversity is shown in Table 5-42. 70% of the data are used to train the model and 30% of the data are used to test the model's performance.

Table 5-42 Parameter variation for slab track stiffness prediction

Parameters	Variation
Slab stiffness	28.8-43.2 GPa
Speeds of the rolling stock	150-200 km/h
Size of irregularity	80-120%
Weights of rolling stock	35.64-43.56 tonnes

Data used to train the machine learning model are ABAs. Because the shapes of features are different based on the speeds of rolling stocks and the data frequency is fixed, data reshaping is done to make all samples have the same shape. The padding is used to reshape samples by adding zero to the samples. An example of ABAs is shown in Figure 5-29. From the figure, ABA from a wheel is presented however ABAs from two wheels will be used to train the machine learning model. The irregularity of the Suining-Chongqing section is shown in Figure 5-30. The size of irregularity is varied as Table 5-42 to create the data diversity.

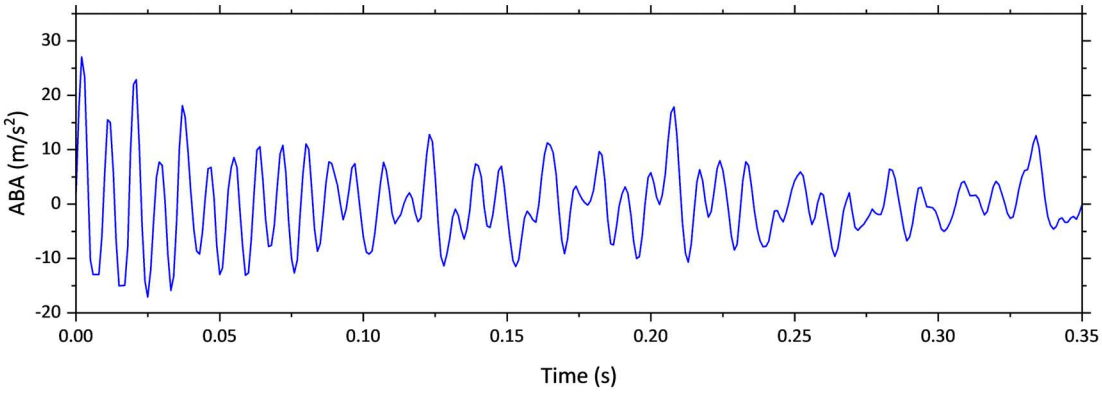


Figure 5-29 Example of ABA for predicting slab track stiffness

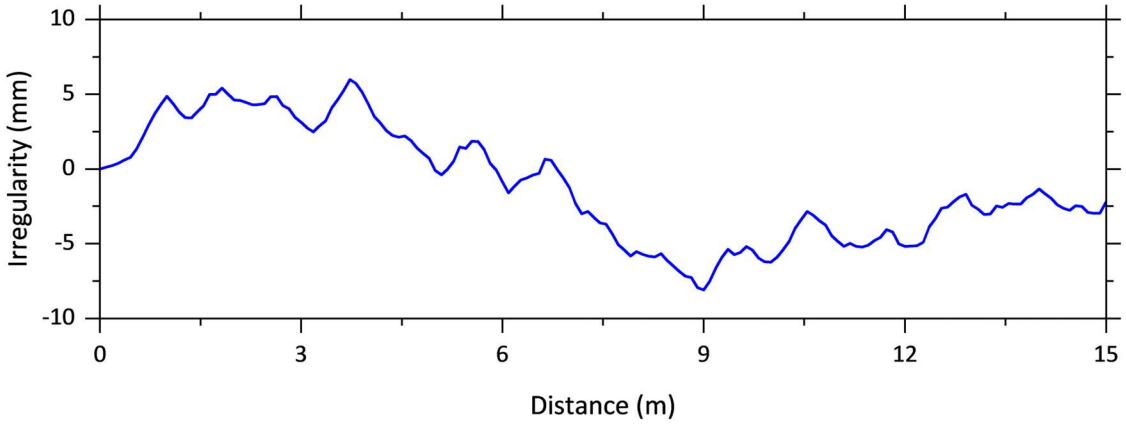


Figure 5-30 Irregularity of the Suining-Chongqing section

5.2.3.4 Results and discussion

From the machine learning model development, the relationship between true and predicted values is shown in Figure 5-31. The R^2 is 0.94 which is significantly high and can be used to conclude that the developed machine learning model using CNN can be used to predict the slab track stiffnesses. Other indicators used to evaluate the performance of the model are shown in Table 5-43. From the table, all errors are low that conforming to the R^2.

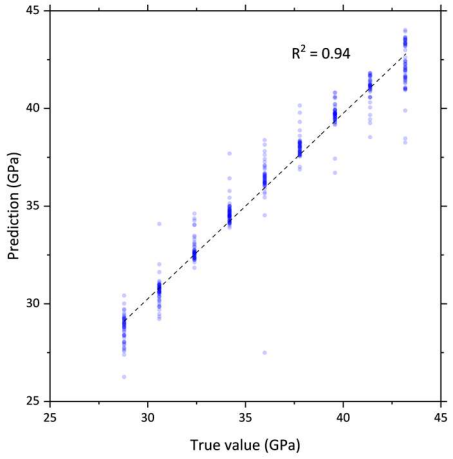


Figure 5-31 Relationship between true and predicted values of slab track stiffness

Table 5-43 Performance on slab track stiffness prediction

Indicators	Values
MAE	0.63 GPa
RMSE	1.13 GPa
MPE	1.75%
R^2	0.94

From the hyperparameter tuning, the set of optimized hyperparameters is shown in Table 5-44.

Table 5-44 Tuned hyperparameters for predicting slab track stiffness

Model	Tuned hyperparameter	Values
CNN	Number of convolutional layers	2
	Filter	64 (conv1) and 32 (conv2)
	Kernel size	3
	Number of pooling layers	0
	Pool size	N/A
	Activation function	Linear
	Batch size	8
	Optimizer	Adam
	Number of hidden layers	2
	Number of hidden nodes	100

From the model training, the accuracy is higher than 98% based on the MPE of 1.75%. In addition, the R^2 is also high. These indicators indicate that the developed predictive model can be used to evaluate the slab track stiffnesses in different situations. railway operators can apply this model or its concept to their projects. They will get the benefit of the ability to monitor the condition of the slab track in real-time. Moreover, data collection can be done easily and fast because this section purposes the approach using ABAs to evaluate the stiffnesses of slab track. There are no additional sensors to be installed on the track side which tends to be expensive. The interruption is also low because data collection can be done using regular operations.

5.2.4 Track geometry prediction

5.2.4.1 Introduction and backgrounds

To evaluate the condition of railway tracks, track geometry parameters can be used as indicators. Tracks in good condition will provide smooth running, good passenger comfort, and good safety. At the same time, tracks with poor conditions cannot provide the desirable characteristics mentioned. Railway operators try to maintain their tracks in good condition for these reasons. Moreover, maintenance costs tend to be higher when railway infrastructure is maintained when it is in very bad condition. Track geometry parameters are traditionally measured by using TGCs. Track geometry parameters consist of superelevation, alignment, longitudinal level, gauge, and twist. They will be further calculated for TQI which will be used to evaluate the track quality. This thesis aims to apply predictive maintenance. Therefore, the ability to predict track geometry parameters is important because railway operators can use this information to prepare their maintenance plans.

This section aims to develop machine learning models to predict track geometry parameters. Features in this section are used in the form of 3D data. The machine learning techniques used in this section are RNN-based models which are suitable for sequence data. The techniques consist of RNN, LSTM, GRU, and attention. The detail of each technique will be presented in Section 5.2.4.3. In this section, track geometry parameters consist of superelevation, two sets of longitudinal levels, two sets of alignments, twists, and gauges. The machine learning models will be used to predict each track geometry parameter independently. This is the advantage of the developed models in this section because most previous studies aimed to predict TQI or deterioration rate which was the summary of track geometry parameters. The ability to predict individual track geometry parameters will convey more detail about the conditions of track quality. Then, railway operators can prepare maintenance plans more efficiently.

Track geometry parameters can be measured using different methods such as linear displacement transducers, laser optics, or tachometers [167]. Track geometry parameters consist of superelevation, alignments, longitudinal levels, gauge, and twists. For longitudinal levels and alignments, values from both rails are used to evaluate the track quality. TQI is used to evaluate the track quality that is calculated based on these seven track geometry parameters. Different railway operators may have different methods to calculate TQIs however the method that is popularly used is the standard deviation. These seven parameters are combined to calculate the standard deviation to indicate the overall condition of the tracks. However, individual track geometry parameters can also be used to evaluate the track condition and allow railway

operators to respond using different approaches. An example of the urgency of response can be categorized as Table 5-45.

Table 5-45 Urgency of response [168]

Urgency	Inspect and verify responses	Action
Emergency 1 (E1)	Before the next train	Before the next train
Emergency 2 (E2)	Within 2 hours or before the next train	Within 24 hours
Priority 1 (P1)	Within 24 hours	Within 7 days
Priority 2 (P2)	Within 7 days	Within 28 days
Priority 3 (P3)	Within 28 days	Plan for respond
Normal (N)	N/A	Routine inspection

The periods for inspection and action are varied depending on the urgency. The urgency of track geometry parameters is depended on each category and the speeds of rolling stocks. For example, the urgency of response based on longitudinal levels is shown in Table 5-46. The table shows both the speed of freight and passenger rolling stocks.

Table 5-46 Urgency of response based on longitudinal levels [168]

Longitudinal levels	Speed in km/h (freight/passenger)					
	20/20	40/40	60/60	80/90	100/115	115/160
0-16	N	N	N	N	N	N
17-20	N	N	N	N	P3	P2
21-24	N	N	N	P3	P2	P1
25-27	N	N	P3	P2	P1	E2
28-30	N	P3	P2	P1	E2	E2
31-32	P2	P2	P1	E2	E2	E2
33-34	P1	P1	E2	E2	E2	E1
35-40	E2	E2	E2	E2	E1	E1
>40	E1	E1	E1	E1	E1	E1

Accelerometers and transducers could be used to measure track geometry parameters [169] however there were limitations in that they could measure only longitudinal levels. There were afforded to apply ABAs to measure track geometry parameters. Ágh [170] tried to find the relationship between ABAs and track geometry parameters using mathematical methods. In that study, the focused parameters were longitudinal levels, alignments, and superelevation. From the study, the correlation between track geometry parameters and ABAs was 0.63. It was noted that the correlation between ABAs and longitudinal levels, alignments, and superelevation was not very high but the correlation between ABAs and twists was even lower. Therefore, there were limitations in that study. Li et al. [171] applied machine learning to predict vertical and lateral loads from wheels when features were track geometry parameters, speeds of rolling stocks, and track designs. They found that the results were satisfying. However, they did not predict the track geometry parameters but they used them as features. Tsunashima et al. [172] developed FEM models to generate ABAs and used them to predict track geometry parameters. The technique used in that study was Kalman Filter. They found that the error was about 3 mm. However, the main limitation of that study was which track geometry parameters they considered and how to calculate them.

Lee et al. [30] deployed SVR to predict alignments and longitudinal levels. Other studies tried to apply machine learning to predict the deterioration of track geometry instead of individual track geometry parameters such as studied sone by Hu and Liu [31], Guler [15], and Soleimanmeigouni et al. [67]. From the literature review, the research gap is a study performing comprehensive track geometry parameter prediction that has never been done. Most studies predicted some

track geometry parameters or their summarized form of them such as TQI or deterioration rate. Therefore, this section aims to fill this research gap by developing machine learning models to predict every track geometry parameter individually which are superelevation, two sets of alignments, two sets of longitudinal levels, gauge, and twists.

5.2.4.2 Data and processing

Samples used to develop machine learning models in this section are processed from two sources of data which are track geometry parameters collected using TGCs as mentioned in Section 3.1 and maintenance records. For track geometry parameters, the data duration is from 2016 to 2019 and measurement is done every foot. The length of the section is 30 km. For the maintenance records, there are seven types of maintenance activities recorded. They are tamping, rail grinding, ballast cleaning, sleeper replacement, rail replacement, fastening replacement, and ballast unloading. Data will be processed to be 3D time-series data consisting of track geometry parameters based on time, track geometry parameters based on distance, and performed maintenance activities. For track geometry parameters based on time, each track geometry parameter in the past of that track section is considered. For track geometry parameters based on distance, each track geometry parameter of the adjacent track sections in the latest year is considered. For performed maintenance activities, every performed maintenance activity in the past of that track section is considered. The overview of features used to train machine learning models is shown in Figure 5-32. For track geometry parameters, the values are real numbers. For maintenance activities, the values are binary numbers or (0, 1) depending on whether each maintenance activity is performed. From the data source, processed data can provide 14,538

samples. The samples are split when 70% are used to train machine learning models and 30% are used to test the performances of models.

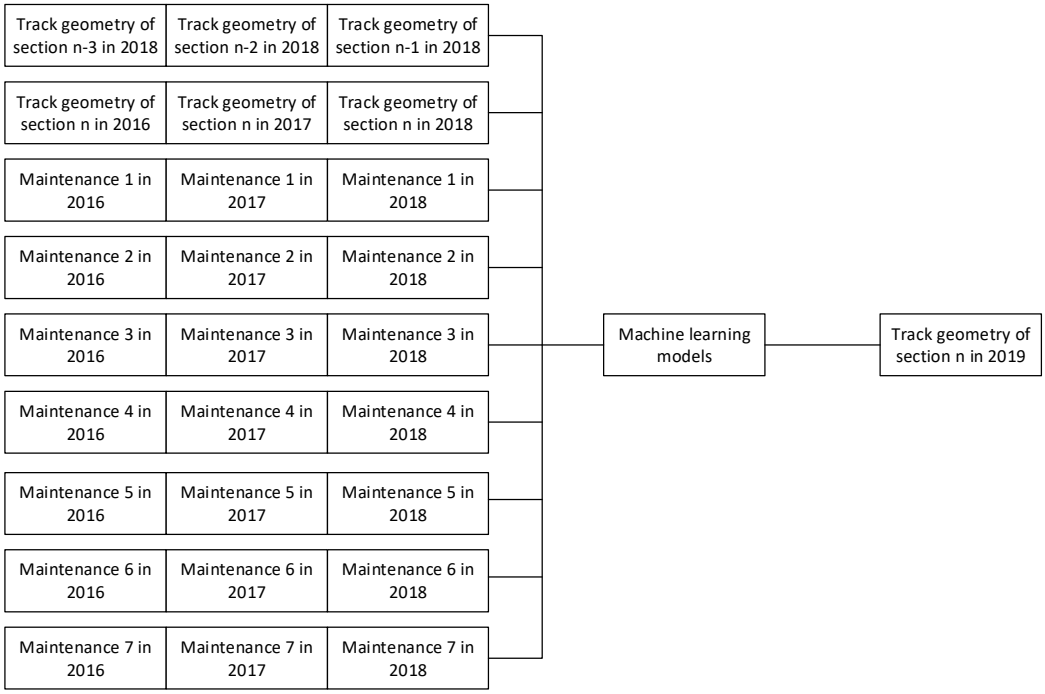


Figure 5-32 Overview of features used for predicting track geometry parameters

5.2.4.3 Methodology

RNN-based models are used to develop machine learning models in this section because they are suitable for time-series data. The techniques that will be used are RNN, LSTM, GRU, and attention.

RNN is the traditional model for time-series prediction. Different from other neural networks, RNN architecture has the memory unit to explore the correlation between previous data and current data. However, the main disadvantage of RNN is it suffers from a vanishing gradient when

the sequence of data is too long. This is because when the sequence of data is longer, the weights of the last features are multiplied many times until the values are close to zero. Therefore, the significance of the last features is low. LSTM is developed to solve the issue found in RNN. The solution is there are three types of gates in the LSTM architecture. They are the input gate, output gate, and forget gate. Input and output gates have the functions of their names to receive and send data through nodes. Forget gates are used to remove unimportant data to keep sequence data not too long. This can reduce the negative effect of the vanishing gradient found in RNN. GRU is further developed based on LSTM. There are two gates in GRU which are update gates and reset gates. In other words, input and forget gates in LSTM are combined to be update gates in GRU. Due to the fewer components in the architecture, GRU can be trained faster than LSTM and should be more stable. Reset gates have the function to memorize and forget data in nodes when the quantity can be varied. Attention is the most advanced RNN-based model. In its architecture, there is a neural network to consider the significance of data before training the model.

MAE, ME, and R^2 are used to evaluate the performances of the machine learning models. hyperparameter tuning is performed and the list of tuned hyperparameters is shown in Table 5-47.

Table 5-47 List of hyperparameters for predicting track geometry parameters

Models	Tuned hyperparameters	
RNN	Normalization	Batch size
	Number of RNN nodes	Number of dense layers
	Number of hidden nodes	Activation functions
	Optimizer	

Models	Tuned hyperparameters	
LSTM	Normalization	Batch size
	Number of LSTM nodes	Number of dense layers
	Number of hidden nodes	Activation functions
	Optimizer	
GRU	Normalization	Batch size
	Number of GRU nodes	Number of dense layers
	Number of hidden nodes	Activation functions
	Optimizer	

5.2.4.4 Results and discussion

From the machine learning model development, different models are tried through hyperparameter tuning. Different indicators are used to evaluate the machine learning models' performance such as MAE, ME, and R^2. The performances of machine learning models are shown in Table 5-48. It can be seen that each track geometry parameter can be predicted the best with different predictive models.

Table 5-48 Performance on track geometry parameter prediction

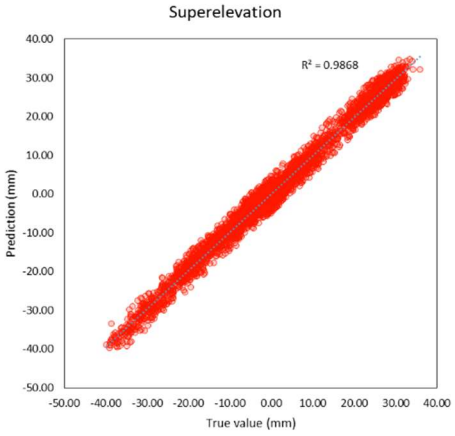
Track geometry parameters	Models	MAE (mm)	R^2	ME (mm)
Superelevation	Attention	1.46	0.99	5.00
Longitudinal level (right rail)	GRU	0.28	0.96	11.26
Longitudinal level (left rail)	GRU	0.33	0.95	10.06
Alignment (right rail)	LSTM	0.13	0.98	5.61
Alignment (left rail)	LSTM	0.20	0.97	6.66
Gauge	LSTM	1.20	0.84	5.99
Twist	LSTM	0.33	0.93	5.60

From the table, the MAEs of all track geometry parameters are less than 1.5 mm which indicates that the developed predictive models can perform well. These performances are also confirmed by R^2 that the values are about 0.85 or higher which means that all models can perform well according to their purposes. It can be seen that different track geometry parameters can be best predicted by different machine learning models. This may be because different track geometry parameters have different characteristics and different models are more suitable in some situations. Most track geometry parameters can be well predicted by the LSTM modes except longitudinal levels that can be better predicted by the GRU model and superelevation that the attention model performs the best. From the table, the RNN model cannot perform the best in any track geometry parameter prediction. This may be because RNN is the least advanced model of RNN-based models. For MEs, all values are acceptable even though they are significantly higher than MAEs however the MEs only represent the maximum values found in rare and extreme cases.

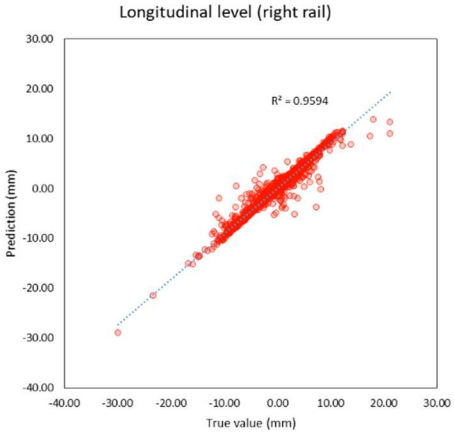
The relationships between true values and prediction can be shown in Figure 5-33. From hyperparameter tuning, the combinations of hyperparameters of each machine learning model can be shown in Table 5-49.

From the results, the developed machine learning models can predict track geometry parameters and provide high accuracy. Railway operators can apply their data to train the machine learning models and use the trained models to predict track geometry parameters in the future. Then, they can use this information for maintenance planning. The only remark is some railway operators might not have the data in the format used in this section however railway operators

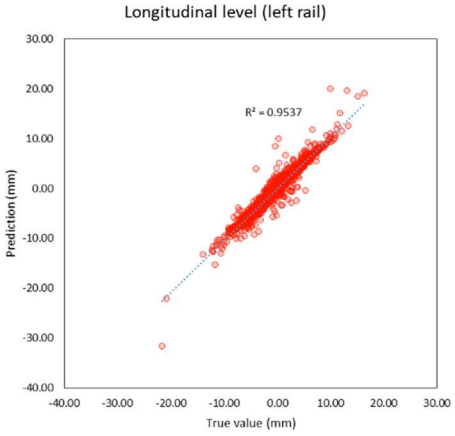
can adjust the architecture of the models and train the models. The developed models show that they have the potential to predict track geometry parameters and can be used in practice.



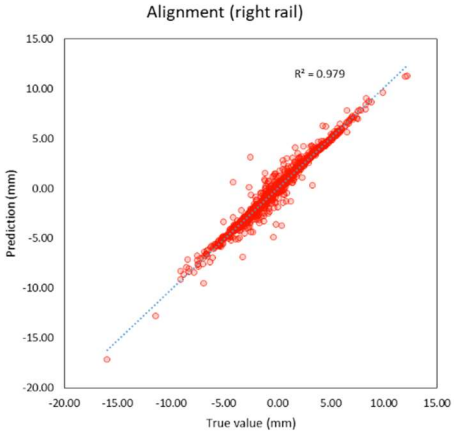
(a)



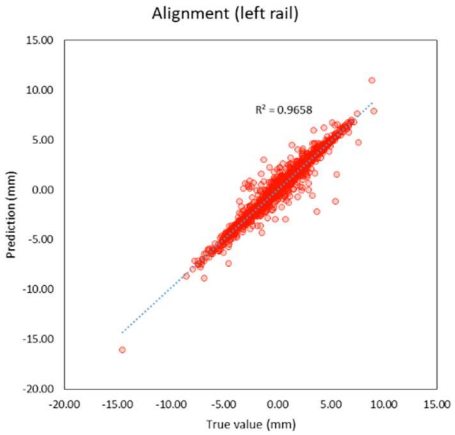
(b)



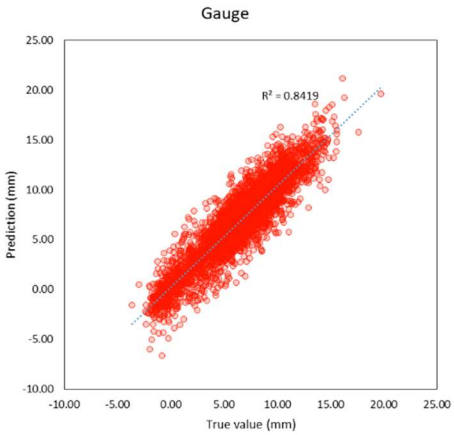
(c)



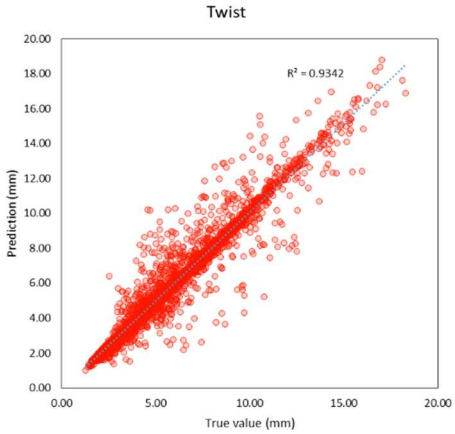
(d)



(e)



(f)



(g)

Figure 5-33 True values and prediction of track geometry parameters (a) superelevation, (b) longitudinal level (right rail), (c) longitudinal level (left rail), (d) alignment (right rail), (e) alignment (left rail), (f) gauge, and (g) twist

Table 5-49 Tuned hyperparameters for predicting track geometry parameters

Models	Tuned hyperparameters	Values
RNN	Normalization	N/A
	Batch size	32
	Number of RNN nodes	200
	Number of dense layers	5
	Number of hidden layers	3

Models	Tuned hyperparameters	Values
	Activation functions	Sigmoid
	Optimizer	Adam
LSTM	Normalization	No
	Batch size	8
	Number of LSTM nodes	3
	Number of hidden layers	2
	Number of hidden nodes	100
	Activation functions	ReLu (dense1) and Linear (dense2 and dense3)
	Optimizer	Adam
GRU	Normalization	N/A
	Batch size	64
	Number of GRU nodes	3
	Number of hidden layers	2
	Number of hidden nodes	100
	Activation functions	ReLu (dense1 and dense2) and Linear (dense3)
	Optimizer	Adam
Attention	Normalization	N/A
	Batch size	64
	Type of attention	Self-attention
	Attention activation functions	ReLu
	Number of hidden layers	2
	Number of hidden nodes	100
	Activation functions	ReLu (dense1 and dense2) and Linear (dense3)
	Optimizer	Adam

6 Unsupervised learning for railway infrastructure maintenance

Unsupervised learning is one of the major types of machine learning. Different from supervised learning, unsupervised learning does not need labels of samples to be trained. Unsupervised learning is used to explore the hidden pattern and insight of data by machines completely without human knowledge. Some patterns and insights may be impossible to discover manually by humans. Therefore, unsupervised learning plays an important role in this function. Two popular techniques of unsupervised learning that will be used in this thesis are K-mean clustering and association rules.

6.1 K-mean clustering

6.1.1 Insights of track component defects based on track design

6.1.1.1 Introduction and backgrounds

This section is an extensive study from Section 5.1.3. In that section, track geometry parameters and track design are used to predict track component defects. In this section, k-mean clustering is used to explore the hidden patterns and insights from the data. K-mean clustering is one of the popular techniques in unsupervised learning. The technique is used to cluster or groups sample together based on defined features. The number of clusters is pre-defined by humans and the algorithm will cluster samples into that defined number of clusters. The concept of k-mean clustering is the machine will randomly define the centres or centroids of each cluster based on features. Then, each sample will be assigned to a cluster by considering the distance between each cluster and sample based on the feature. The technique used to calculate the distance is

Euclidean distance. After all samples are assigned to clusters, the centres of each cluster will be re-calculated again using the average values of features of each sample in the clusters. Then, samples will be assigned to new clusters. These processes will be repeated until there is no change in sample assigning and clusters' centre calculation.

Features used in this section are from track design. Two main features are the radius of curvature and the gradient of tracks. Samples will be grouped as clusters depending on these two features. Then, samples in each cluster will be further explored about the defects occurring in the clusters to see the relationship between two features and occurring defects. Railway operators will get benefit from knowing the types of defects that tend to emerge at some locations of tracks based on their characteristics. For example, tracks with a specific radius of curvature and gradient tend to have a specific defect rather than other characteristics of tracks.

6.1.1.2 Data and processing

Data and processing in this section have the same concept as Section 5.1.3 due to this section is an extensive study of that section. In conclusion, track geometry parameters are removed from the machine learning model training. Therefore, only data from track design reports and defect inspection reports are used. For the track design reports, two data are extracted from the reports which are the radius of curvature and the gradient of track sections. For the defect inspection reports, track component defects are used to explore the defect types in each cluster after samples are grouped using k-mean clustering.

6.1.1.3 Methodology

As mentioned, k-mean clustering is used to cluster samples. The concept is to assign samples into clusters when the number of clusters is defined. The samples are assigned to different clusters using the distance between samples and clusters. The distance can be calculated using Equation 7. From the equation, there are two features which are x and y. Then, the assigned cluster is considered using Equation 8 when c is centroid and k is the number of clusters. In other words, defined features are used as centroids of each cluster. Samples will be assigned to the clusters where the distance of features is the closest one. Then, new centroids are calculated again using Equation 9. The new centroids are the average values of features of samples in that cluster [173].

$$dist(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad \text{Equation 7}$$

$$c_j = \left\{ x : \min_k dist^2(x, c_k) = j \right\} \quad \text{Equation 8}$$

$$c_j = \frac{1}{m_j} \sum_{x \in c_j} x \quad \text{Equation 9}$$

Features used in this case are the radius of curvature and the gradient of tracks because this study wants to investigate the relationships between defects and track design which are unchanged. Other changeable parameters can also be considered such as speed and weight of rolling stock in further studies. After k-mean clustering is conducted, track component defects

that are extracted and processed from defect inspection reports will be analyzed to see the relationships between track characteristics and track component defects.

6.1.1.4 Results and discussion

In this section, results from k-mean clustering are presented. As mentioned, the number of clusters has to be defined in the early stage of model development. In this case, the number of clusters is set to four based on the radius of curvature and gradient. It is assumed that these two values with high and low values suppose to affect the types of defects. Therefore, the number of clusters is defined as four. Results are shown in Figure 6-1. It is worth noting that the radius of curvature of the tangent track is infinity. The figure shows the values of the radius of curvature until 4,500 but the samples beyond that value are categorized in cluster 3. From the figure, there are four clusters, from cluster 0 to cluster 3. It is conformed to the hypothesis that the clusters should be conducted in the way of low and high values of the radius of curvature and gradient. To convey a clearer view, a tree diagram is used to present the clustering as shown in Figure 6-2. The centroids of each cluster are shown in the figure. The first number is the gradient and the second number is the radius of curvature. It can be seen that if the radius of curvature is more than 4,000 m, the samples are assigned to cluster 3 without any further consideration. After that, the samples are assigned to cluster 0 if the gradient is negative or downhill. If the gradient is positive and the radius of curvature is less than 1,800 m, the samples are assigned to cluster 1. The rest of the samples are assigned to cluster 2. From the result, it makes sense that the smaller radius of curvature is related to the higher number of defects. At the same time, the higher gradient also results in a higher number of defects. It can be inferred that this phenomenon is

caused by the higher friction and impact from the sharp curve and braking force that create defects to track components.

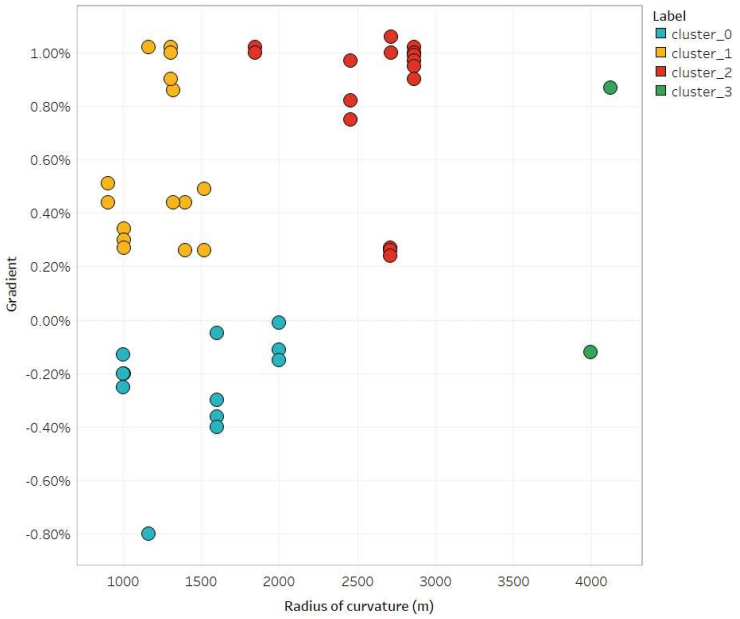


Figure 6-1 Results of k-mean clustering

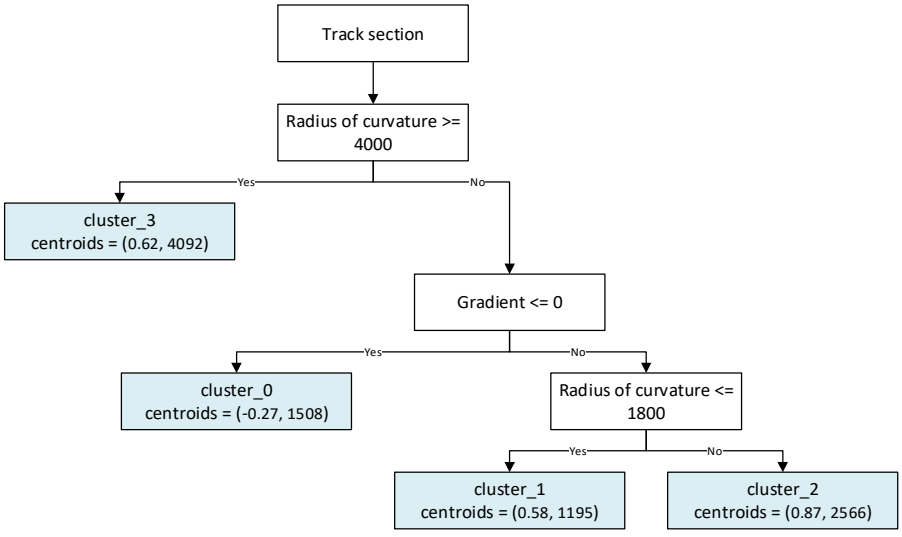


Figure 6-2 Tree of clustering

When the radius of curvature is increasing more than 3,000 m, the number of defects drops significantly. The relationships between the clusters and component defects are shown in Table 6-1.

Table 6-1 Cluster and sample distribution related to each other

Based on component defects				
	cluster_0	cluster_1	cluster_2	cluster_3
S&C[1]	12%	73%	8%	8%
Fastener[2]	53%	21%	26%	0%
Rail joint[3]	39%	26%	32%	3%
Rail[4]	38%	38%	17%	7%
Based on clusters				
	S&C	Fastener	Rail joint	Rail
cluster_0	2%	7%	34%	56%
cluster_1	15%	3%	24%	59%
cluster_2	3%	6%	48%	43%
cluster_3	10%	0%	20%	70%

From the table, different insights can be drawn from k-mean clustering. First, there is no defect found when the gradient is negative and the radius of curvature is between 2,000 and 4,000 m. The majority of switch and crossing defects are categorized in cluster 1 where the radius of curvature is less than 1,800 m and the gradient is positive. Therefore, it can be inferred that switch and crossing are sensitive in the location with this characteristic. However, most of the other defects are in cluster 0 which is the sharp curve and downhill. These track components may

[1] Besides S&C defects, bolt related to S&C, frog, switch point, crossing, switch plate, switch rod, switch heel, guard rail, point blade, stock rail, and wing rail are also included.
[2] Besides fastener, clip and spike are also included.
[3] Besides rail joint defects, joint, weld and bolt related to joints are also included.
[4] Besides rail defects, obstructions on rails, track geometry, and buckling are also included.

suffer from the high braking force and defects develop. Very few defects are in cluster 3 when the tracks have a high radius of curvature and downhill as mentioned. Additionally, there are no fastener defects occurring in this cluster. When considering based on clusters, the type of defect that most frequently is found in almost every cluster is the rail defect. However, rail joint defects are the majority in cluster 2. This can be inferred that the rail is the track component that is contacted by wheels and the loads are directly applied to the rail. Therefore, it is the component that frequently has defects. This also applies to cluster 3 when the radius of curvature is large and the track is downhill. The defects that are usually found are also rail defects.

These findings will be beneficial to railway operators in terms of monitoring and awareness. Some characteristics of track cause more defects than others. At the same time, some kinds of defects tend to be found in specific track characteristics. This will allow railway operators to be aware of where they should pay more attention to inspect defects and maintain railway infrastructure carefully.

6.2 Association rules

6.2.1 Insights of track component defects based on relevant defects

6.2.1.1 Introduction and backgrounds

This section is an extensive study from Section 5.1.3. In that section, grouped track component defects are explored to see their relationships. Therefore, the main features used in this section are grouped track component defects only. In this section, association rules are used to explore the hidden patterns and insights from the data. Association rules are popular unsupervised

learning used to discover the hidden pattern in the data. This technique is used to find events that frequently happen together. When some events happen together frequently, association rules can be defined as hidden patterns.

Features used in this section are from defect inspection reports. Different track component defects are used as features to train the unsupervised learning model. The insights from this section will discover the cooccurrence of defects. Railway operators will be able to know what kinds of defects tend to take place together and be aware of other kinds of defects when some kinds of defects are found.

6.2.1.2 Data and processing

Data and processing in this section have the same concept as Section 5.1.3 due to this section is an extensive study of that section. In conclusion, track geometry parameters and track design data are removed from the machine learning model training. Therefore, only data from defect inspection reports are used.

6.2.1.3 Methodology

To consider the types of defects that emerge in track sections, foot-by-foot data is not sufficient to see the overview of defects because it is too detailed and the correlation between defects is not clear. In this section, a track section of 50 ft is used to explore the occurrence of track component defects. If defects occur in the same 50-ft track section, it will be considered that those defects occur together. The data processing is conducted in this way to suit the machine learning model development. The association rules are used to discover the insights between

track component defects. To analyze the relationship, some indicators are used to identify the strength of the relationship. Some relationships may be too weak and defects occur together by chance. However, some insights can be indicated if the relationships are strong enough. To identify the strength of the relationships, different indicators are used consisting of support, confidence, and lift. These indicators can be calculated using Equation 10 to Equation 12. x and y are events that frequently happen together. In this case, there are two types of track component defects happening in the same track section. N is the total number of samples. It is worth noting that Equation 10 to Equation 12 present simplified equations related to association rules. In real situations, more than two events are considered for the relationships. However, these equations are aimed to convey a simple understanding of the function of association rules and the calculations.

$$Support = \frac{frq(x,y)}{N} \quad \text{Equation 10}$$

$$Confidence = \frac{frq(x,y)}{frq(x)} \quad \text{Equation 11}$$

$$Lift = \frac{support(x \cup y)}{support(x) \times support(y)} \quad \text{Equation 12}$$

The support presents how frequently selected events or defects take place together in the whole sample track section. The higher values indicate that they are likely to be found together. Confidence is the main indicator of association rules. It identifies the frequency of the rules becoming true when defect x emerges. In other words, it indicates how much possibility of defect

y is found when defect x is found. Lift is the proportion between the support to the expected support of the rules. It measures how much more likely defect y is found when defect x is found compared to when there is no association between them. In this section, track component defect data are used but they will not be processed to be groups of track components. Raw data from the defect inspection reports will be used directly to see the variety of defects and their relationships.

6.2.1.4 Results and discussion

There are many rules found in association rules. In terms of support, there are 62 rules that have a support value higher than 0.05 or are found more than 5% in the total samples. However, to demonstrate the strong relationships between events, the rule with a confidence higher than 0.8 will be presented in Table 6-2. In the table, there are left-hand side and right-hand side events. The left-hand side events can be considered as x events which are the events that can be detected first and right-hand side events can be considered as y events or the consequent events.

Table 6-2 association rules with a confidence higher than 0.8

Left-hand side	Right-hand side	Support	Confidence	Lift
Failed ties in switch, Surface defect on frog	Rail surface defect	0.056	0.957	1.926
Worn wing rail, Failed ties in switch, Worn point blade	Rail surface defect	0.052	0.953	1.918
Failed ties in switch, Worn point blade	Rail surface defect	0.056	0.948	1.908
Worn wing rail, Worn point blade	Rail surface defect	0.059	0.927	1.867
Rail surface defect, Failed ties in switch, Worn point blade	Worn wing rail	0.052	0.927	6.204
Failed ties in switch, Worn point blade	Worn wing rail	0.054	0.922	6.171

Left-hand side	Right-hand side	Support	Confidence	Lift
Rail surface defect, Worn point blade	Worn wing rail	0.059	0.898	6.015
Rail surface defect, Worn wing rail, Worn point blade	Failed ties in switch	0.052	0.878	6.896
Failed ties in switch, Worn point blade	Rail surface defect, Worn wing rail	0.052	0.878	7.805
Worn wing rail, Worn point blade	Failed ties in switch	0.054	0.855	6.712
Rail surface defect, Worn point blade	Failed ties in switch	0.056	0.852	6.686
Worn wing rail, Failed ties in switch	Rail surface defect	0.073	0.841	1.694
Worn point blade	Rail surface defect	0.065	0.837	1.684
Surface defect on frog	Rail surface defect	0.074	0.823	1.657
Worn wing rail, Worn point blade	Rail surface defect, Failed ties in switch	0.052	0.815	7.922
Worn point blade	Worn wing rail	0.063	0.810	5.426
Failed ties in switch	Rail surface defect	0.103	0.807	1.625

It is worth noting that the support may not be high because the number of events is compared to the total number of samples which is very high. However, some events have a support value higher than 0.1, or 10% of the samples have this phenomenon. Many events have high values of confidence which means the right-hand side events tend to be found when the left-hand side events are detected.

This finding will support railway operators in railway inspection and maintenance because they can better know what defects should be attended to when they find some specific defects. for example, they can expect to find rail surface defect when they have found failed ties in the switch and surface defect on frog which is the association rules with the highest confidence. Other rules can also be applied in the same way.

7 Reinforcement learning for railway maintenance scheduling

7.1 Railway maintenance efficiency improvement

7.1.1.1 Introduction and backgrounds

This section presents how to apply reinforcement learning which is one of the main types of machine learning to improve railway maintenance. The railway system is complicated due to the number of components and related activities in the system. They are all relevant and affect each other in different ways. Therefore, decision-making about railway maintenance is a complicated task for railway operators. This section will explore the potential of machine learning in supporting decision-making in railway maintenance. In addition, this section will present whether reinforcement learning can improve the efficiency of railway maintenance. This thesis focuses on predictive maintenance so the ability to prepare maintenance plans using reinforcement learning will be beneficial for railway operators to optimize maintenance activities. Excessive maintenance will lead to unnecessary maintenance costs and interruption. On the other hand, insufficient maintenance will lead to poor conditions of railway infrastructure. Reinforcement learning is expected to resolve these issues by providing optimized maintenance plans. Railway infrastructure will be in standard condition while the maintenance is limited as necessary.

This section applies reinforcement learning to improve the maintenance efficiency of railway infrastructure which has not been popular in the railway system nowadays. A preliminary concept of reinforcement learning is the agent of the reinforcement learning model will try to maximize rewards to get the best solutions for different problems. The rewards of reinforcement learning

models can be varied from maintenance cost, maintenance duration, number of defects, or carbon emissions. In this section, maintenance cost and the number of defects are used as rewards to train the agent.

This section aims to apply reinforcement learning to improve railway maintenance efficiency. The number of defects and maintenance costs are used as rewards for the reinforcement learning agent. Defect data are collected from field data when they are in the form of track geometry parameters and component defects. For track geometry parameters, data sources are from MRS Logística S.A. as mentioned in Section 3.1. Therefore, the inspection manual from the same organization is used to indicate the defects. From the manual, the threshold of each track geometry parameter is indicated. If the values of parameters exceed the thresholds, they can be considered defects. For track component defects, they are collected from defect inspection reports. Track geometry parameter deterioration and defect occurrences are related to maintenance activities. Therefore, performed maintenance activity data are collected from maintenance records. Then, these data are analyzed together to find the relationships and develop the environment for the reinforcement learning model. There are many reinforcement learning techniques that can be used for the task in this section. This section will apply A2C as its advantages. Many studies indicated that the performance of A2C is better than the traditional techniques such as DQN and the training time is shorter [174]. The expected benefits of this section are railway operators can apply the developed model and concept in this section to reduce the number of defects, maintenance costs, and possession time. Each track geometry parameter and the component defect will be considered independently by the reinforcement

learning agent to prepare the maintenance plan. In addition, the model also has the benefit of sequence decision-making that supervised and unsupervised learning techniques do not have this capability.

Different techniques have been used to plan maintenance activities in the railway system. Sedghi et al. [175] reviewed the previous studies related to this topic. They found that machine learning had not been widely used to improve the efficiency of railway maintenance. Most studies applied mathematical techniques or probability methods to plan railway maintenance. Examples of these approaches were stochastic models, mathematical simulations, mixed integer programming, and the Markovian model. They also mentioned that machine learning had the potential to improve railway maintenance efficiency. In addition, machine learning started drawing attention from society nowadays because of its performance.

Lopes Gerum et al. [176] applied machine learning to predict the occurrence of railway defects. The techniques they applied were RF and RNN. They claimed that the RNN and RF models could achieve accuracies of 80% and 77% respectively. In addition, they also applied the machine learning models to predict the severity of defects when they classified the severities into two groups which were minor and major defects. After that, they deployed discounted Markov decision process for planning the inspection and maintenance. They stated that the developed approach could save some inspection and maintenance costs. They suggested that further development could be done by including a detailed maintenance plan.

Consilvio et al. [34] focused on earthwork and track circuits in the railway system. They applied machine learning techniques to create numerical data to optimize maintenance planning. For the machine learning techniques, they used to generate the numerical data, they applied k-mean and SVM. After that, mixed integer linear programming and Monte Carlo simulations were used to optimize the maintenance planning. To evaluate the performance of the developed models, they applied key performance indicators (KPIs) to evaluate the track condition resulting from the maintenance planning. They concluded that the developed mathematical models could save maintenance costs. It can be seen that the research gaps of that study are the data generated and the machine learning or reinforcement learning that has not been used to optimize railway maintenance.

Mohammadi and He [177] deployed DQN to prepare the maintenance plan for the railway system. The criteria or states they provided to the reinforcement learning agent to select actions were TQI and hazard index. As the study in the previous paragraph, they applied RF to obtain these data. The available actions or action space consisted of five maintenance activities. They are tamping (preventive and predictive maintenance), grinding (preventive and predictive maintenance), and renewal. They concluded that the developed DQN model could reduce TQI and hazard index which demonstrated the better condition of tracks. However, it can be seen that the number of available actions that the agent can choose is limited. In addition, more advanced techniques rather than DQN can be used to develop the reinforcement learning model. These research gaps are aimed to be fulfilled by this section. This thesis aims for predictive or conditional-based maintenance but the previous study included preventive maintenance

indicating that the reinforcement learning model might not demonstrate its full potential. In addition, reinforcement learning application in railway predictive maintenance has not been comprehensively explored and some parameters should be considered such as track geometry parameters or defects collected based on field data. Therefore, this section will explore the potential of reinforcement learning in railway infrastructure predictive maintenance.

7.1.1.2 Methodology

Reinforcement learning is the most advanced and difficult machine learning technique compared to supervised and unsupervised learning. The main reason is the reinforcement learning agent will learn from rewards and feedback under specific situations. That means the training of reinforcement learning requires a specific circumstance or customized environment depending on the problems aimed to be solved. In other words, the environments have to be developed to suit the problems and cannot be reused in other problems. Because of its potential, reinforcement learning is increasingly attended by many industries and aimed to be used to solve problems.

For reinforcement learning, agents interact with the environment by learning and receiving feedback via rewards [176]. Environments are the components of the reinforcement learning models that define the rules of the models including action spaces and constraints. Stages are timesteps ($t = 1, 2, 3, \dots, n$) that agents have to interact with environments. Agents will select actions to interact by considering states ($s_t \in S$ where S is possible states) which can be considered inputs or information perceived by agents. Agents will be trained to take action ($a_t \in$

$a(s_t)$ where $a(s_t)$ is possible actions). In the beginning, agents may select actions randomly when the training is not finished. Then, agents will learn how to select the best actions by considering rewards (R_t) provided by environments. The aim of agents is to maximize rewards. Therefore, when agents are trained multiple times, agents will know which actions are the best under specific situations. These processes will be repeated until the training is finished. In this section, the overview of the training process can be shown in Figure 7-1.

There are 12 states that the agent will use to take action. These states are based on field data which are seven track geometry parameters and five component defects. For the action space or maintenance activities that the agent can choose to perform, there are seven available maintenance activities. The reinforcement learning technique used in this section is A2C. For the rewards, they can be considered using two indicators which are the maintenance costs and the number of defects. The agent has to balance between maintenance activities and defect occurrence. If the maintenance is performed excessively, defects may not occur but the maintenance costs will be high. This is the challenge that railway operators also try to achieve.

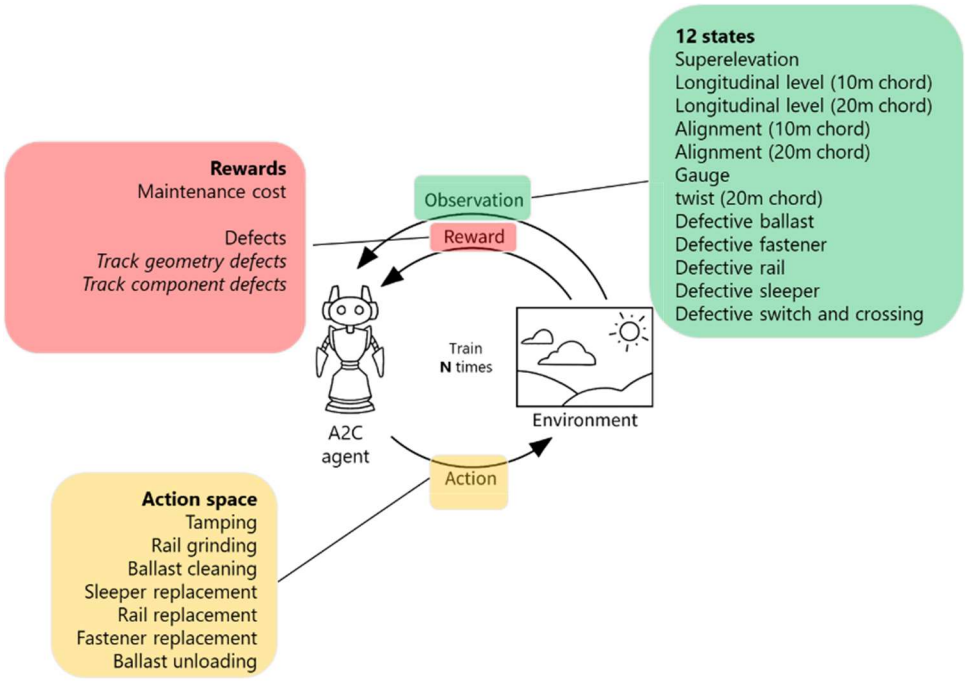


Figure 7-1 Overview of training process of the reinforcement learning model for maintenance efficiency improvement

From the workflow, the problem in this section cannot be solved by supervised or unsupervised learning because supervised and unsupervised learning does not have the capability to make a decision in a series of time. Reinforcement learning agents are trained based on rewards while supervised and supervised learning are trained by the loss function. Therefore, reinforcement learning agents will have the ability to consider short-term (each stage) and long-term (last stage) rewards by using value functions.

As mentioned, this section applies A2C to develop the reinforcement learning model. A2C is an on-policy reinforcement learning technique. Reinforcement learning can be categorized as on-policy and off-policy. Off-policy reinforcement learning is suitable when the agent has to create

new data such as painting [178]. On-policy reinforcement learning will have the advantage when action spaces are limited and can be defined. In addition, on-policy reinforcement learning is more stable when the aim of the model is clear. In this section, on-policy reinforcement learning is more suitable. Examples of popular on-policy reinforcement learning techniques are A2C and PPO.

Besides on-policy and off-policy models, reinforcement learning models can be categorized as policy-based and value-based models. A2C is a hybrid of them because A2C will learn how to take actions based on states which are characteristic of policy-based models. At the same time, A2C agents will also learn from the forecast values of states and actions which is characteristic of value-based models. Therefore, A2C will obtain the advantages of both categories. A2C contains two neural networks acting as the actor and critic. The actor part selects actions to take using Policy Gradient. The critic part considers the policy function $\pi_\theta(s_t, a_t)$ using Q-learning to consider whether the taken actions are good or not using an advantage function $A(s_t, a_t)$. The advantage function is calculated using the value function $\hat{q}_w(s_t, a_t)$ and average value of states $V(s_t)$. States (s_t) and actions (a_t) are used to calculate the Q-value $\hat{q}_w(s_t, a_t)$. The equation is shown in Equation 13.

$$A(s_t, a_t) = \hat{q}_w(s_t, a_t) - V(s_t) \qquad \text{Equation 13}$$

In this section, available actions are rail grinding, tamping, sleeper replacement, ballast cleaning, fastening components replacement, rail replacement, and ballast unloading. Because the customized environment developed in this section allows the combinations of performed

maintenance activities, the agent is flexible to select any maintenance activities to perform. The number of possible actions or action space will be 128 which can be calculated using Equation 14 when n is the total number of available actions and r is the number of selected actions in each stage. In this case, the agent can choose to perform nothing (zero maintenance activity) until all maintenance activities (seven activities). Therefore, r can range between zero to seven.

$$Combinations = \frac{n!}{(n-r)!\,r!} \qquad \text{Equation 14}$$

There are 12 states that the agent can consider to take actions which are seven track geometry parameters and five component defects as mentioned. for the first stage of the training, states are based on field data. The states of the following stages will be based on the actions that the agent takes. The changes in states are based on field data. Because the data used in this section are field data, the changes in states are considered using the normal distribution analyzed from field data. Track geometry parameters are in the form of real numbers when the component defect occurrences are in the form of probability. Rewards are in the form of maintenance costs and the number of defects. To consider defects, track geometry parameters that exceed the threshold will be considered defects. The threshold of each track geometry parameter is shown in Figure 7-2. For component defects, the states are the probability of defect occurrence. At the end of each stage, if component defects occur based on their probability, they are considered defects. However, if defects do not emerge, the probability of occurrence will be higher in the next stage.

The overall workflow of the reinforcement learning model development is shown in Figure 7-3.

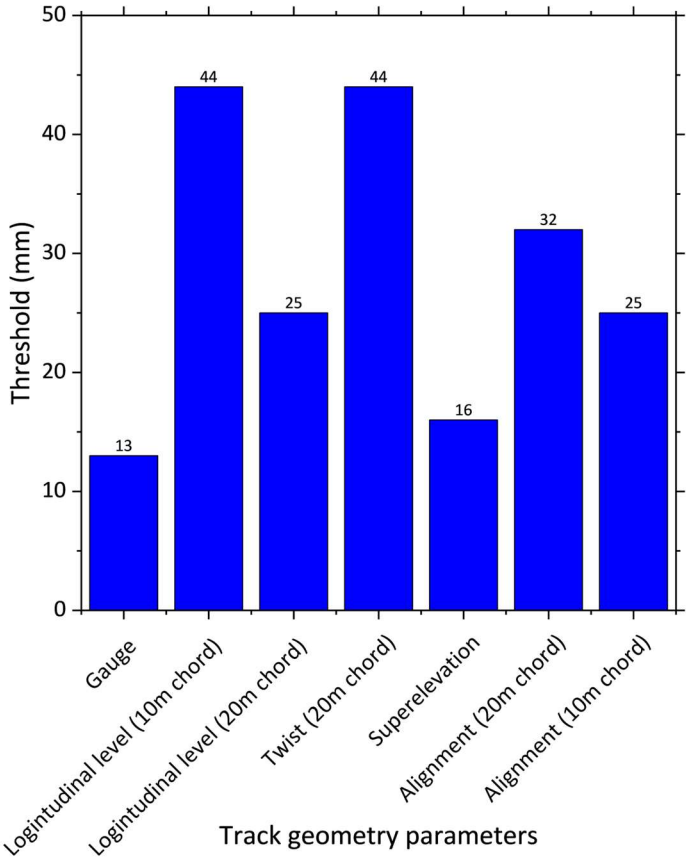


Figure 7-2 Threshold of track geometry parameters

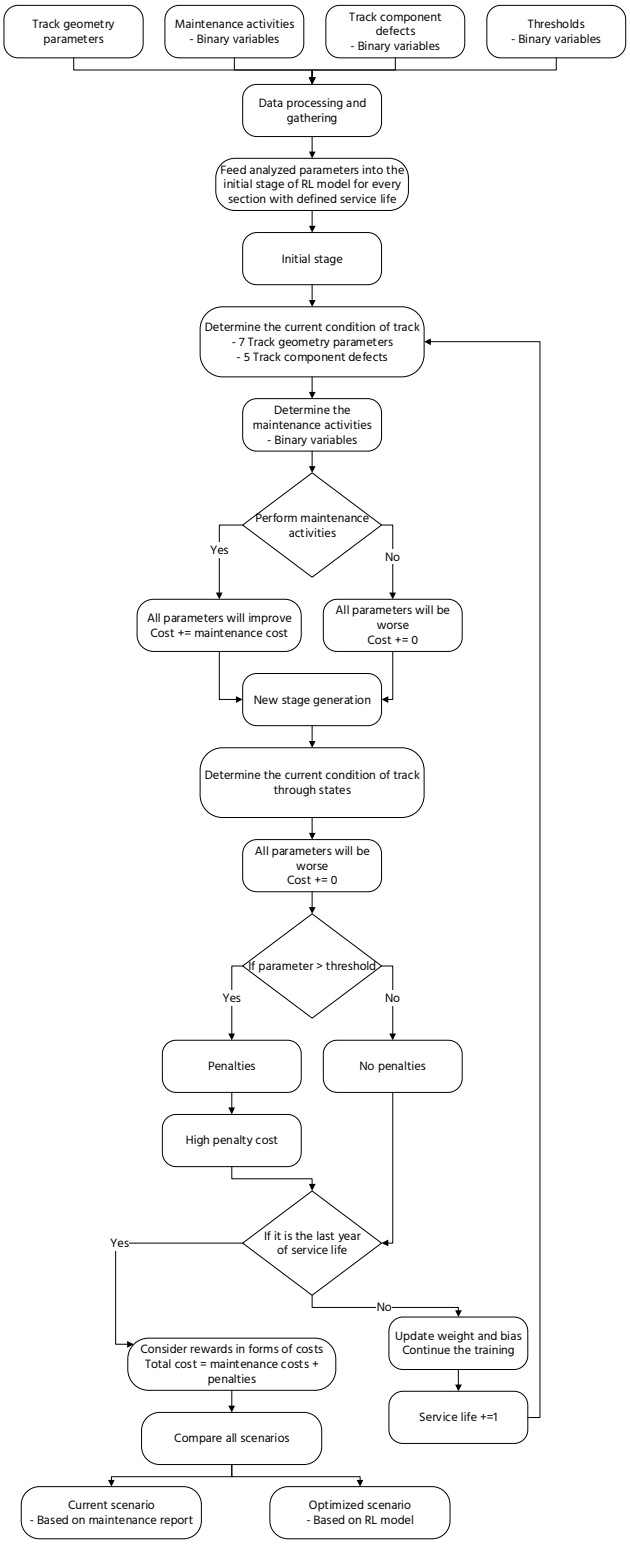


Figure 7-3 Workflow of A2C model development

Hyperparameter tuning is also applied. Different hyperparameters are tuned using grid searches such as learning rate, discount factor, and epsilon. A sensitivity analysis of hyperparameters is performed to see the effect of each hyperparameter on model performance. In the section, the total number of stages is defined to be 100,000.

7.1.1.3 Data and processing

Data used in this section is the field data from MRS Logística S.A. as mentioned in Section 3.1. Sources of data are track geometry parameters from TGCs, maintenance manual defining the track geometry parameter threshold, defect inspection report, and maintenance records.

MRS Logística S.A. defines the values of each track geometer parameter as priority levels from levels 1 to 4. In this section, thresholds from level 4 are used because this section aims to minimize the number of defects. For track component defects, the defect inspection reports identify 71 types of defects which may not be suitable for developing the reinforcement learning model because the data are too scattered. Therefore, defects are grouped based on track components which are rail, sleeper, switch and crossing, fastener, and ballast.

To demonstrate the correlation of each state (track geometry parameters and component defects), correlation analysis is performed. The results are shown in Figure 7-4. From the figure, track geometry defects are highly correlated to each other except gauge that the correlation is about 0.7. Therefore, it can be expected that if one track geometry defect is found, other track geometry defects except gauge defects have a high possibility to be found as well. Track component defects have lower correlations. This can be inferred that each track component

defect tends to be independent of each other track component defects. The only component defect that has a high correlation with track geometry defects is switch and crossing.

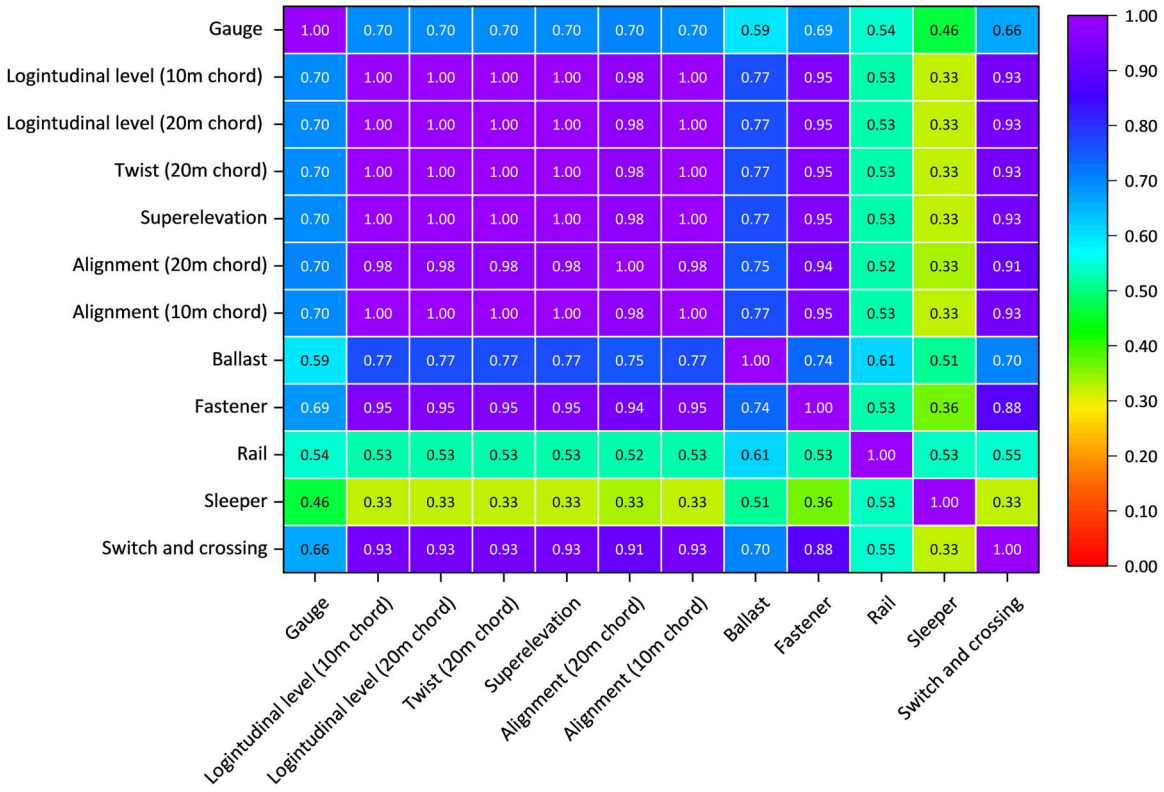


Figure 7-4 Correlation between defects

From the preliminary analysis, the average number of defects found in every track section is two. For the maintenance activities, there are seven available activities as mentioned. The agent will select to perform each activity in the form of binary space (do or not do). There are 128 possible combinations in which the agent can take action. Each combination results in different improvements and deterioration of states. Because the period of field data is four years, the data are comprehensive for every combination of maintenance activities. When maintenance

activities are performed, the condition of railway infrastructure will be improved as well as states. The improvement and deterioration of each state when maintenance activities are performed and not performed are shown in Figure 7-5 to Figure 7-7. The first two figures present how each track geometry parameter improves and deteriorates when tamping is and is not performed. The third figure presents the changes in the probability of defect occurrences when rail grinding is performed and is not performed. The improvement and deterioration are presented in the form of mean and standard deviation. The environment of the reinforcement learning model is developed to use these values with Monte Carlo simulation to consider the states of the next stages when the agent chooses actions to take.

A preliminary analysis is performed. The results show that the maintenance was not done efficiently. Some track sections with good conditions are maintained while some track sections with bad conditions are not maintained sufficiently. Therefore, there are a number of defects found in the track sections.

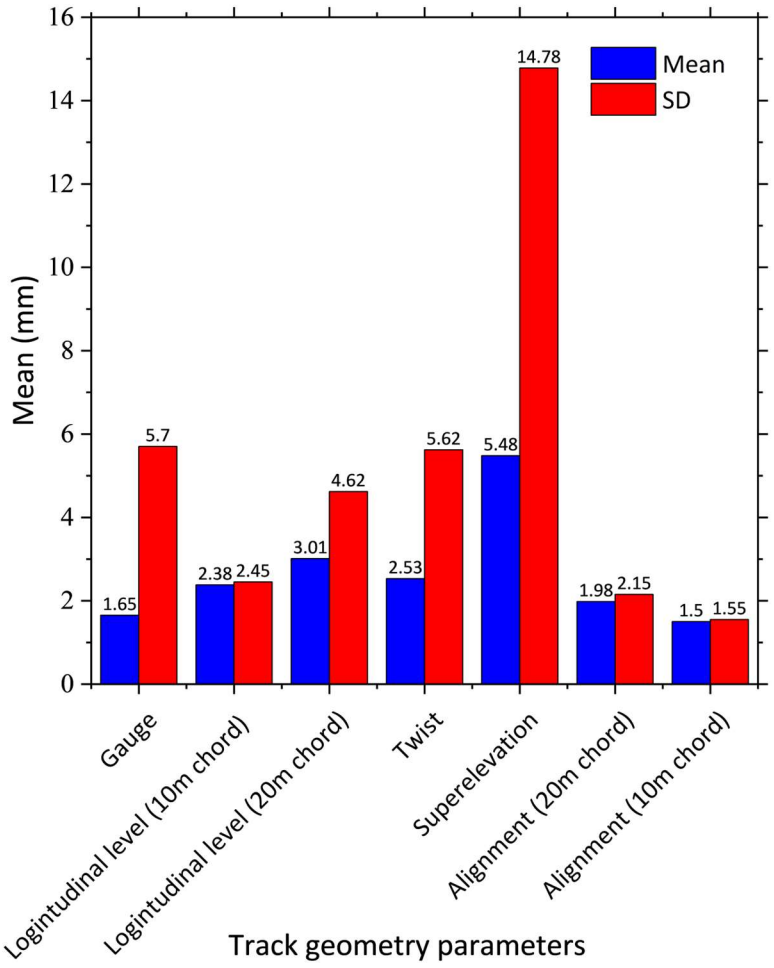


Figure 7-5 Track geometry parameter improvement when tamping is performed

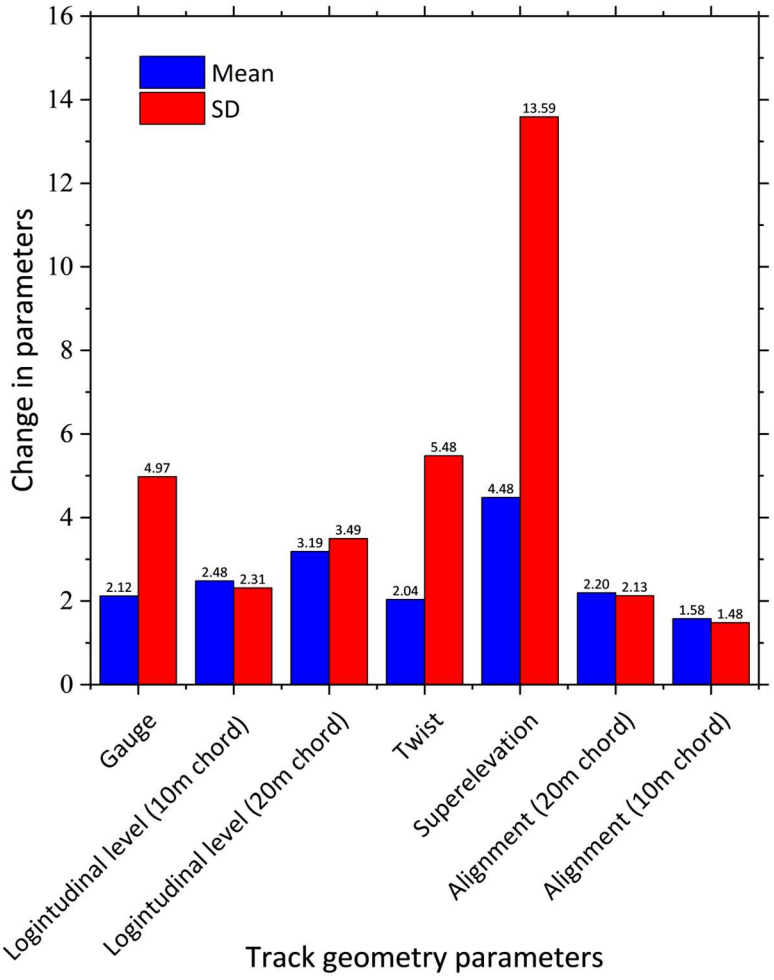


Figure 7-6 Track geometry parameter deterioration when tamping is not performed

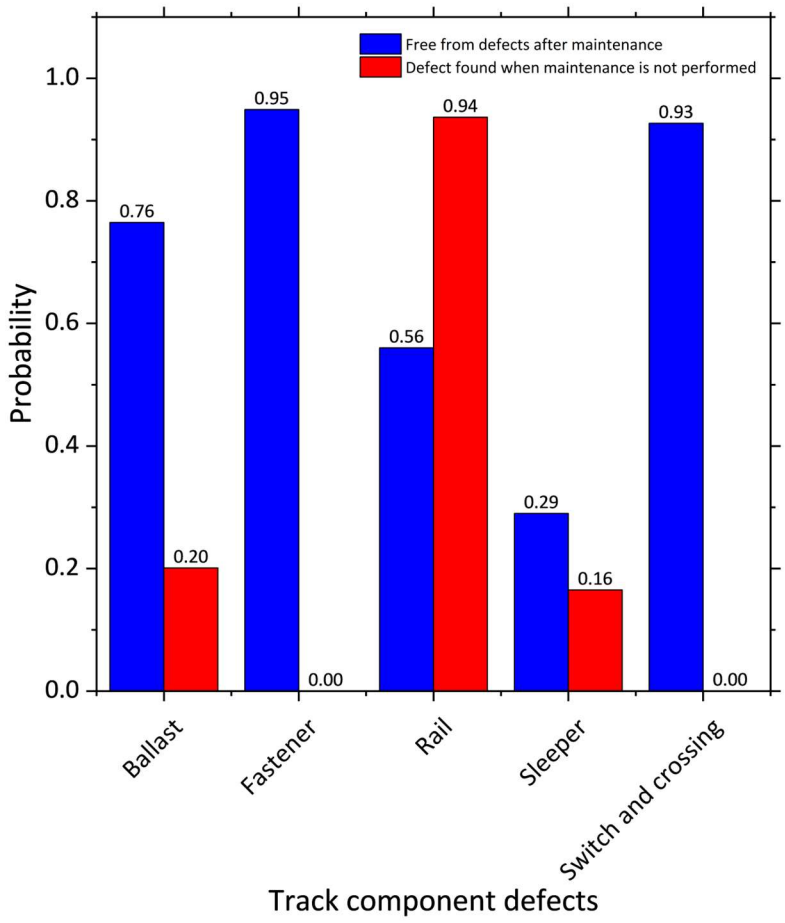


Figure 7-7 Changes in probability of defect occurrence when rail grinding is and is not performed

The customized environment of the model is developed based on the problem in this section. Five components of the models which are the agent, action spaces, states, environment, and rewards are designed and developed based on the characteristic of the problem and the data.

7.1.1.4 Results and discussion

From the model development, the reinforcement learning agent is trained to find the optimized maintenance activities based on provided states which are track geometry parameters and component defect occurrences. The agent is trained 100,000 times before being tested with the field data. Hyperparameter tuning is also performed to ensure the performance of the model. The technique used to develop the reinforcement learning model is A2C. From the training, loss and policy entropy are shown in Figure 7-8 and Figure 7-9 respectively. These two values are expected to be zero when the performance of the model is good. From the figures, the loss is close to zero when the model has been trained 5,000 times and the policy entropy is close to zero when the model has been trained 15,000 times.

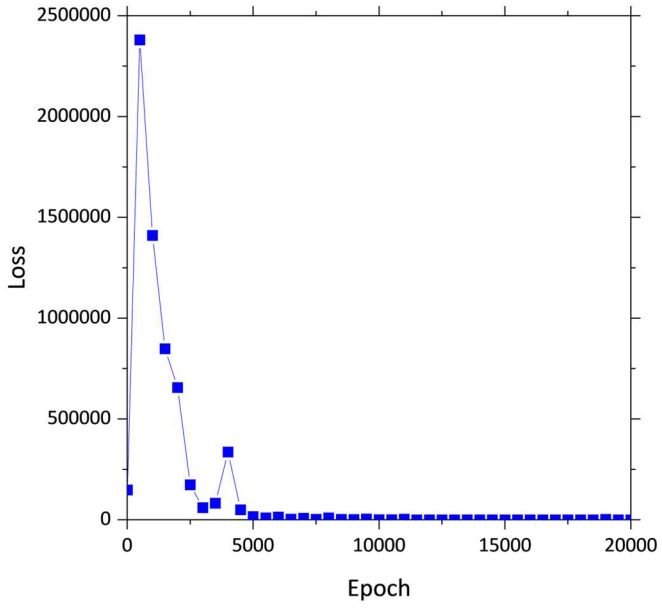


Figure 7-8 Loss from A2C model training

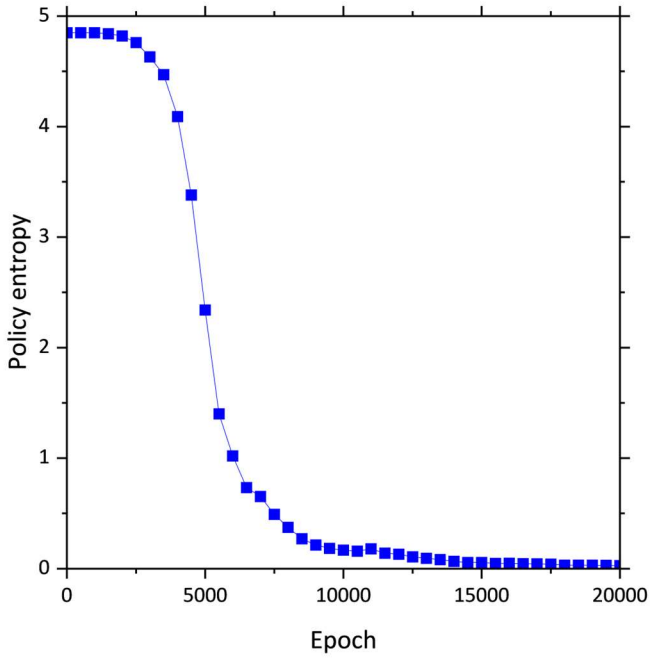


Figure 7-9 Policy entropy from A2C model training

From the hyperparameter tuning, three targeted hyperparameters which are learning rate, epsilon, and discount factors have the optimized values of 0.007, 1e-5, and 0.99 respectively.

To evaluate the performance of the developed model, the comparison between the field data and the results from the reinforcement learning model is conducted. The results from the comparison are shown in Figure 7-10. From the figure, the number of maintenance activities and defects significantly decreases. For maintenance activities, the use of a reinforcement learning model can reduce the number by 20.76% or from 963k to 763k. For the number of defects, the reinforcement learning model can reduce the number by 68.42% or from 520k to 164k. This indicates that the use of reinforcement learning can improve the efficiency of railway

infrastructure maintenance by reducing the number of defects while the maintenance activities are less required. From the preliminary analysis, the current maintenance planning is not efficient because some track sections with good conditions are excessively maintained while some track sections with bad conditions are not maintained sufficiently. The use of reinforcement learning can provide the benefit to deal with this issue by indicating the sections that really need maintenance. From the results, the number of maintenance activities is reduced by 20% while they can reduce the defect occurrence by up to 68% which identifies the potential of reinforcement learning in improving maintenance efficiency. On average, the numbers of defects are about 2 defects/section and 0.5 defects/section in the case of the field data and reinforcement learning. The average numbers of maintenance activities are 3.4 activities/section and 2.7 activities/section in the case of the field data and reinforcement learning.

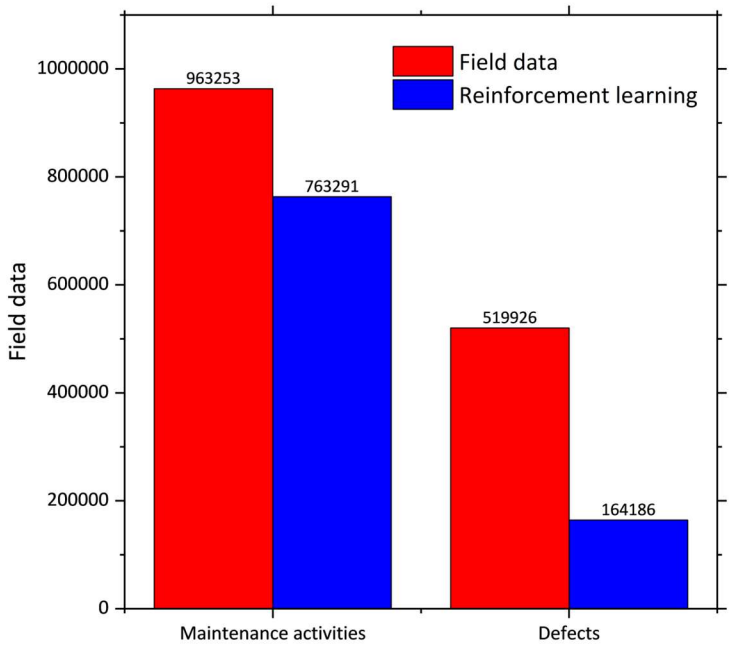
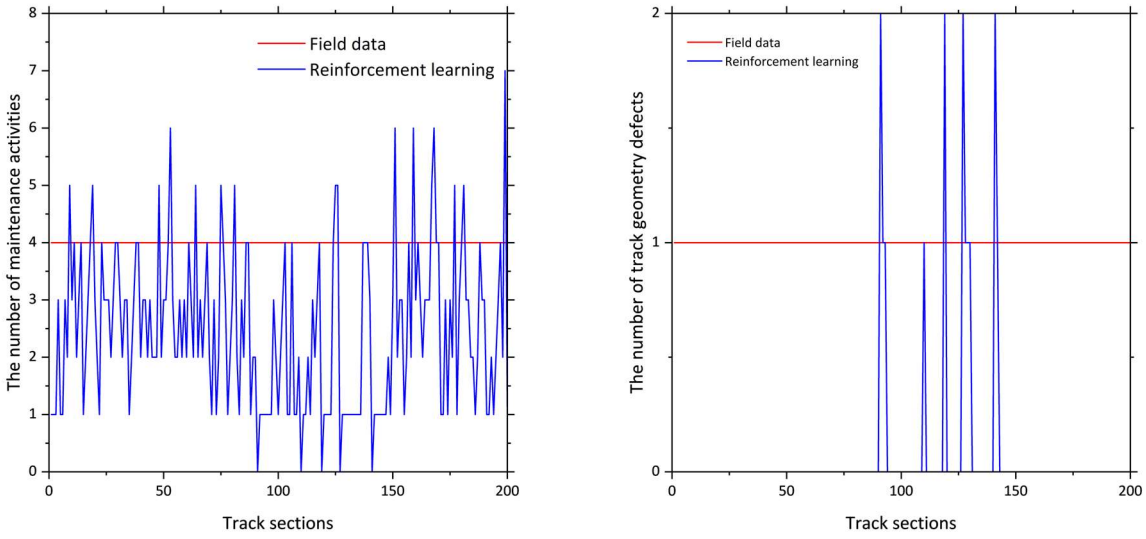
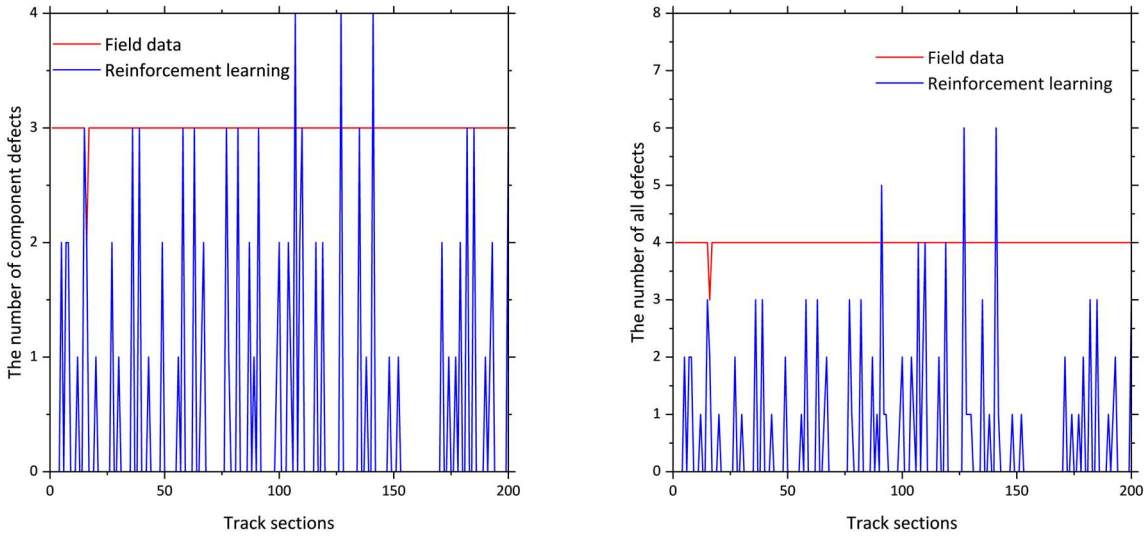


Figure 7-10 The comparison between field data and the A2C model

To provide more detail about the maintenance efficiency improvement, examples of the first 200 sections of the track are demonstrated in terms of the number of maintenance activities, track geometry defects, track component defects, and total defects as shown in Figure 7-11. The figures demonstrate both the filed data and results from reinforcement learning. From the maintenance activities, the number from the field data is constant which is four which means the maintenance is not done based on the track condition but it follows that typical maintenance plan. The results from reinforcement learning are a better response to the current condition of the track. The number of defects is also reduced from the figures. It is worth noting that although the number of defects from reinforcement learning is higher than the field data in some cases, the results from reinforcement learning are normally better than the field data. This phenomenon can refer to the stochastic behaviour in the railway industry in that defect occurrence is difficult to predict.



(a) (b)



(c) (d)

Figure 7-11 Examples of the efficiency improvement in railway maintenance (a) maintenance activities, (b) track geometry defects, (c) track component defects, and (d) all defects

For the track geometry parameter, superelevation is used as the example to demonstrate the improvement in track geometry as shown in Figure 7-12. From the railway operator manual, the threshold of superelevation at level 4 is 16 mm. Superelevation from the field data all exceed the threshold of 16 mm while the results from reinforcement learning rarely exceed the threshold. This phenomenon is also found in other track geometry parameters indicating that the use of reinforcement learning can improve the railway infrastructure predictive maintenance in every aspect.

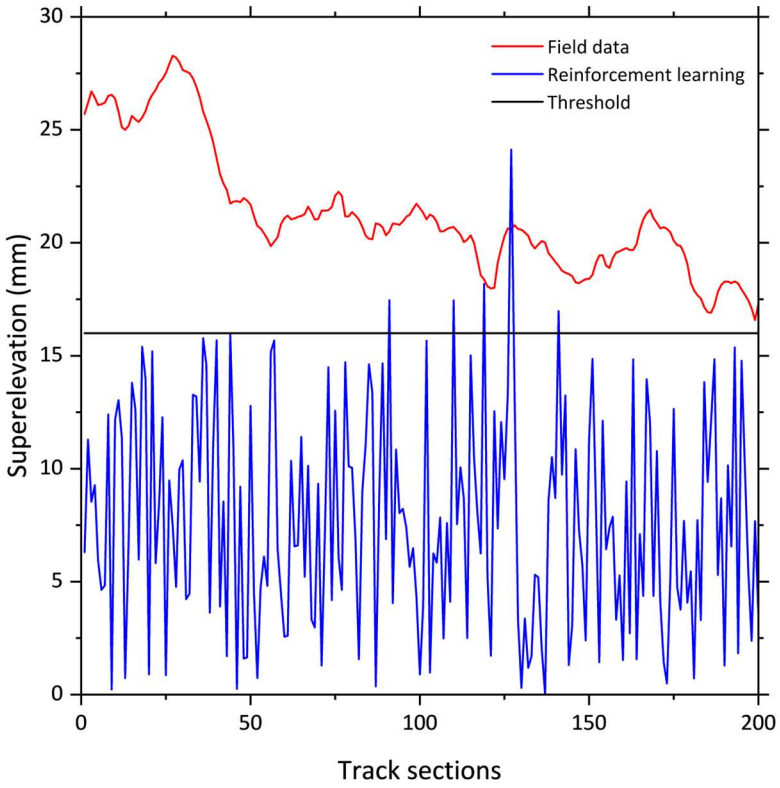


Figure 7-12 Examples of superelevation from the A2C model and field data

From the sensitivity analysis, the results are shown in Figure 7-13. Hyperparameters are plotted compared to the losses. As mentioned, the losses are expected to be zero which indicates the good performance of the model. From the figures, it can be concluded that the learning rate should not be higher than 0.5 and the epsilon should not be higher than 0.01 otherwise the performance of the model will drop. However, the discount factor does not affect the performance of the model.

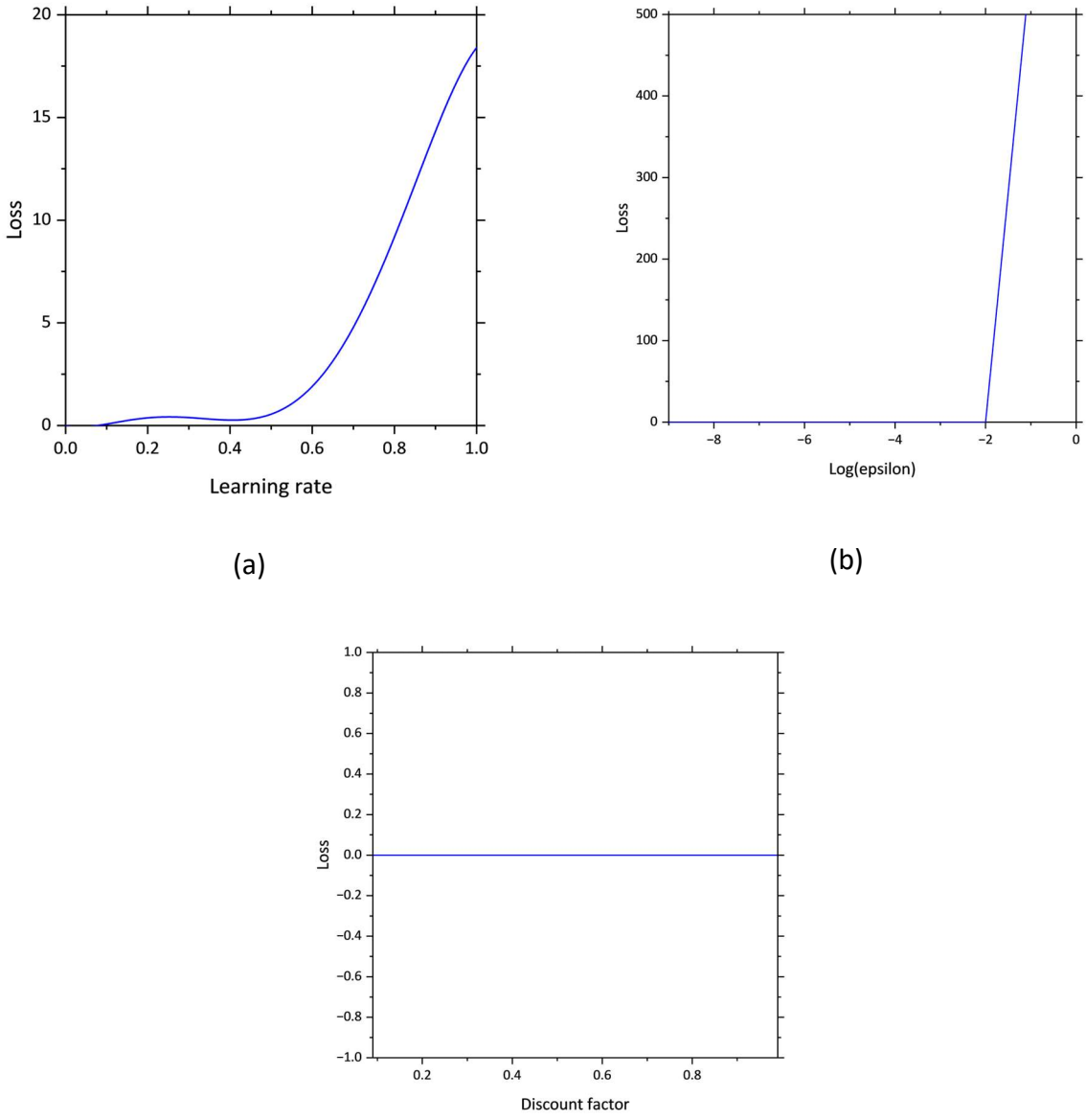


Figure 7-13 Sensitivity analysis of (a) learning rate, (b) log(epsilon), and (c) discount factor

From the results, it can be concluded that the developed model has the potential to improve the maintenance efficiency of railway infrastructure. The model can provide a maintenance plan that

reduces the number of required maintenance activities and defects. The model can save maintenance costs, the track will have a better condition, and railway operators need less possession time to perform the maintenance. Railway operators can apply the model or the concept in this section to their data and prepare the maintenance plan. The model can be used as the guideline for railway maintenance or the data-driven tool when the data are more comprehensive and the model is trained by using their data. The model can be further developed to better reflect the realistic operation such as the limitation of time, cost, and resources for maintenance.

7.2 Carbon emission reduction for railway maintenance

7.2.1.1 Introduction and backgrounds

This section is an extensive study based on Section 7.1. Different from the previous section, this section focuses on carbon emission reduction from railway maintenance. Global warming is an important issue that is prioritized by the world community. Greenhouse gas is one of the causes of global warming and climate change. It retains the heat from the sun and raises the surface temperature of the earth which can lead to droughts, heatwaves, rising sea levels, storms, and loss of biodiversity. This section will select carbon dioxide as a representative of greenhouse gas because it is the majority of greenhouse gas.

Although the railway system is recognized as one of the environmentally friendly transportation, it also generates carbon emissions through its activities such as planning, design, construction, operation and maintenance, and decommissioning. Naturally, the service life of railway projects

is long so they can generate a significant amount of carbon emission. The carbon emission from the maintenance stage of railway projects is about 6% of the total carbon emission of the whole service life [179]. That is the reason why carbon emissions from railway maintenance cannot be ignored and carbon emission reduction can create benefits in terms of the environmental aspect. This section aims to develop a reinforcement learning model for reducing carbon emissions from railway maintenance activities. Different from the previous section that used A2C to develop the model, this section will explore the potential of PPO. There are studies that identified that PPO is powerful when the model is more stable compared to other reinforcement learning techniques. Therefore, PPO will be less sensitive when it deals with data with high uncertainty or stochastic as the railway system. Less sensitivity also refers to the better capability to deal with changeable parameters. PPO also requires a shorter time to train although the environment is complicated which is suitable for the railway system. In addition, PPO is one of the most advanced reinforcement learning algorithms and it can maximize rewards better although the training stages are few [180].

The data used to develop the environment and the PPO model are similar to Section 7.1 which is the field data. The expected contributions of this section are to reduce the number of defects and required maintenance activities. At the same time, the model aims to reduce carbon emissions from railway maintenance activities. From the literature review, a study using reinforcement learning to reduce carbon emissions from railway maintenance activities has not been done. Therefore, this section aims to fill this research gap and, at the same time, explore

the potential of reinforcement learning in carbon emission reduction apart from the maintenance efficiency improvement demonstrated in the previous section.

7.2.1.2 Methodology

This section applies PPO to develop the reinforcement learning model. PPO is developed to improve the efficiency and stability of the policy gradient method which updates the policy to maximize the rewards. However, the main disadvantages of the policy gradient method are it is sensitive to hyperparameters and it requires big samples to train. PPO resolves these issues by involving a surrogate objective function that is easier to optimize. To make the training more stable, PPO also includes clipping to limit the change of policy. As a result, PPO is sophisticated and outperforms other state-of-the-art reinforcement learning techniques [181].

The training process of the PPO model is shown in Figure 7-14. The main components of the model are similar to Section 7.1.1.2. The main difference is the rewards in this section are calculated based on carbon emissions and the number of defects. The states consist of 12 states and there are seven available actions in the action space as in the previous section. Carbon emissions are calculated based on the carbon emissions created by electricity used for maintenance and the carbon emissions from material production. Carbon emissions from electricity are calculated based on the duration of each maintenance activity and machine that can be obtained from manufacturer manuals. For material production, the carbon emissions are calculated based on the quantity required for each maintenance activity.

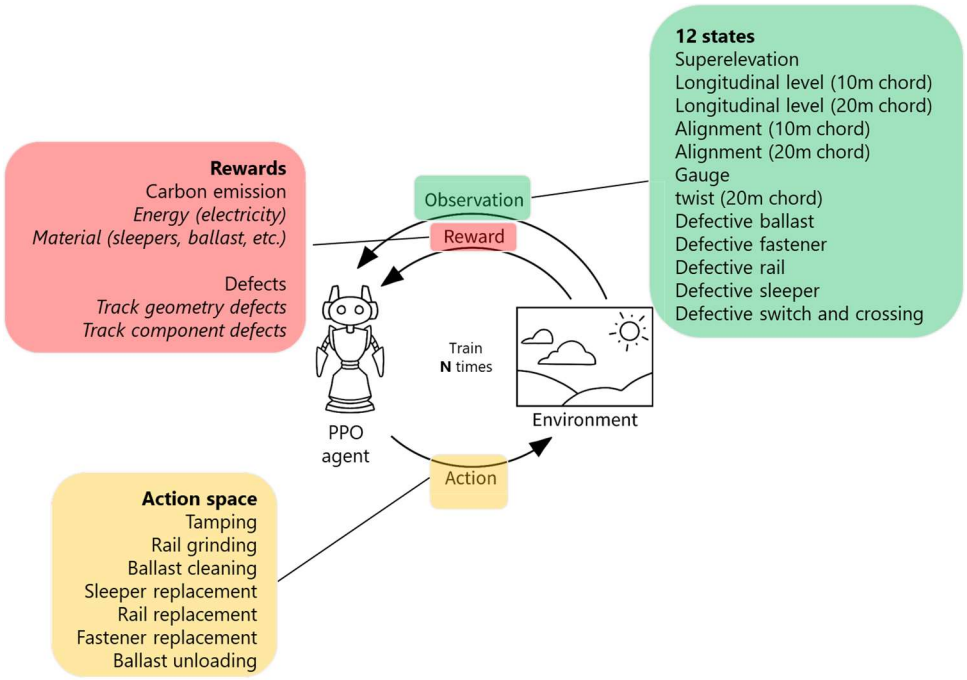


Figure 7-14 Overview of training process of the reinforcement learning model for carbon emission reduction

The calculation of carbon emissions is shown in Table 7-1. The speed of the maintenance is considered based on the manufacturer manuals. Carbon emissions from electricity are calculated based on the speed of the maintenance. In this section, the length of each track section is one foot. The carbon emissions generated from electricity generation is 0.509 kg CO2e/kW [182]. For material production, carbon emissions are calculated based on the quantities of each material required for each maintenance activity. The carbon emissions from the production of ballast, sleepers (70 kg of concrete), and steel are 3.4 [183], 0.124 [184], and 1.85 [185] kg CO2e/kg respectively.

Table 7-1 Carbon emission calculation for railway maintenance

Maintenance activities	Speed (km/h)	Duration for a section (h)	Power (kW)	Energy consumption (kWh)	CO2e from Electricity (kg CO2e)	Type of material	Material use	CO2e from material (kg CO2e)	Total CO2e (kg CO2e)
Tamping	56 [186]	5.36E-06	130.5 [186]	6.99E-04	3.56E-04	-	-	-	3.56E-04
Rail grinding	8 [187]	3.75E-05	120.0 [187]	4.50E-03	2.29E-03	-	-	-	2.29E-03
Ballast cleaning	48 [188]	6.25E-06	838.9 [188]	5.24E-03	2.67E-03	Stone	2,900 kg [188]	6.16E-06	2.67E-03
Sleeper replacement	48 [189]	6.25E-06	36.0 [189]	2.25E-04	1.15E-04	Concrete	1 set	8.68	8.68E+00
Rail replacement	160 [190]	1.88E-06	20.0 [190]	3.75E-05	1.91E-05	Steel	0.6 m	66.6	6.66E+01
Fastening components replacement	-	-	-	-	-	Steel	2 sets (19.48 kg) [191]	36.04	3.60E+01
Ballast unloading	5 [192]	6.00E-05	1491.4 [192]	8.95E-02	4.55E-02	-	-	-	4.55E-02

7.2.1.3 Data and processing

The data used in this section and how they are processed are referred to in Section 7.1.1.3.

7.2.1.4 Results and discussion

The number of iterations set for the reinforcement learning to be trained is 100,000. Figure 7-15 and Figure 7-16 present the loss and policy entropy from the model training. As mentioned, the expected values of both are zero. If the values are close to zero, it indicates that the model has been optimized. The figures demonstrate the loss and policy entropy from the first 40,000 iterations that the model has been optimized even though the model is not trained for 100,000 iterations.

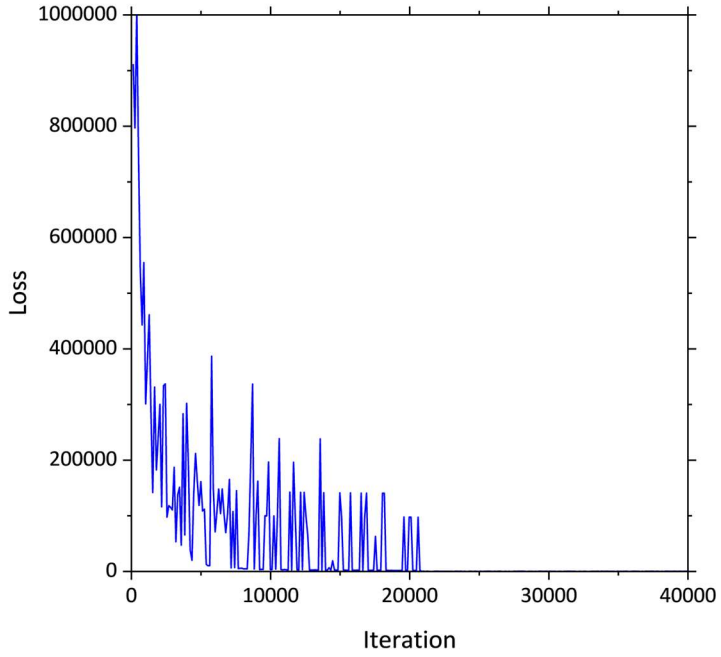


Figure 7-15 Loss from PPO model training

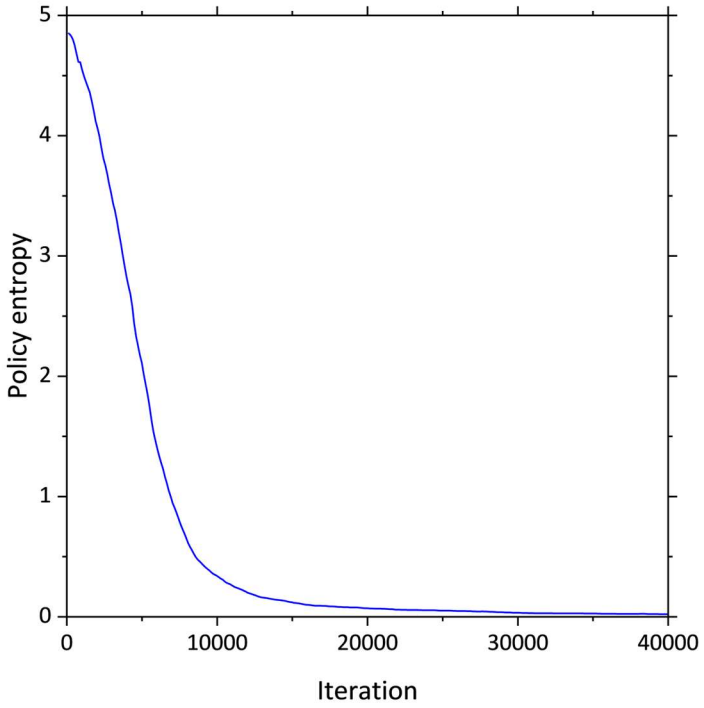


Figure 7-16 Policy entropy from PPO model training

Figure 7-17 demonstrates the comparison between the results from the field data and the reinforcement learning model developed using PPO. It can be seen that the use of the PPO model can reduce the number of required maintenance activities, the number of defects, and carbon emissions from maintenance activities. For the number of required maintenance activities, the PPO model can reduce the number of maintenance activities by 48% although the purpose of the model is defined to reduce carbon emissions. For the number of defects, the PPO model can reduce the number by 68%. Finally, the PPO model can reduce the amount of carbon emissions from railway maintenance in the form of material use and electricity by 65%. It can be seen that the PPO model can improve the efficiency of railway maintenance in every aspect such as engineering and environmental aspects.

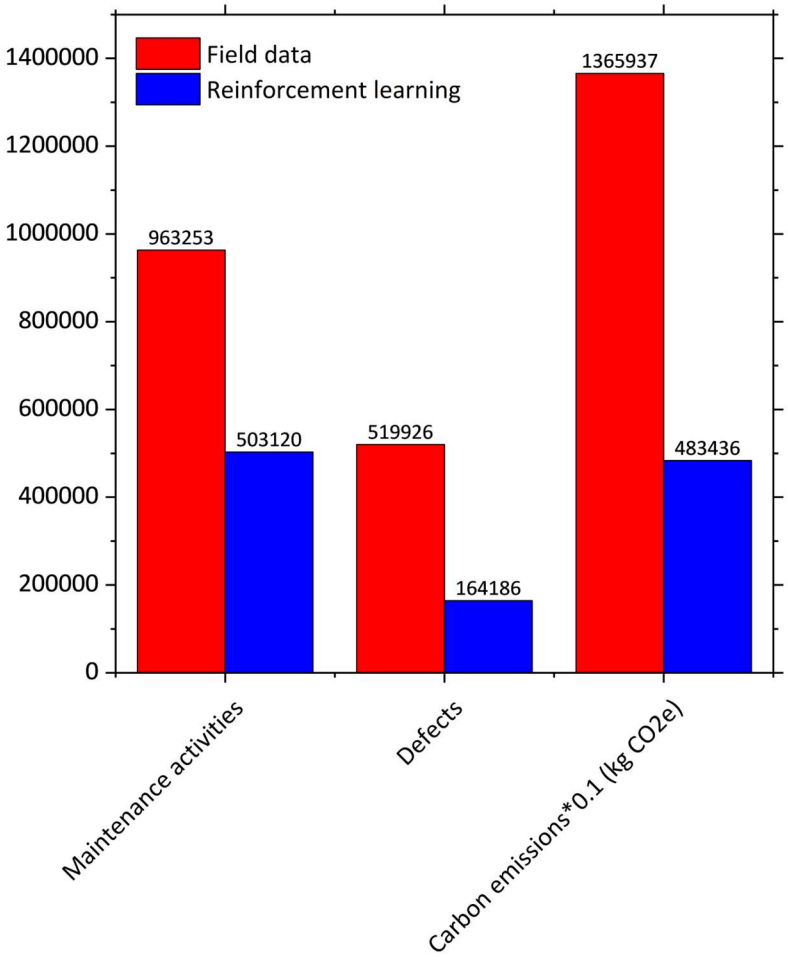
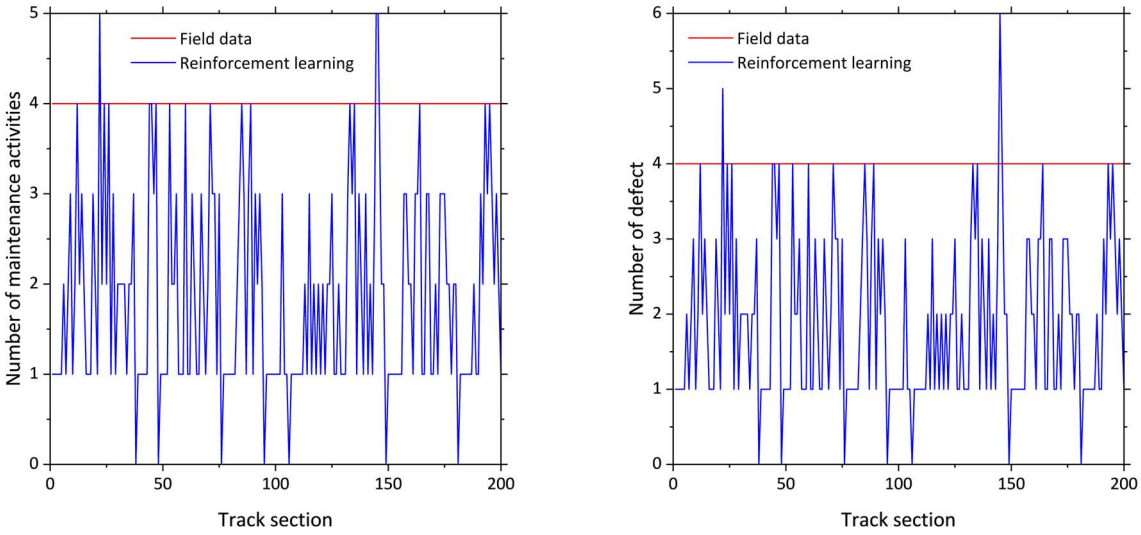


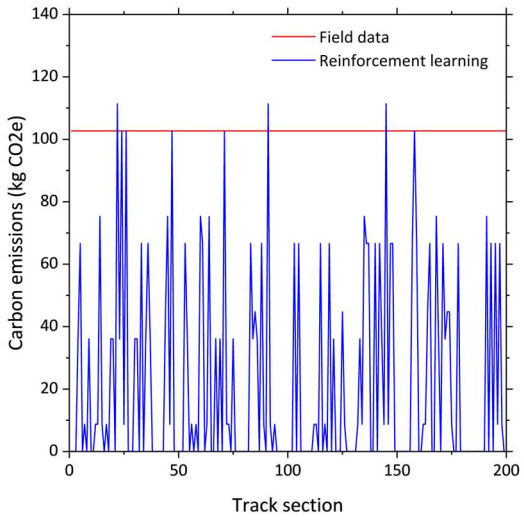
Figure 7-17 The comparison between field data and the A2C model

A detailed comparison between the field data and the PPO model is shown in Figure 7-18. The first 200 track sections are presented. As the results found in Section 7.1, the use of reinforcement learning can improve maintenance scheduling in every aspect. Although some results from the PPO model are worse than the field data, most of the results from the PPO model

are better than the field data or the current practice. The overall results from the PPO model indicate an improvement compared to the field data.



(a) (b)



(c)

Figure 7-18 Examples of the efficiency improvement in carbon emission reduction (a) maintenance activities, (b) defects, and (c) carbon emission

Figure 7-19 presents the differences in superelevation from the field data and the PPO model. The superelevation's threshold is 16 mm according to the railway operator's manual. The superelevation from the field data is higher than the threshold in every track while a few from the PPO model is higher than the threshold. This presents the potential of reinforcement learning in carbon emission and defect reduction. When comparing the result in Figure 7-19 to Figure 7-12, the results might look similar however carbon emissions are also considered for the result in Figure 7-19. In theory, the superelevation result in Figure 7-19 should be worse than Figure 7-12. However, PPO is more advanced than A2C used for Figure 7-12 so the superelevation and other track geometry parameters are not worse than the model considering only defects. In addition, considering carbon emissions can significantly reduce the carbon emissions in railway maintenance. Carbon emissions from Section 7.1 when the carbon emissions are not considered is 10,546,280 kg CO2e while the carbon emissions in this section are only 4,834,360 kg CO2e. Therefore, addition constraints do not spoil the model performance and some benefits can be obtained if the model is advanced enough and developed approximately.

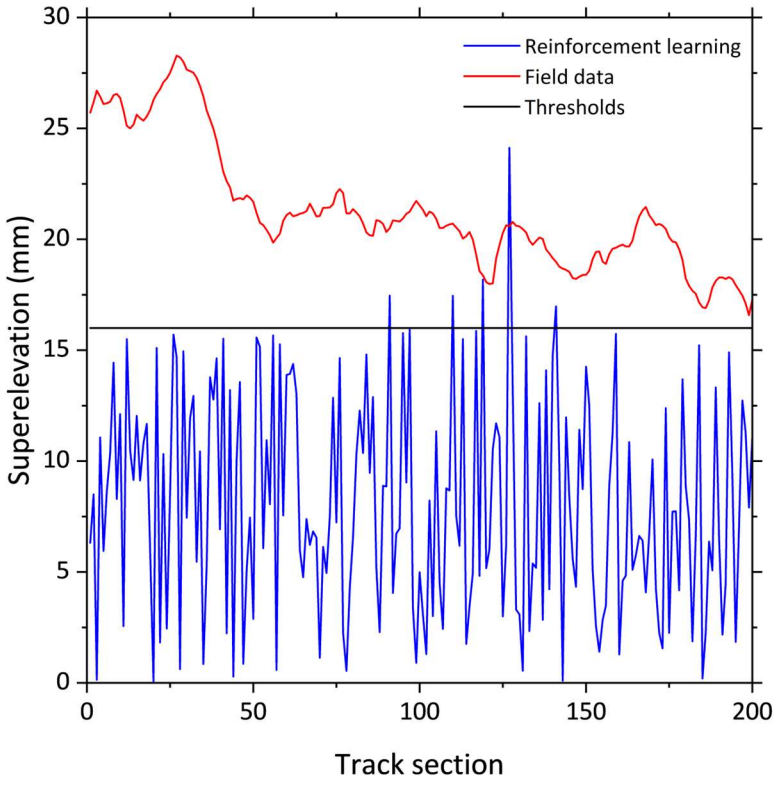


Figure 7-19 Examples of superelevation from the PPO model and field data

The findings in this section will support the decision-making of railway operators in carbon emission reduction. Moreover, railway operators can also use the developed model to improve railway maintenance efficiency in terms of maintenance and defect reduction. The model complexity can be further developed by adding some constraints to mimic the limitations of different railway projects such as the limited budget and resources. Railway operators can develop their own models using the concept proposed in this section and use their data to train the models. They can use the developed models for railway predictive maintenance planning which has proven that better than corrective and preventive maintenance.

8 6D BIM and machine learning integration and development

8.1 BIM model development and machine learning integration

BIM is used to store information related to design, quantity, schedule, and railway maintenance in this thesis. BIM is popularly used and a requirement in many countries due to its benefits. For example, the UK government requires BIM level 2 in every state project since 2016 [37]. This thesis will use the benefits of BIM not only in the design but also in the maintenance stages which can be considered the most dynamic stage because information takes place all the time. To utilize stored information, machine learning is applied as mentioned. This section presents how to integrate BIM and machine learning for railway infrastructure predictive maintenance and how to deploy the developed approach. This section will present how to develop BIM models using Civil 3D and Ferrovia, how to integrate data and information in the BIM models, how to do quantity take-off using the BIM software, how to integrate project schedules with the BIM models using Navisworks, and how to develop GUI to allow users to deploy the developed approach easier.

8.1.1 Civil 3D

Autodesk AutoCAD Civil 3D will be used to create BIM models in this thesis. It can be used to create a BIM model by itself as demonstrated in this section. In the beginning, BIM models are developed as 3D models. Then, information will be included and models are further developed to meet the aim of 6D BIM models. The additional dimensions of BIM models in this thesis are

cost and quantity, time or schedule, and maintenance aspects for the fourth, fifth, and sixth dimensions.

The process of BIM model development starts with creating horizontal and vertical alignments of railway projects. In this step, different characteristics of railway projects are defined such as superelevation, intersection, curve, radius of curvature, gradient, and turnout. Horizontal and vertical alignments are designed based on the geography and topography of project areas. They have to comply with standards and design codes such as the maximum gradient and radius of curvature that refer to design speed. Civil3D has a function to select a specific standard that railway projects need to comply with. The software will detect parameters that do not comply with the standard and show a warning informing issues. An example of a warning is users have to input the design speed of the railway project and select a standard they want to comply with. Then, if the software detects any curves that are toll small compared to the design speed, the software will show warnings at those curves. After horizontal and vertical alignments are designed, users need to define sample lines. Sample lines are used to represent crossing sections along railway routes. The distance between sample lines can be defined. Sample lines are also used to do the quantity take-off for cut and fill earthwork. Based on the sample line, assemblies are defined to design the cross-sectional characteristics of the railway routes. The foundation, track structure, and other components such as rails, sleepers, and retaining wall are defined using assembly. The last step is to design a corridor that will combine horizontal and vertical alignments, assembly, and topography to create solid objects. An example of the model created using Civil3d is shown in Figure 8-1. Until this step, the 3D BIM models are achieved and they can

be exported to deploy in other BIM software. To do this, all solids are exported as an Industry Foundation Classes (IFC) file. IFC files can be used in any BIM software as they are standard formats for BIM. An example of an IFC file of the BIM model is shown in Figure 8-2.

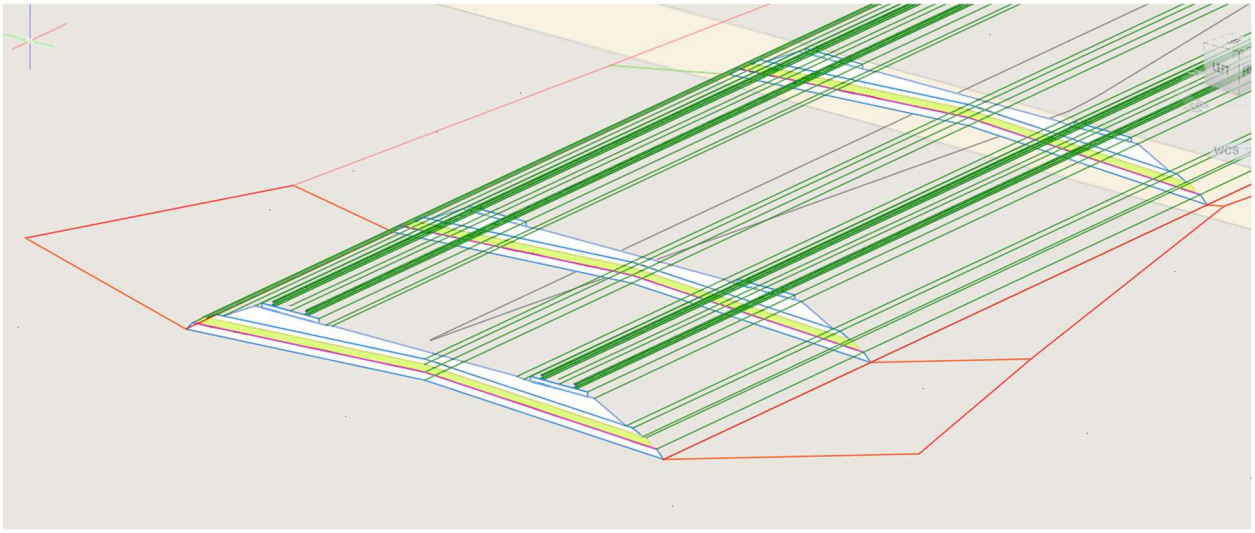


Figure 8-1 An example of a BIM model created using Civil3D

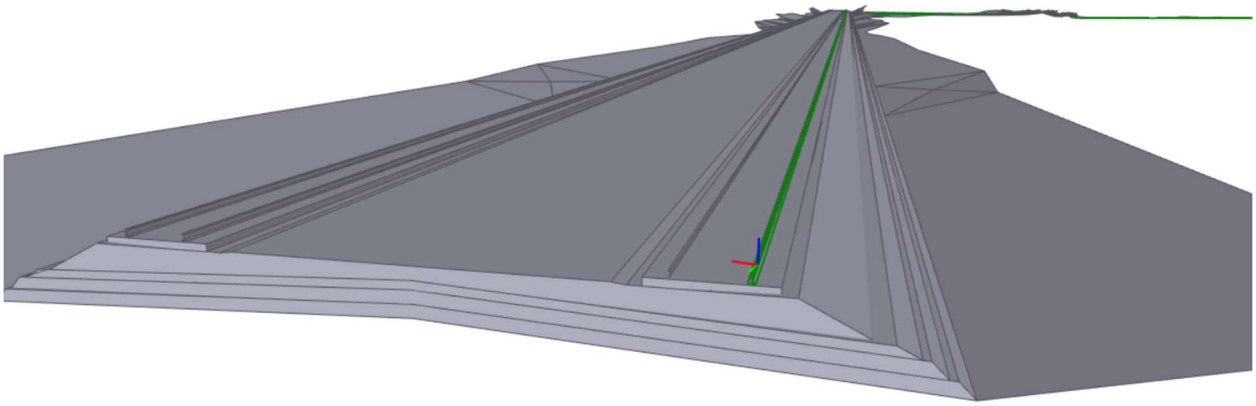


Figure 8-2 An example of a BIM model in the format of an IFC file

Solids in the BIM models can be integrated with information using property set definitions. This is the fundamental of BIM models that have the capability to integrate with any information and are developed for further dimensions. For example, design speeds of the railway route can be added using property set definitions as shown in Figure 8-3. Information stored in property set definitions can be added in the models using the schedule as shown in Figure 8-4. It is worth noting that this schedule is different from the project schedule and the time aspect. The result of the exported schedule is shown in Figure 8-5.

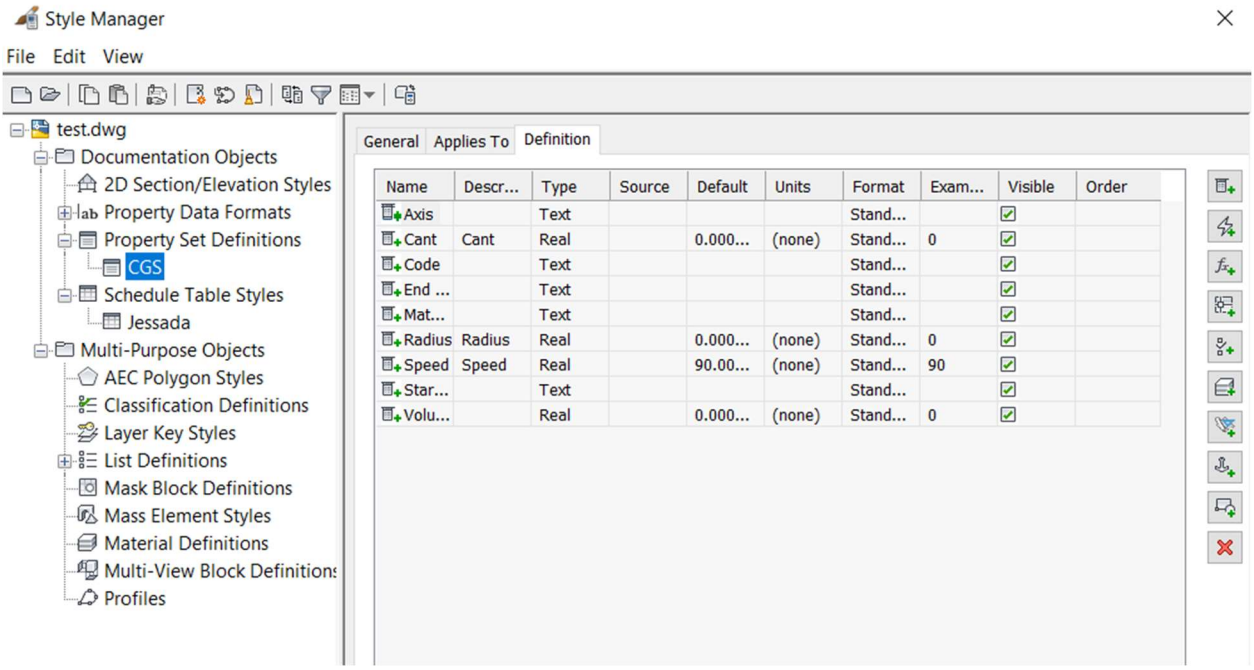


Figure 8-3 Property set definitions used to integrate information with BIM models

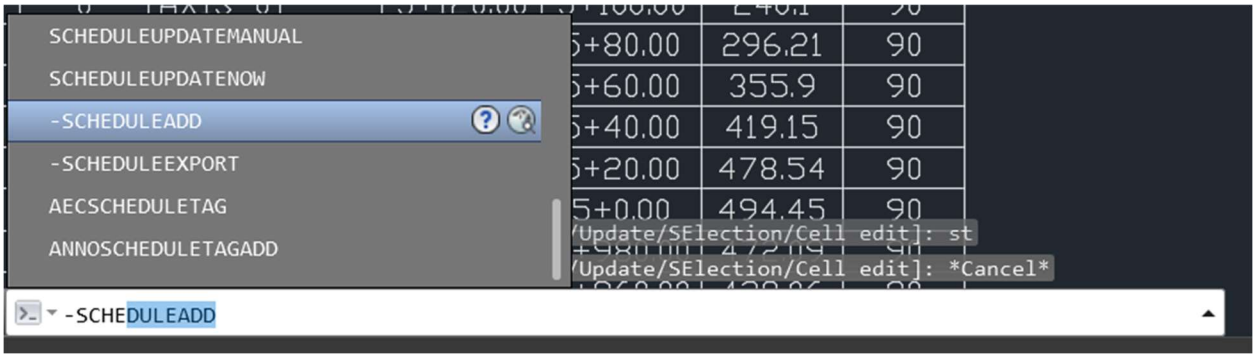


Figure 8-4 A command used to add the schedule to the BIM model

Schedule Table

Cant	Radius	Axis	Code	End	Start	Volume	Speed
0	0	AXIS_0		5+280.00	5+260.00	82.85	90
0	0	AXIS_0		5+260.00	5+240.00	109.02	90
0	0	AXIS_0		5+240.00	5+220.00	156.54	90
0	0	AXIS_0		5+220.00	5+200.00	202.04	90
0	0	AXIS_0		5+200.00	5+180.00	189.91	90
0	0	AXIS_0		5+180.00	5+160.00	147	90
0	0	AXIS_0		5+160.00	5+140.00	146.83	90
0	0	AXIS_0		5+140.00	5+120.00	187.56	90
0	0	AXIS_0		5+120.00	5+100.00	240.1	90
0	0	AXIS_0		5+100.00	5+80.00	296.21	90
0	0	AXIS_0		5+80.00	5+60.00	355.9	90
0	0	AXIS_0		5+60.00	5+40.00	419.15	90
0	0	AXIS_0		5+40.00	5+20.00	478.54	90

Figure 8-5 Result of exported schedule

The exported schedule can be exported in the form of an Excel spreadsheet file or XLS files as shown in Figure 8-6 and Figure 8-7. Civil3d has a function to do the quantity take-off for cut and fill earthwork including volume of material. Therefore, besides earthwork, other components of railway infrastructure can be taken off such as sleepers, ballasts, rails, and embankments. An

example is shown in Figure 8-5. The software calculates the volume of earthwork automatically when positive numbers represent fill earthwork and negative numbers represent cut earthwork. Until this stage, the BIM models are 4D. To further improve the BIM models to 5D and 6D, the detail will be demonstrated in 8.1.3 and 8.1.4. Then, cost estimation can be done based on the quantity take-off by the software. This also reduces the error in the quantity take-off process because it is not done manually.

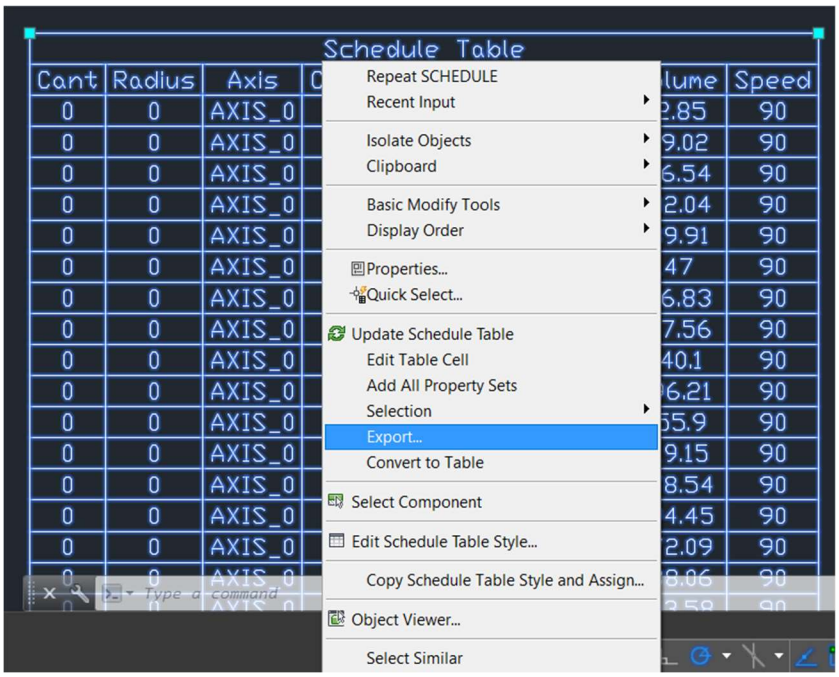


Figure 8-6 A command to export the schedule to XLS files

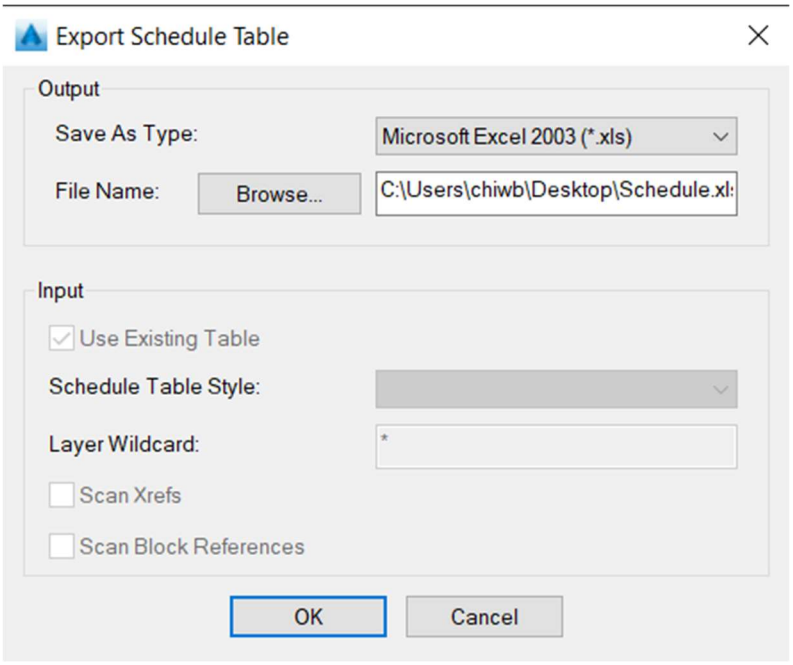


Figure 8-7 A window for configuring the exported XLS files

8.1.2 Ferrovia

Ferrovia is a software developed by CGS Labs that is used to develop BIM models for railway projects. The software includes many standards that can be applied to the design. Ferrovia can be integrated with many BIM software such as Revit, Civil3D, and BricsCAD. To develop 6D BIM models using Ferrovia, the process starts with importing the terrain data from Google Maps. The image of the terrain can also be included in the BIM models. Then, Digital Terrain Model (DTM) is created based on the imported terrain data. DTM can be created in the form of contours, boundaries, or triangles. The next step is designing the alignment. Design speeds of routes and gauges are defined in this step. The outcome from DTM and alignment creation is shown in Figure 8-8.

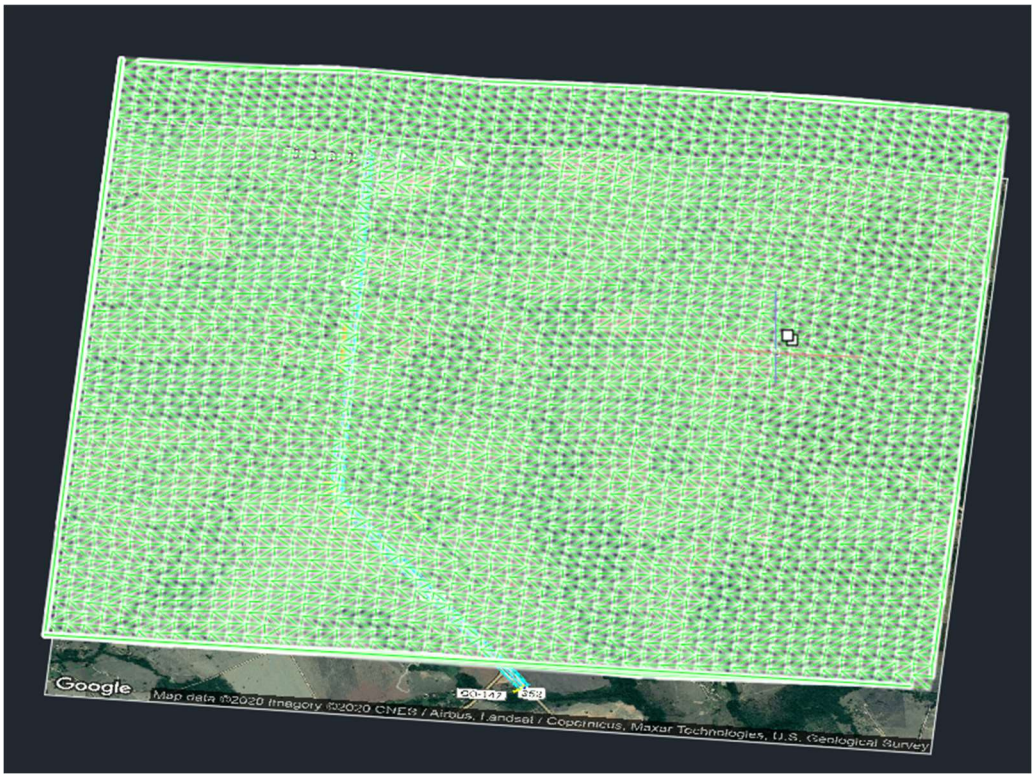


Figure 8-8 Outcome from DTM and alignment creation

Then, sample lines are created by defining the interval between sample lines as in Section 8.1.1. Sample lines are used to refer to cross sections and create cross sections along the alignment. The next step is to design the profile of the railway route. The profile will show the terrain on the alignment, then designers can create the profile based on the terrain and defined standard. If there are any issues that conflict with the standard, the software will be shown warnings. An example of the profile is shown in Figure 8-9.

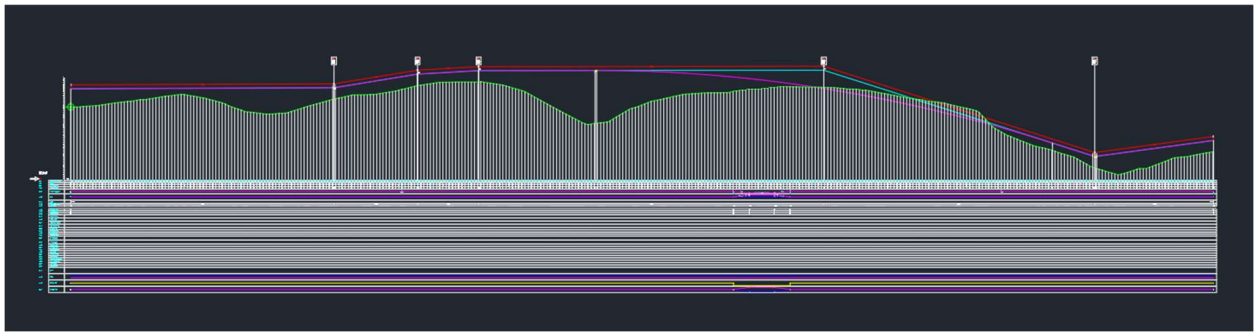


Figure 8-9 Outcome from profile creation

After that, cross-sections are created based on the alignment. In this step, railway infrastructure and components are designed. This is a critical step to define the characteristics of the railway projects. Some examples of design variation are single track, double track, and tunnels. The creation of the cross-section is shown in Figure 8-10 to Figure 8-12. In Figure 8-10, the cross-section is a single track that has only ballast supporting sleepers. In Figure 8-11, the cross-section is designed to be a double track. More railway components are included such as subgrade, subbase, and retaining wall. In Figure 8-12, the cross-section is designed as a railway tunnel. Piping, box, and envelope are included.

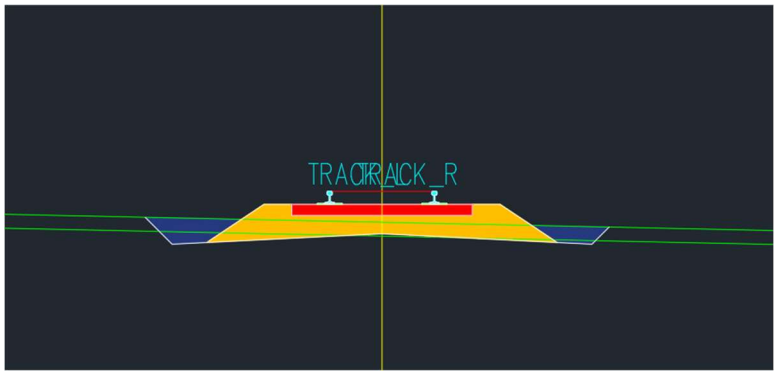


Figure 8-10 Cross section of single track

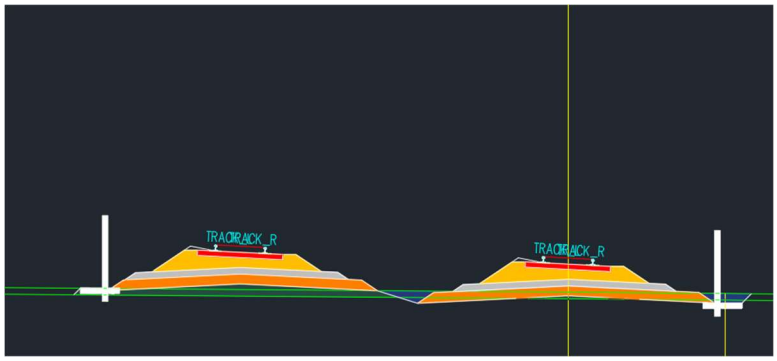


Figure 8-11 Cross section of double track

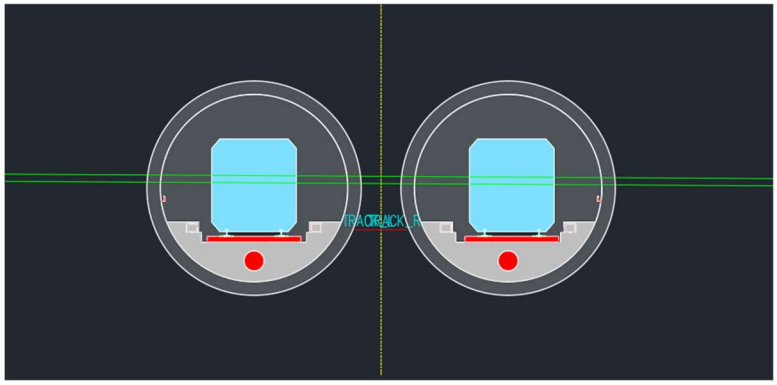


Figure 8-12 Cross section of railway tunnels

From this, 3D BIM models can be created as 3D solid as Section 8.1.1 and information can be included using property set definitions as mentioned. Examples of 3D BIM models are shown in Figure 8-13 to Figure 8-15. More components can be included such as the overhead catenary system shown in Figure 8-14. To integrate with other BIM software, the 3D BIM models can be exported as IFC files. Outcomes from the export are shown in Figure 8-16 to Figure 8-18. Ferrovia also has the function to perform quantity take-off. It can be done by selecting the target material and the software will estimate the quantity automatically. Now, the BIM models can achieve the 4D BIM models.

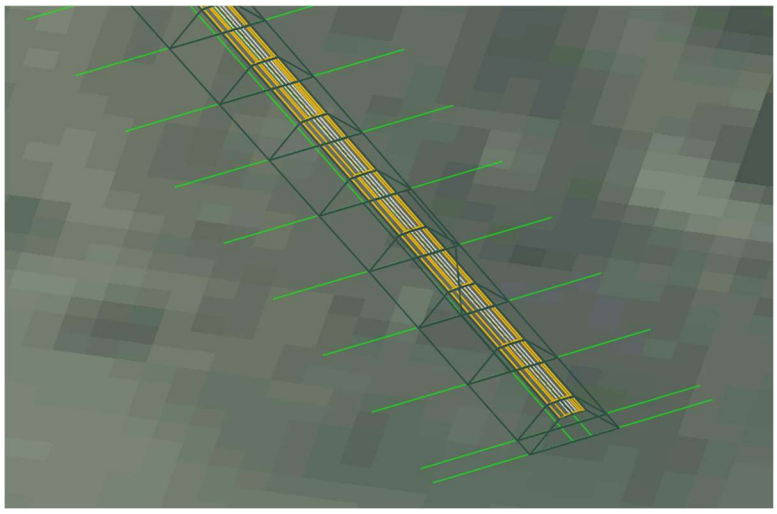


Figure 8-13 3D BIM model created using Ferrovia for single track

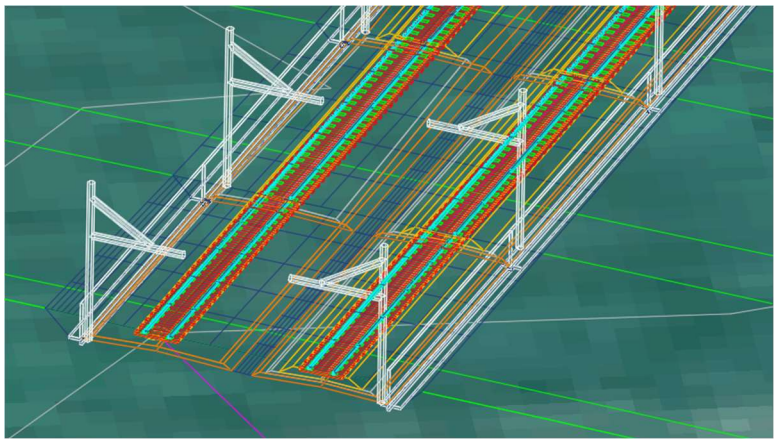


Figure 8-14 3D BIM model created using Ferrovia for double track

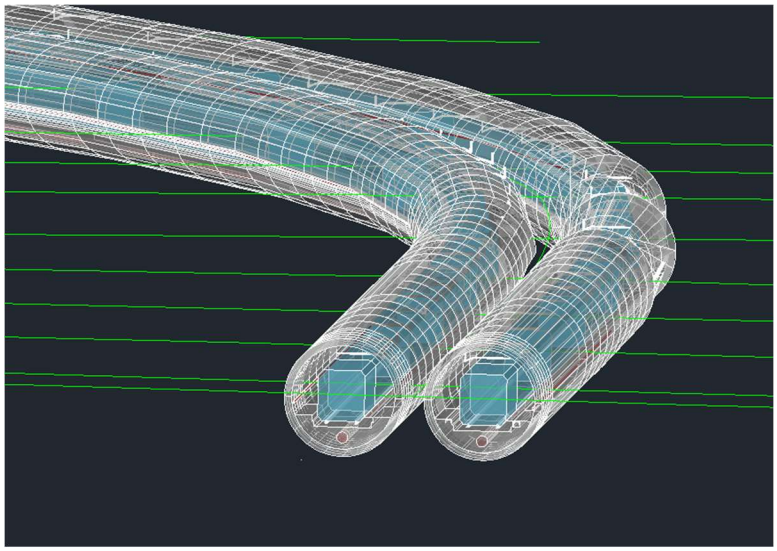


Figure 8-15 3D BIM model created using Ferrovia for railway tunnels

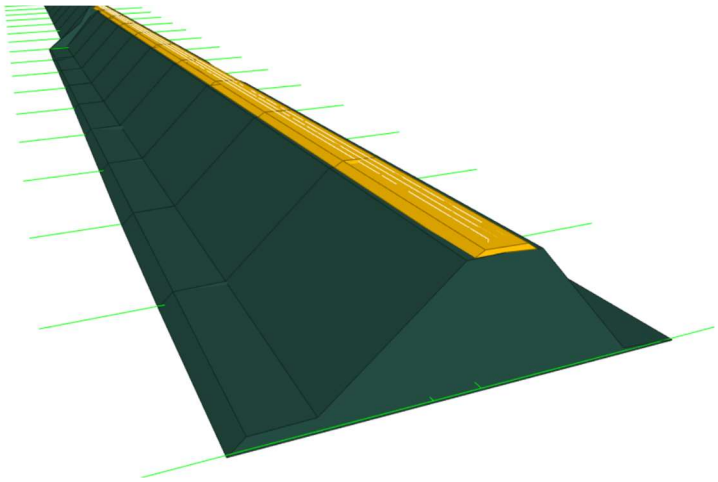


Figure 8-16 IFC file for single track

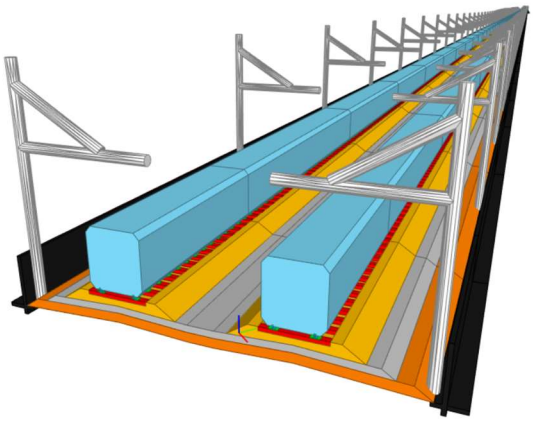


Figure 8-17 IFC file for double track

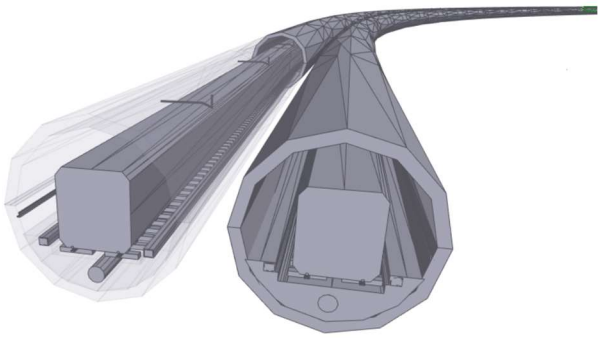


Figure 8-18 IFC file for railway tunnels

The developed BIM models can be extended for further information. Terrain and location can be included to demonstrate the area of the project as shown in Figure 8-19. At the same time, the BIM model can show the detail of the component level as shown in Figure 8-20 to Figure 8-24. In addition, the developed BIM models can be combined with other BIM models to demonstrate the wider view of the BIM models such as BIM models of other buildings to develop the BIM model at the city level as shown in Figure 8-25.

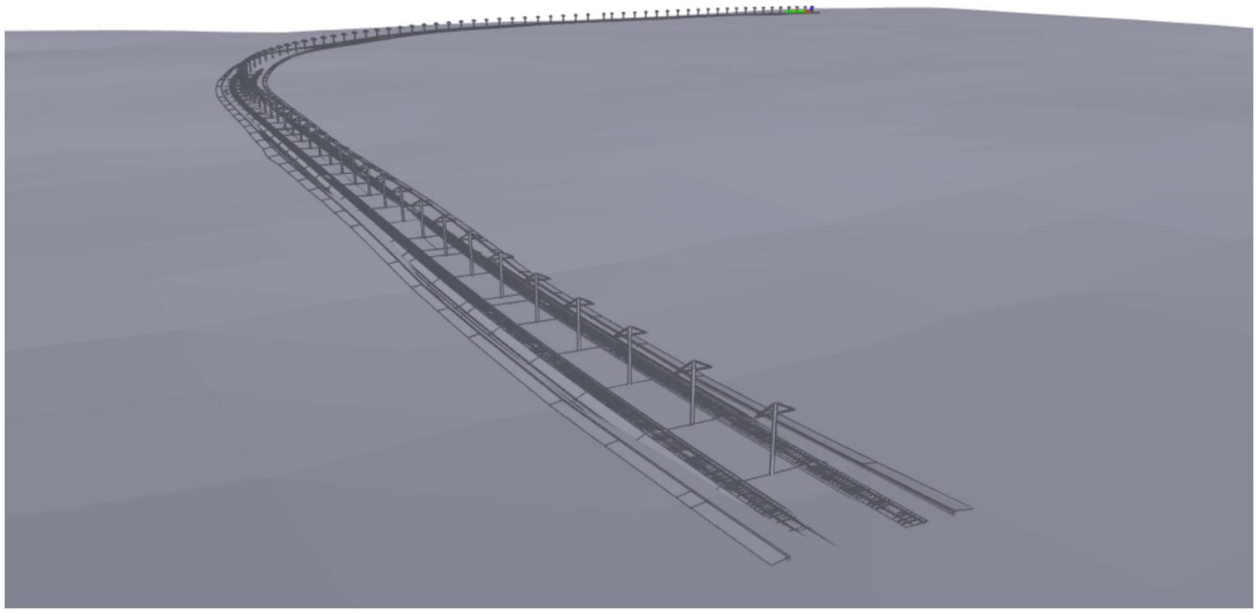


Figure 8-19 BIM model with included terrain

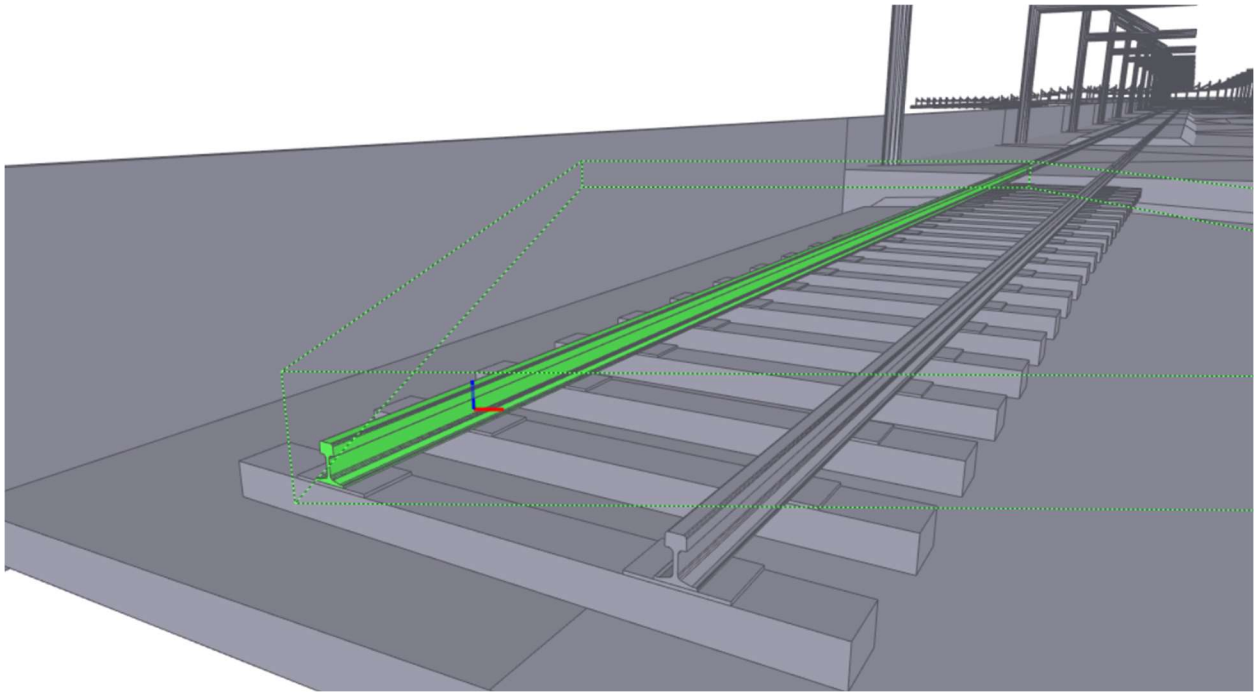


Figure 8-20 BIM model of the rail

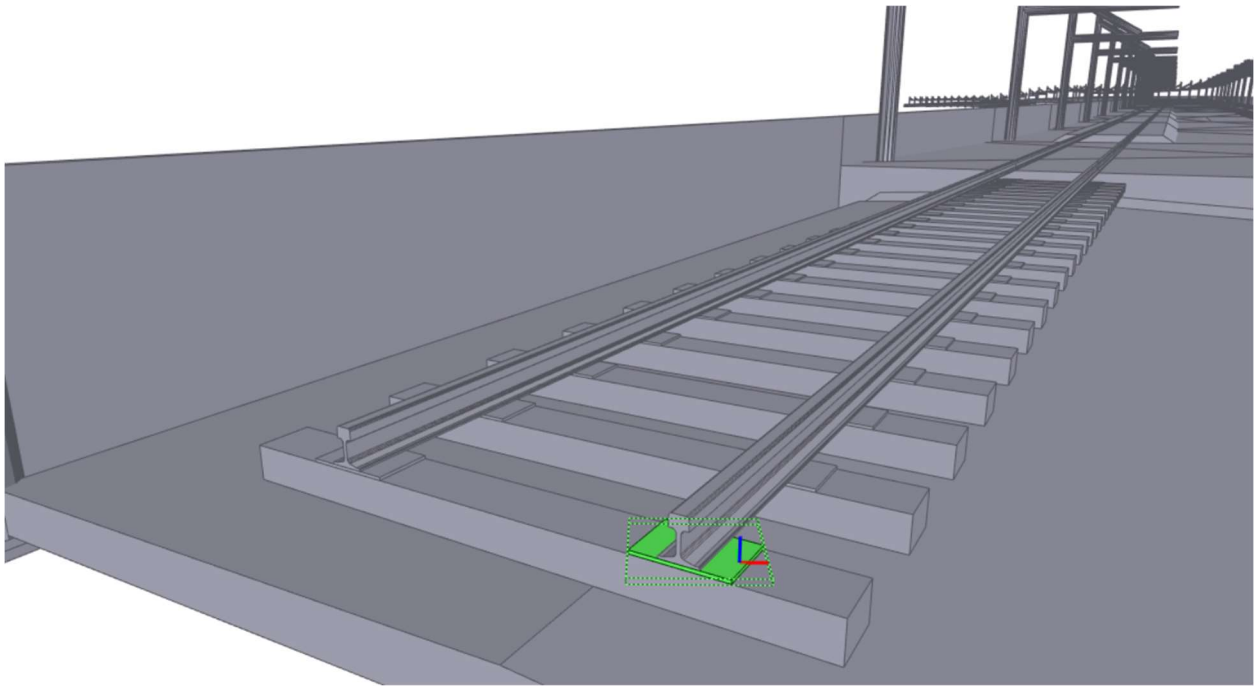


Figure 8-21 BIM model of the base plate

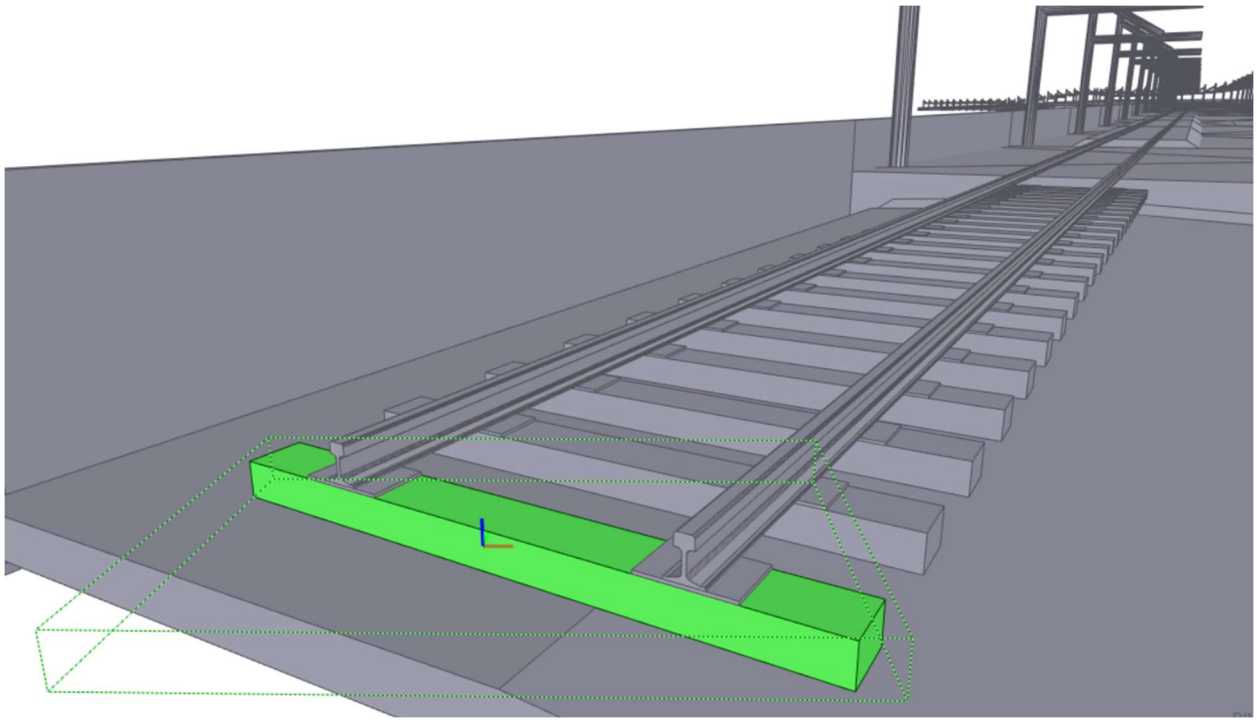


Figure 8-22 BIM model of the sleeper

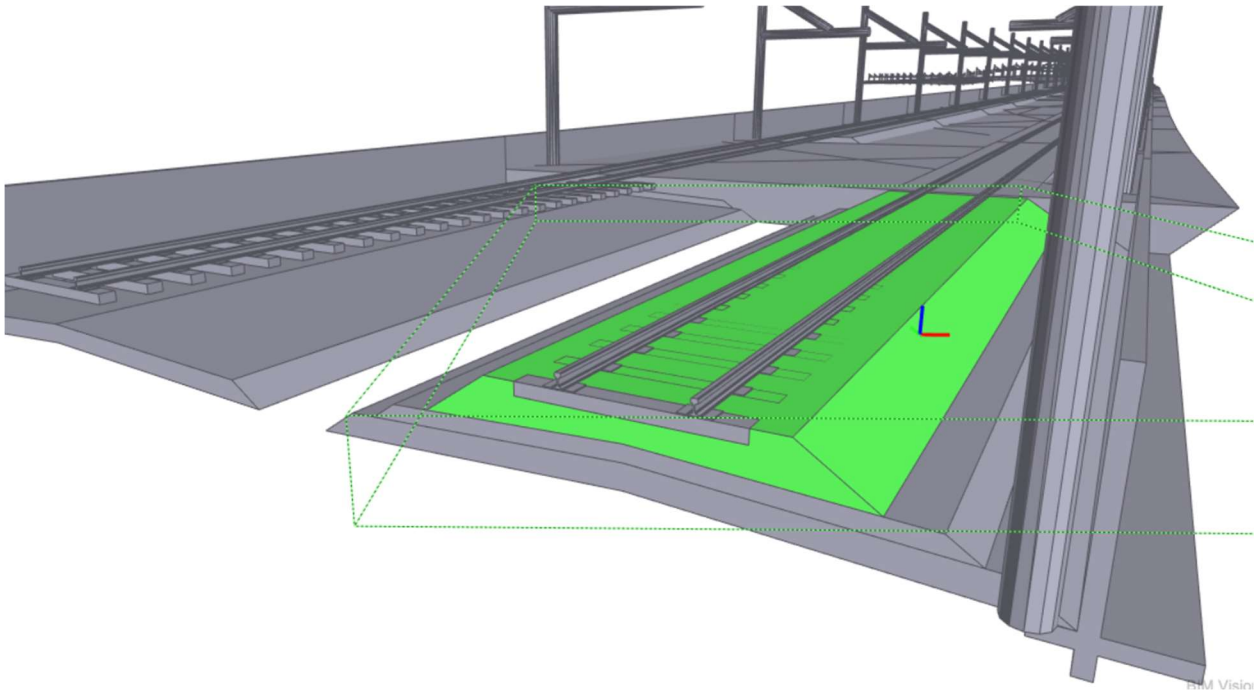


Figure 8-23 BIM model of the ballast

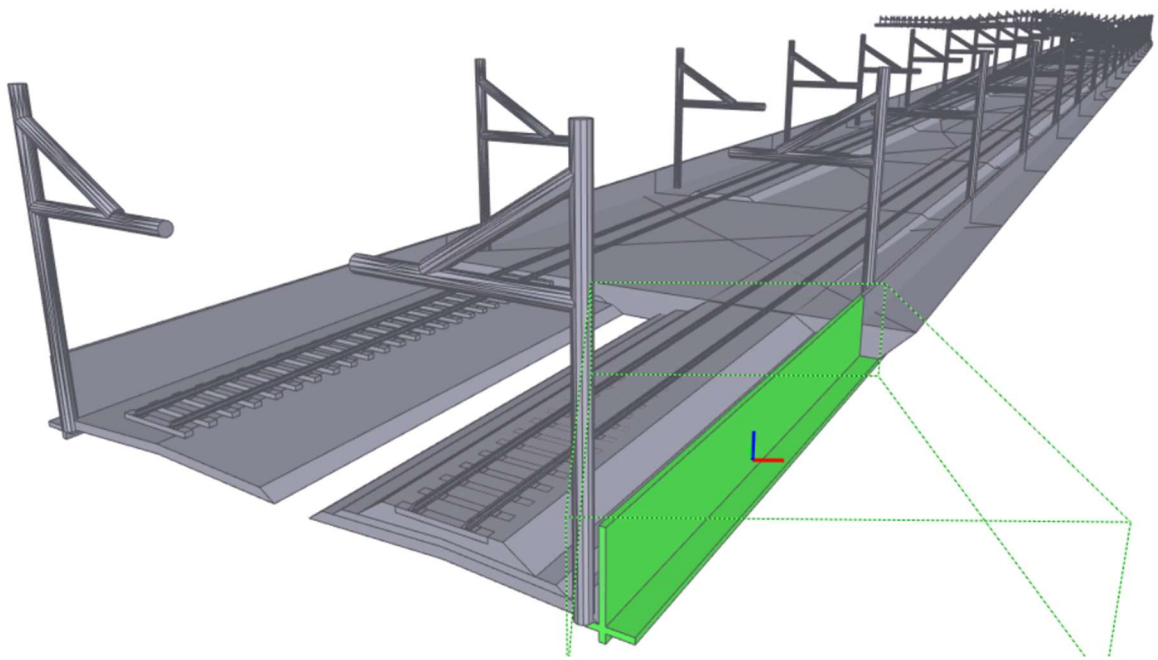


Figure 8-24 BIM model of the retaining wall

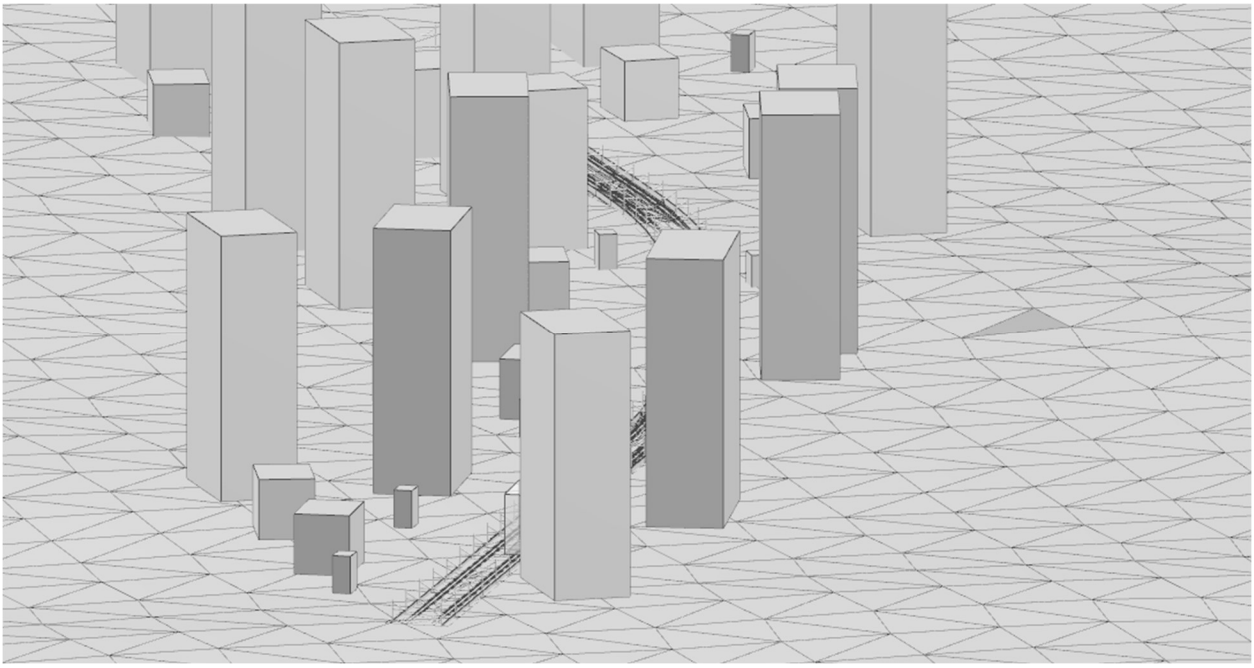


Figure 8-25 BIM model at the city level

8.1.3 Navisworks

To develop 4D BIM models into 5D BIM models, the project schedule or time aspect is included in the BIM models. In this thesis, Navisworks is used to achieve the aim of 5D BIM models. As mentioned, objects in the BIM models are modelled as solids. Each solid is exported to IFC files for further deployment. The IFC files can be imported to Navisworks when each solid is independent. Therefore, they can be scheduled according to the construction or maintenance plans. To facilitate the workflow, Navisworks allows users to synchronize the schedule and incorporate it with other schedule software such as Primavera P6 or Microsoft Project. The progress of the schedule can be monitored and animation of the progress can be visualized. An

example of a BIM model integrated with Navisworks is shown in Figure 8-26. Until this step, the BIM models are 5D.

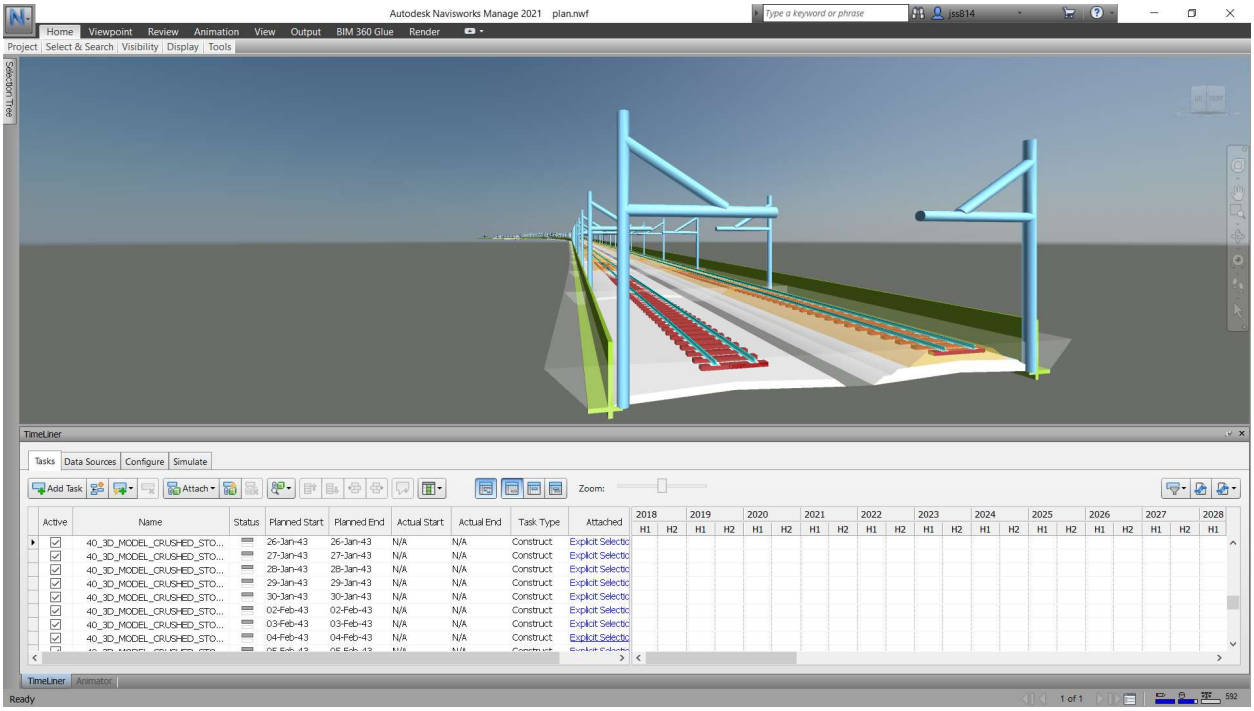


Figure 8-26 A BIM model integrated with Navisworks

8.1.4 Dynamo and VBA for data integration

To develop the 5D BIM models to 6D BIM models, maintenance aspect information is added. From Sections 5 to 7, there are many parameters related to the maintenance aspect. They can be integrated with the BIM models. However, the integration has to be done carefully otherwise the size of the BIM models will be too big and not suitable for deployment. From the test, the preliminary suggestion is independent numbers such as speed and weight of rolling stock, track geometry parameters, or defects can be stored in the BIM models directly. However, time-series

data or sequence data such as ABAs are not suitable for being stored in the BIM models directly because the size of the data themselves is large and they will make the whole BIM models even larger. Therefore, it is suggested to integrate this kind of data in the form of hyperlinks instead as shown in Figure 8-27. From the figures, it can be seen that the independent information can be stored in the BIM models directly such as defects and speed. However, ABAs are integrated in the form of hyperlinks. Hyperlinks are used to indicate the locations of information or data. They can be a local repository, online server, or cloud. The solid objects also link with the hyperlinks as shown in Figure 8-28.

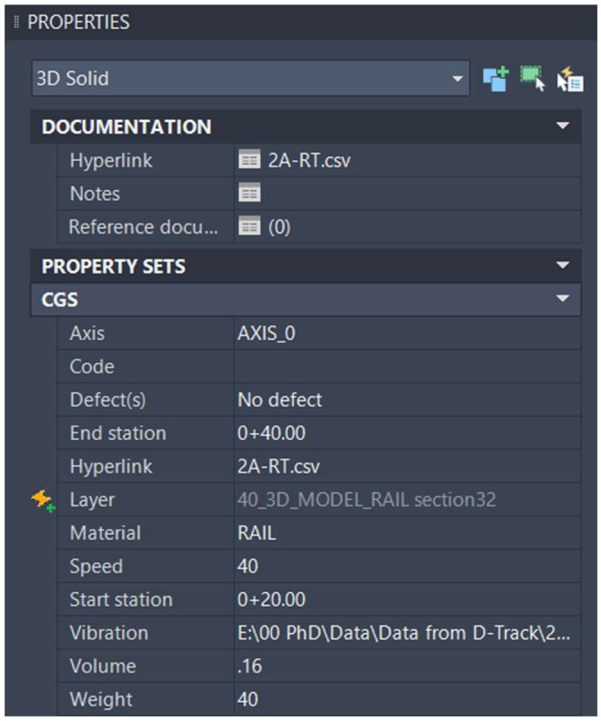


Figure 8-27 Examples of data integration in BIM models

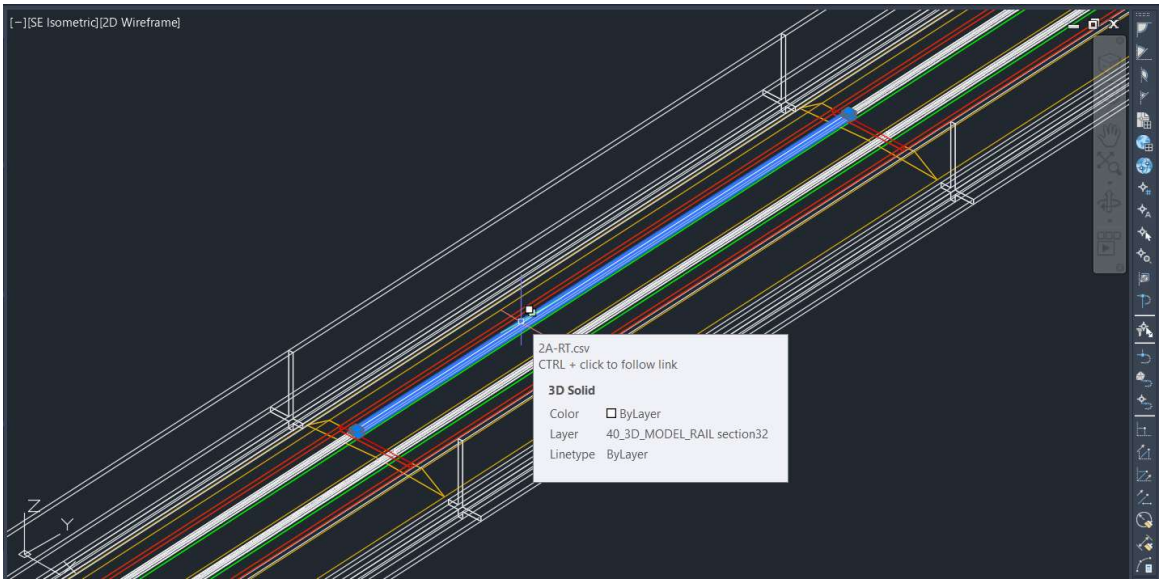


Figure 8-28 Hyperlink stored in BIM models to contain time-series data

When the property set definitions are defined to suit the purpose of BIM models, BIM models are ready to be integrated and able to store information. To perform the integration, Autodesk Dynamo is used. Dynamo is a visual programming software that allows users to create designed workflows for BIM used in different software such as Revit or Civil3D. It is useful for users when they need to perform repetitive tasks or automate workflows including data analysis. For BIM models that have not been designed for information integration in the first place, information tends to be outdated. To update these outdated BIM models and integrate information, additional property set definitions have to be added to the BIM models to allow additional information to be added. Then, the BIM models can represent the real system due to new information from data collection.

The advantage of Dynamo is it provides a visual interface for creating a workflow. Sub-functions are in the form of blocked scripts that are connected with other scripts. Therefore, users can obtain benefits from this function because they do not need to code to create workflows. An example of a Dynamo application is demonstrated in Figure 8-29. The workflow is used to export values in property set definition to CSV files. These exported CSV files can be used for data analysis by railway operators or training machine learning models.

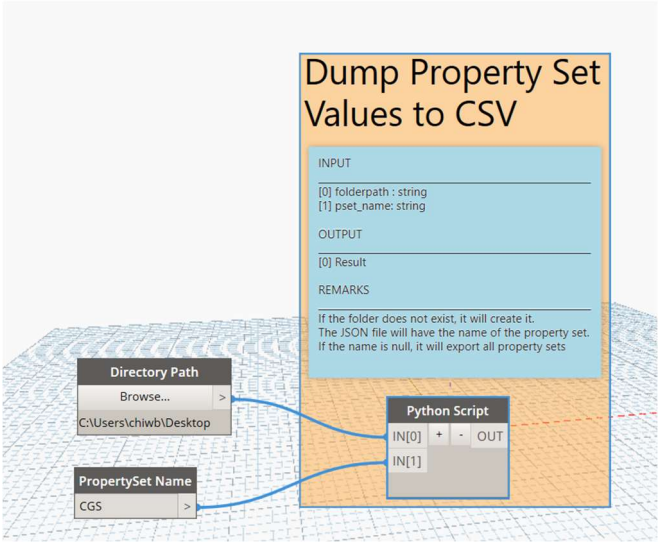


Figure 8-29 Dynamo for exporting property set values

Python script can be included in the Dynamo workflow as shown in Figure 8-30. Therefore, the flexibility of Dynamo is high and can be used to facilitate BIM and machine learning integration. For further data analysis, VBA can be used. An example of a VBA application is shown in Figure 8-31. The code is used to include hyperlinks in BIM models to store time-series data in the BIM models.

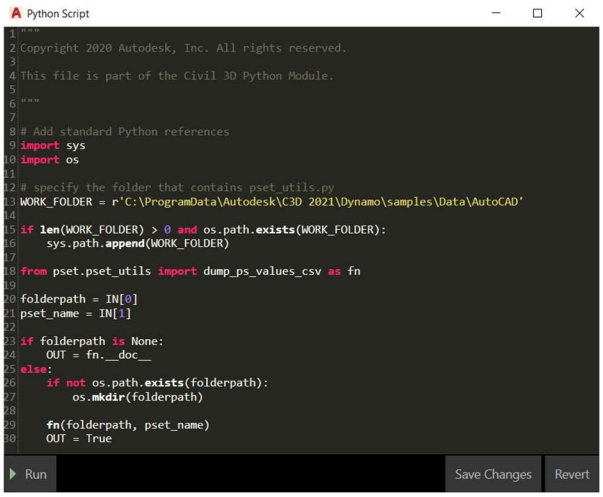


Figure 8-30 Python script for exporting property set values

```
Sub A_CopySpeedAndWeightToBIM()

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.EnableEvents = False

Dim c As Long
Dim x As Long
Dim y As Long
x = 102
y = 1

For c = 1 To 5000
    If Left(Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x, 11), 24) = "40_3D_MODEL_RAIL section" Then
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x, 8) = Workbooks("simplified data.csv").Worksheets("simplified data").Cells(y,
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x, 9) = Workbooks("simplified data.csv").Worksheets("simplified data").Cells(y,
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x + 1, 8) = Workbooks("simplified data.csv").Worksheets("simplified data").Cell
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x + 1, 9) = Workbooks("simplified data.csv").Worksheets("simplified data").Cell
        Filename = "E:\00 PhD\Data\Data from D-Track\20201103 Dipped joint and settlement\D-Track report\" & y & "A-RT.csv"
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x, 10) = Filename
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x + 1, 10) = Filename
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x, 12).Hyperlinks.Add Anchor:=Workbooks("PropertySet_CGS.csv").Worksheets("Proj
        Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x + 1, 12).Hyperlinks.Add Anchor:=Workbooks("PropertySet_CGS.csv").Worksheets("
        x = x + 2
        y = y + 1
    Else
        x = x + 1
    End If
    If Workbooks("PropertySet_CGS.csv").Worksheets("PropertySet_CGS").Cells(x, 1) = "" Then
        Exit For
    End If
Next c

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.EnableEvents = True

End Sub
```

Figure 8-31 Example of VBA application for including hyperlinks in BIM models

When the CSV files containing property set definition values are updated, they can be used to update the values in the BIM models. Figure 8-32 demonstrates the Dynamo workflow for

updating the values in property set definitions and Figure 8-33 demonstrates the Dynamo workflow for including the hyperlinks of time-series data in BIM models.

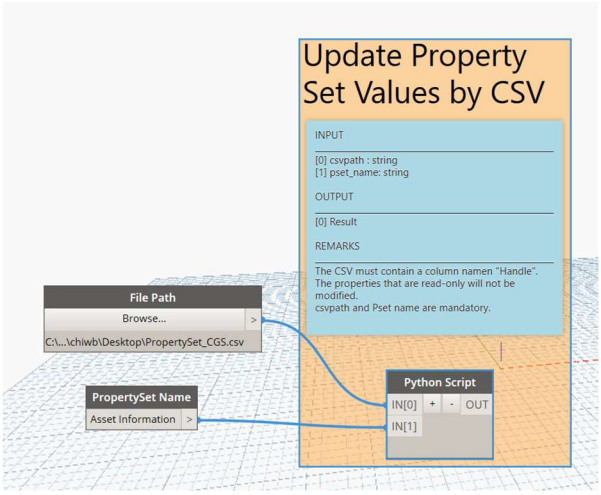


Figure 8-32 Dynamo for updating property set values

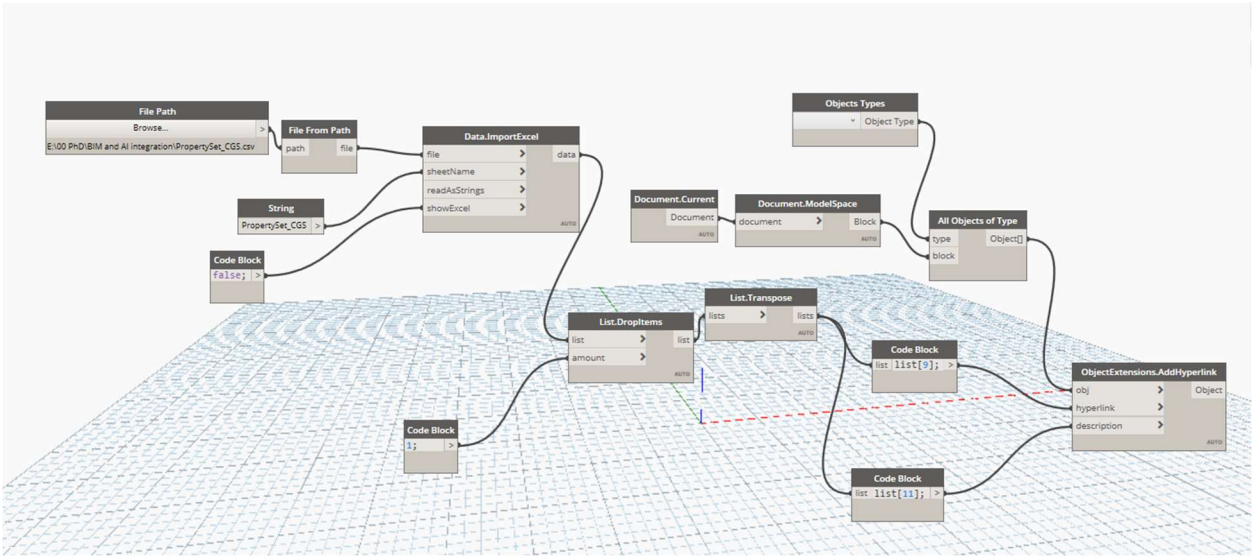


Figure 8-33 Dynamo for including hyperlinks in BIM models

From this, it can be seen that the application of Dynamo is flexible and can be used for various purposes. For example, it can be used to create a notification system when machine learning models predict the occurrence of defects in railway infrastructure. Figure 8-34 presents the Dynamo for creating the notification system. The property set values in terms of defects from machine learning models are used to detect defects. Dynamo is used to identify the railway components that have defects and change the colour of those components. Figure 8-35 presents the BIM model when there is no defect detected and Figure 8-36 presents the BIM model when Dynamo can detect defects in the BIM model. This is an additional system to facilitate railway operators besides the data in the tabular form as CSV files for detecting defects or other data analysis.

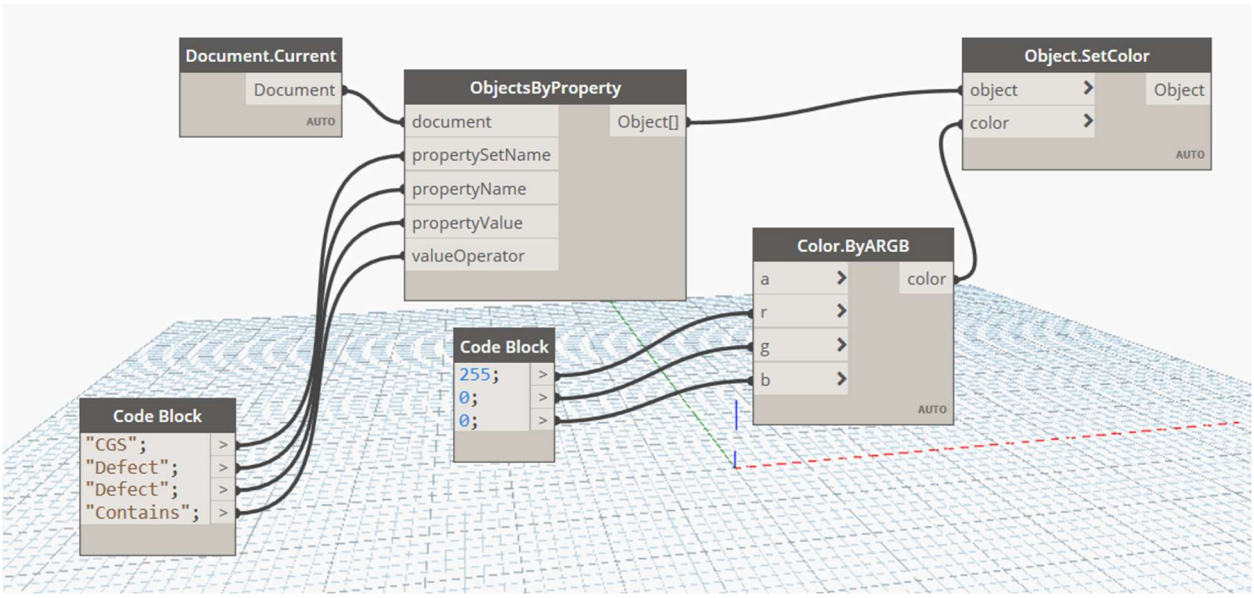


Figure 8-34 Dynamo for creating the notification system when defects are detected

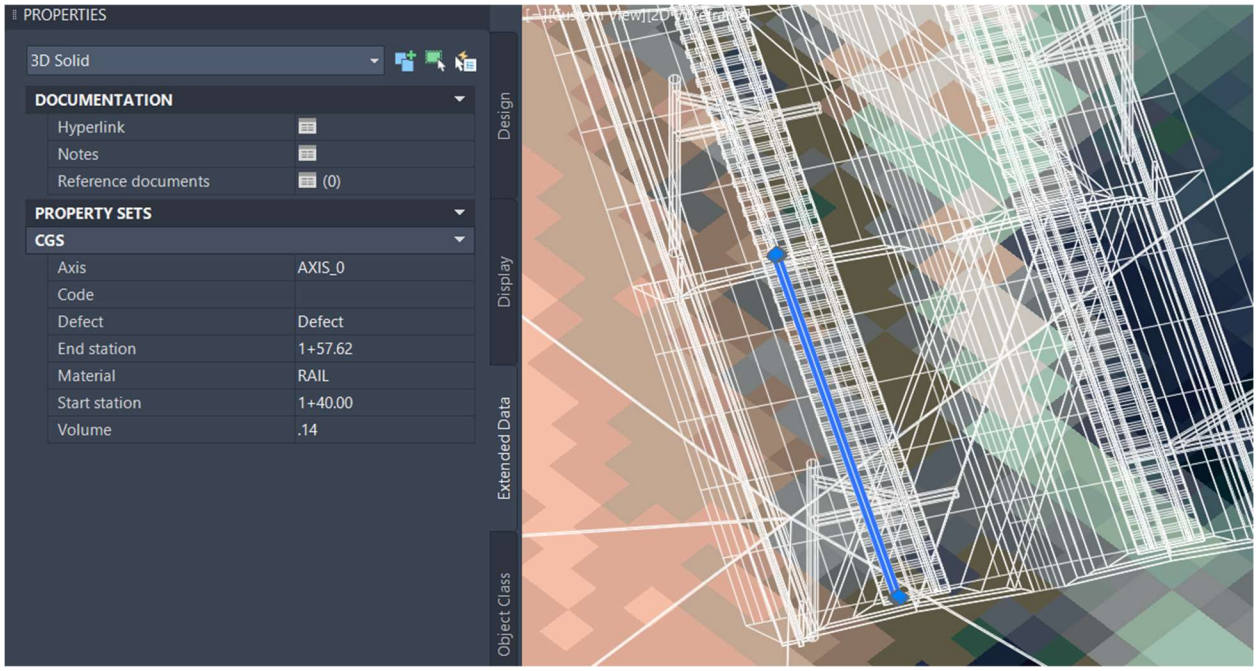


Figure 8-35 BIM model when there are no defects detected

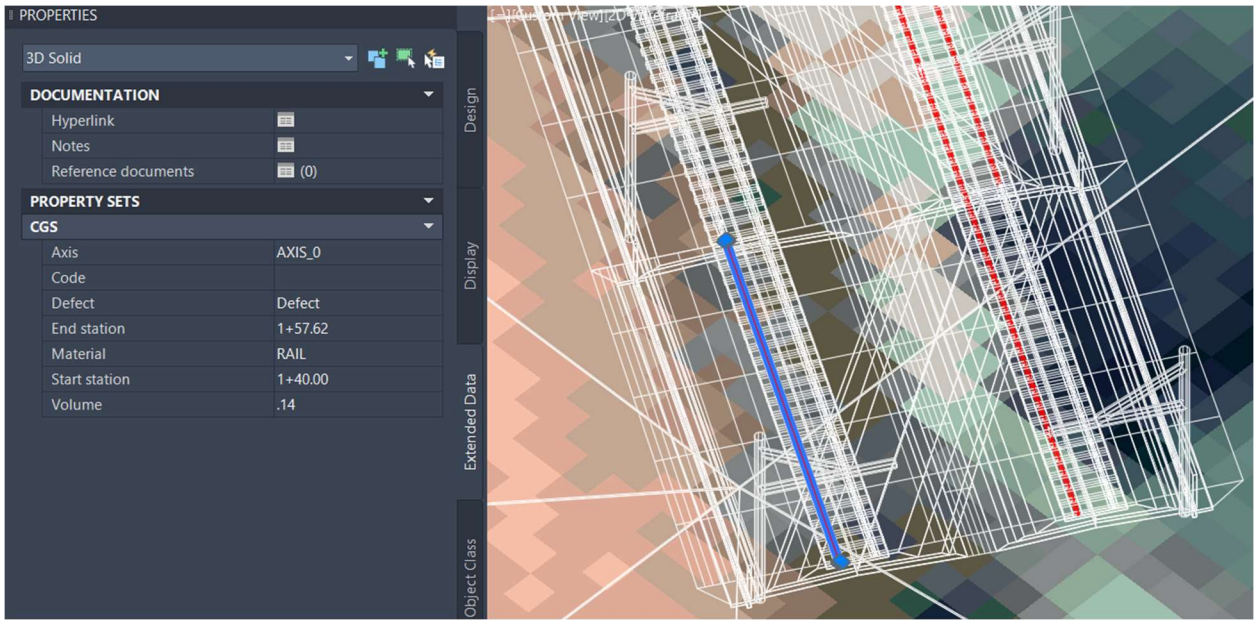


Figure 8-36 BIM model when defect detected

In conclusion, BIM and machine learning can be integrated and exchangeable using the developed approach. The summarized workflow can be shown in Figure 8-37. First, data related to the maintenance aspects are collected through inspection using different methods. Then, data are prepared and processed to make sure they are in forms that are suitable for being stored in the BIM models. Processed data will be saved in the local repository, servers, or cloud depending on the organization. These data are linked with the BIM models with the support of VBA and Dynamo as mentioned. However, the BIM models will have the capability to store information or data when they are designed and developed in the beginning stage. This is conducted using property set definitions available in the BIM models. Then, data stored in the BIM models can be utilized for machine learning and other data analysis by using Dynamo and Python to export data in the form of CSV files. Exported CSV files containing data are processed for training machine learning models for different purposes as mentioned in the previous chapters of this thesis. Besides training, this step is used to include predictions from machine learning in the BIM models to inform railway operators about defects and required maintenance activities. The predictions can also be included in the BIM models. This is the overall process to integrate BIM and machine learning for railway predictive maintenance.

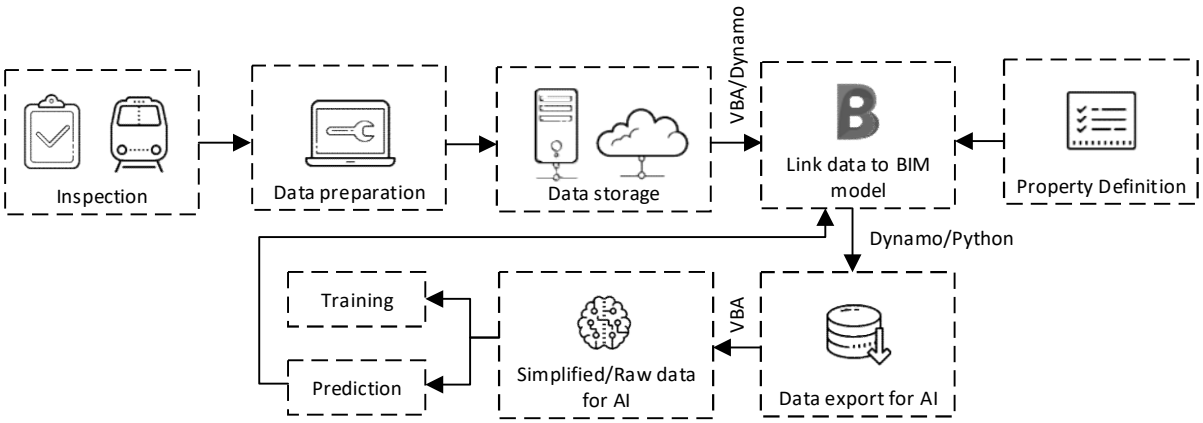


Figure 8-37 Workflow for BIM and machine learning integration

8.2 GUI development for deployment

To make the deployment of the developed approach easier, GUI is also developed to facilitate users who are not familiar with Python code. The process starts with saving the trained models in the form of H5 files. When the machine learning models are trained and ready for deployment, H5 files can be used as pre-trained models otherwise machine learning models have to be trained again every time. H5 files are one of the popular formats that can be used to save the pre-trained models. H5 files contain the architecture, weights, and compile information of the trained models. The pre-trained models can be deployed easily and quickly by importing the models through H5 files. This section will use H5 files to save and load pre-trained machine learning models to shorten the time when railway operators need to use the developed machine learning models.

The next step to develop the GUI is selecting a Python library that will be used. In this thesis, the Customtkinter package is used to develop the GUI. It is one of the popular libraries that is widely used to develop GUIs for different purposes.

To use Python to develop the GUI using Customtkinter, the process starts with importing the Customtkinter module. Then, the main window is created. In this window, all elements of the GUI are contained. Additional windows can be created using the toplevel function.

Different components of the GUI are called widgets in Customtkinter. Examples of available widgets are buttons, labels, checkboxes, entries, frames, and radio buttons. To add different widgets to the main window, they have to be defined first. Different properties of widgets can be adjusted and defined depending on the types of widgets. Examples of widgets' properties are colours, sizes, fonts, and commands. However, widgets will not be added to the main window until they are positioned.

To position the widgets, many methods can be used such as pack, grid, and place. For the pack, the widgets will be added automatically in order. Therefore, it might be difficult to position the widgets according to the GUI design. The place function is used to specify the exact locations of the widgets in the main window. Therefore, it is easier to organize the locations compared to the pack function. However, it starts being complicated when there are many widgets and they are all needed to be positioned manually. This section will use the grid function because it can compromise the ability to organize and the simplicity of the organization together. The grid function determines the positions of the widgets in the form of rows and columns.

When all widgets are defined and positioned on the main window, the design of the GUI is complete. However, to make the developed GUI works, commands of different widgets have to be defined. For example, some buttons are used to run different models while some are used to select inputs or display the results.

The result from the GUI development is shown in Figure 8-38. The first part is used to select the available data. The available data will define which models or functions can be used given the available data. The disable and enable functions are used as commands to allow and not allow to use of different functions in the GUI.

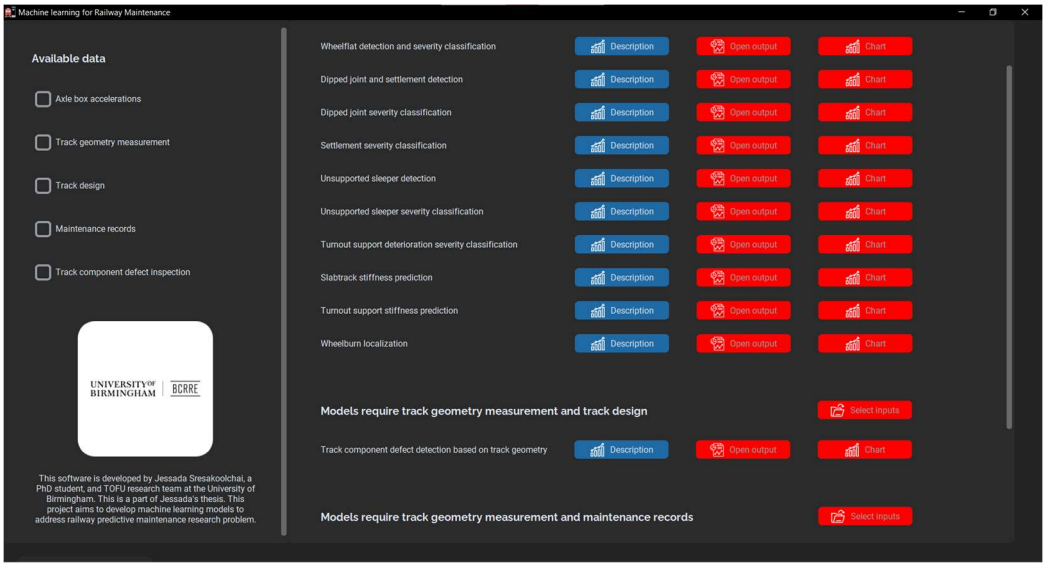


Figure 8-38 The GUI developed by using Customtkinter

When the available data are selected, the support functions will be available as shown in Figure 8-39. The buttons' design is performed to clearly demonstrate the available function by changing colours.

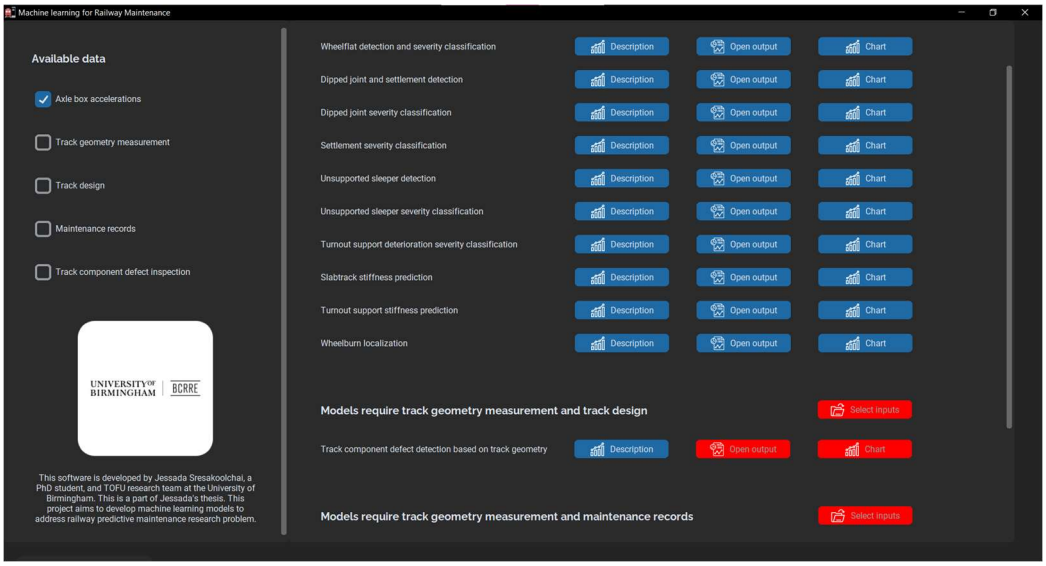


Figure 8-39 Function enabling when available data are selected

Descriptions of the machine learning models' functions are added to describe the function of each machine learning model. An example is shown in Figure 8-40. The description windows are created using the toplevel widget. It is used to create additional windows from the main window. Figures and descriptions as the texts can be added to the toplevel window as labels.

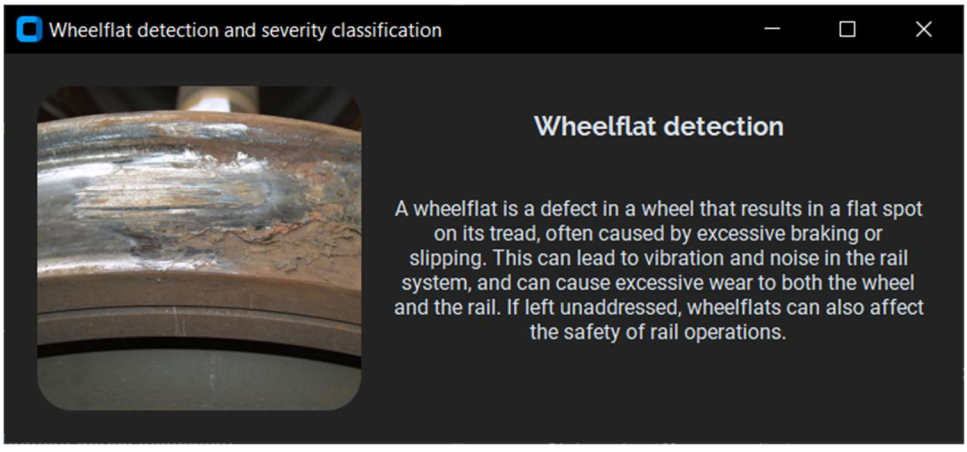


Figure 8-40 An example of a description of machine learning models' functions

When available data are selected, inputs can be selected to run the pre-trained machine learning models. For the machine learning models that use the same inputs, all machine learning models will be run at the same time to save time. Then, individual reports and charts can be selected to be shown in Figure 8-41 and Figure 8-42 respectively. To run the pre-trained models, the GUI will load the saved pre-trained models using H5 files as mentioned. The reports are saved as XLS files or Excel spreadsheets.

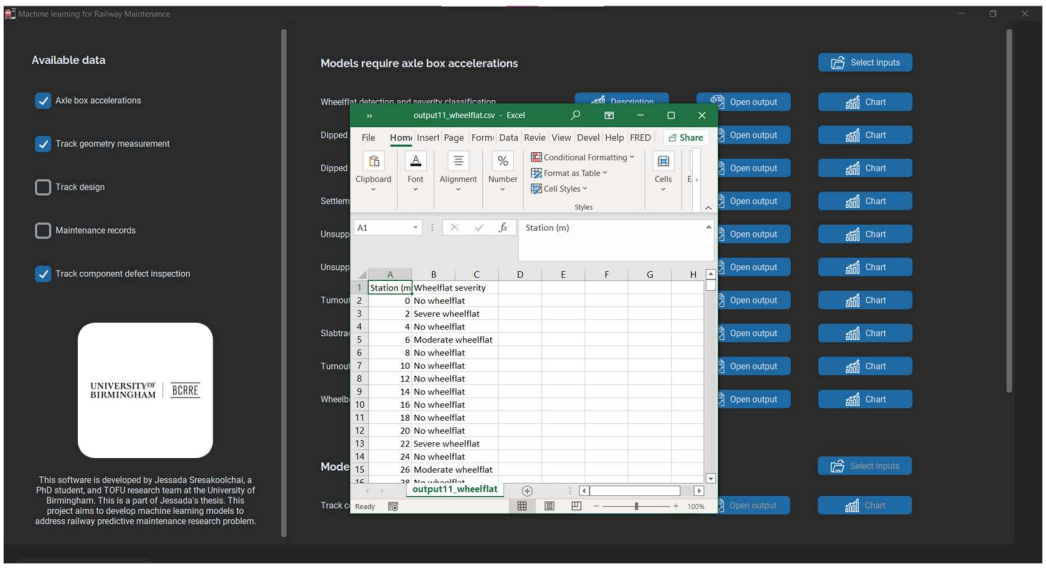


Figure 8-41 An example of reports from pre-trained machine learning models

For the charts, Matplotlib is used to display the charts. The types of charts can be defined such as scatter plots or line charts. To make the charts can better display the results from the machine learning model's predictions and allow users to display the interesting range or track locations, the function used to adjust the range of the x-axis is added as shown in Figure 8-42 and Figure 8-43. The minimum and maximum range of the x-axis can be adjusted by inputting the target

values. This can be done by using entry widgets from Customtkinter. Entries are used to receive the values for further processing. However, Customtkinter and Matplotlib do not have the function to adjust and replot the charts. Therefore, an additional step is added to the plot button. The plot button is used the start the process of replotting. It will get the values from the entries of minimum and maximum ranges of the x-axis. Then, the charts will be deleted and the new ranges are used to display the new charts with new ranges. The charts can be saved as PGN files by clicking the save figure button which has the function to save the chart by selecting the locations of the saved figures. Last, the machine learning models are linked with the BIM models. The button to open the BIM model is also included at the top of the main window.

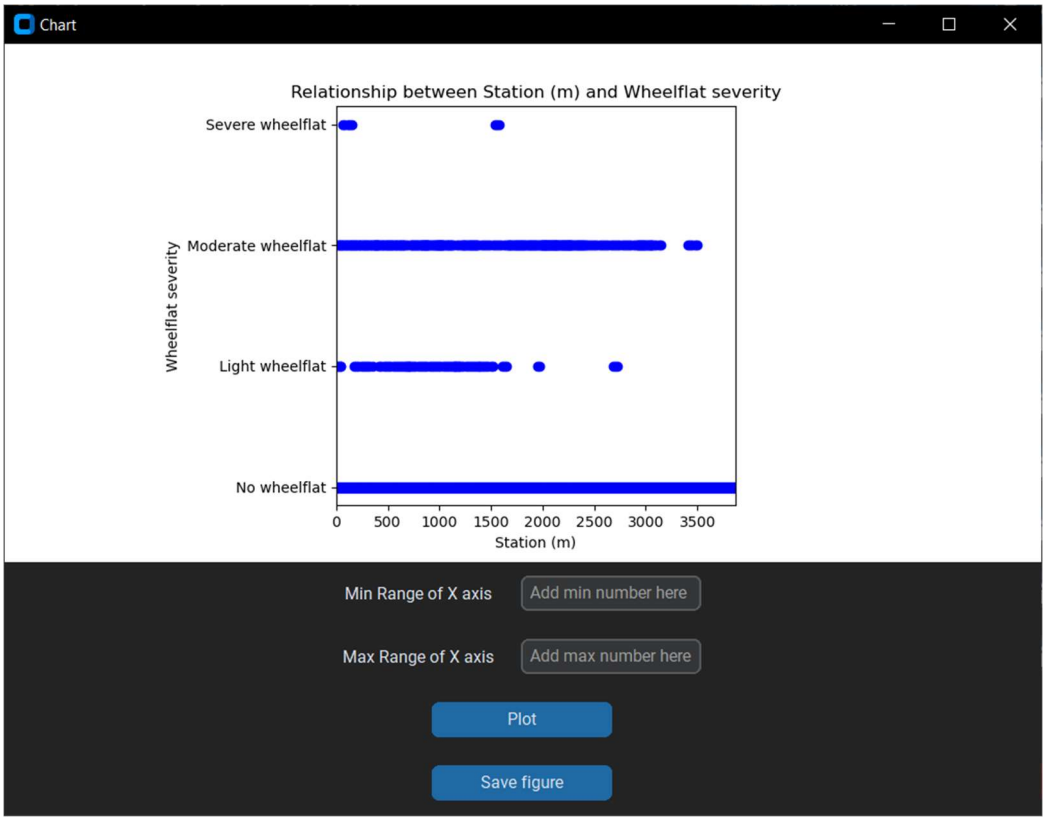


Figure 8-42 An example of charts from pre-trained machine learning models



Figure 8-43 Result from adjusting the range of the x-axis

The developed GUI for deploying integrated BIM and machine learning can provide a user-friendly experience for users. It can improve accessibility for users who are not familiar with coding. The developed GIU can also improve productivity even for users who are familiar with coding because they do not need to edit the code manually which can help reduce human error and save time for debugging. Other contributions of the developed GUI are it can improve visualization. The outputs from the machine learning models can be displayed as charts and reports which are easy to understand by users with different backgrounds in the railway system and machine learning. The GUI can be used as the framework that can promote collaboration.

Additional functions can be added to the developed GUI. Therefore, railway operators or software developers can further develop machine learning models and added to the GUI to increase the capability of the software. It can be concluded that the developed GUI can be used by users who have different backgrounds and skill levels which is characteristic of the railway system that required multi-discipline knowledge to operate the railway system.

9 Machine learning and BIM model deployment

This section presents how to deploy the developed approach for the integration of BIM and machine learning. In the previous section, BIM models are developed to support the integration with machine learning models. At the same time, machine learning models for different purposes are trained and ready for deployment. There are different options for the deployment which will be described in this section. They all have different advantages and disadvantages. Railway operators have to consider each option and choose the options that are most suitable for their application.

9.1 Python code

The first option is to deploy the developed approach using the Python code directly. Therefore, there is no additional step for converting or processing the developed software. The developed software can be used offline or online. For example, VS code can be used to run the code directly as the development platform. Another option is uploading the code to online platforms such as Google Colab which might have more powerful computational hardware and the developed software can be shared with other users.

The advantages of this option are:

- It is easy to use for both developers and users who are familiar with Python codes. Developers do not need to convert the codes and be able to use the codes as they are developed. Users can also see the detail of the codes and understand how the codes work.

At the same time, users can edit the codes to suit their needs and add more functions or modify the codes to obtain the functions they need.

- The implementation is fast because Python is fast to comply with compared to other languages.
- This option provides more flexibility than other options because the codes can be modified to support new data or edit models' architecture. Therefore, any users can be developers themselves when they need to deal with new problems.
- Python codes can be deployed on any platform such as Windows, Linux, or macOS. Therefore, any user can technically deploy the developed software.

The disadvantages of this option are:

- Additional installation is required for setting up the environment. Python codes require specific libraries to run the codes. In some cases, specific versions of different libraries are limited to make the codes runnable. Therefore, it is sometimes challenging for users to perform complex installations.
- Python codes can be security risks because they can be easily accessed and modified. Therefore, it can be risky if the shared Python codes are not protected and monitored properly.
- Limitation about the scalability of Python codes is one of the concerns. This is because normally Python codes have limitations in execution speed and memory management. Therefore, the deployment may be slow when the computational hardware is not powerful enough.

- Python is one of the most dynamic languages. Related packages and libraries including the language are changeable. Therefore, the codes have to be updated regularly. The debugging is challenging due to the mentioned reasons so it is difficult to identify and correct errors, especially for users who are not familiar with the codes.

9.2 EXE

The second option to deploy the developed approach is converting the code to the EXE file. This can be done by using the supported Python libraries such as PyInstaller or Auto Py to Exe. The libraries will convert the Python codes to EXE files that are easier to execute. This thesis uses Auto Py to Exe library to generate the EXE file because the library provides the GUI to make it easy to convert the code and understandable different settings. However, the GUI in this thesis is developed using Customtkinter which there are some limitations and requires additional steps and requirements. First, the onefile option is not useable because the Customtkinter library includes Json and Otf files. Therefore, the converting needs to use onedir option instead. Second, the Customtkinter directory has to be added along with the converting process otherwise the concerted EXE file will not be able to execute. This is because Auto Py to Exe library does not automatically merge related Json files properly.

The advantages of this option are:

- There are no required additional or complex installations. Therefore, it is easy and convenient to distribute the developed software to any user. The EXE file is also easy to use and does not require coding skills.

- The EXE includes all necessary libraries and packages so the installation is easy. In addition, it can be ensured that users can definitely deploy the developed software.
- The security is better than the Python codes because the EXE file is digitally signed and encrypted. Therefore, no one can edit the code when it is converted to the EXE file.
- The EXE file can be run on different platforms such as Windows, Linux, and macOS. Therefore, it guarantees that every user can deploy the developed software.

The disadvantages of this option are:

- Flexibility is limited because the EXE file cannot be modified easily. It cannot be used to train machine learning models using new data or new models' architecture.
- Limitation about the scalability of Python codes is one of the concerns. This is because normally Python codes have limitations in execution speed and memory management. Therefore, the deployment may be slow when the computational hardware is not powerful enough.
- Debugging options are limited due to less flexibility. In cases of errors, it is difficult to debug and modify the codes. Therefore, users will not be able to analyze the errors efficiently.
- The conversion can be complex because the creation of EXE files is more complex than deploying the developed software using Python code directly. Especially, the developed GUI in this thesis requires additional steps to convert the PY file to the EXE file.

- The size of the EXE files and their related files can be large based on the necessary libraries including in the Python codes. The complexity and number of the machine learning models also affect the sizes of the EXE files.

9.3 Replit

The third option is Replit. It is a cloud-based integrated development environment (IDE) where the Python codes can be modified and tested. It also supports other languages besides Python such as JavaScript, Ruby, and more. Therefore, it can support that collaboration between different parties using different platforms. Moreover, this can be done in real-time. Replit provides a hosting service so developers can deploy the codes and apply the codes directly and it can be used through browsers. Therefore, it is easy to deploy the developed approach. It can also be integrated with GitHub which allows developers to manage the code repositories easily. One of the main advantages is developers do not need to install any packages and libraries which can be complex and time-consuming. By this, users can deploy the developed approach easily because they do not need to have a coding background. At the same time, developers can still access the codes. However, it requires access through the internet so there might be some limitations when the internet is not available. It can be seen that Replit is one of the most powerful and versatile platforms for deploying the developed approach.

The advantages of this option are:

- Replit is a cloud-based development environment that can be accessed easily as long as the internet is available.

- Collaboration is promoted using Replit because many developers and users can access and work on the codes simultaneously.
- Replit supports different languages including Python that is used in this thesis. Pre-installation can be done by experienced developers in the first place. Therefore, it can reduce the complexity of the deployment when users need to deploy the software or developers want to modify the codes. It can save time and effort in managing the software.
- It is easy to deploy even for users who do not have any background in coding. Therefore, it is suitable for staff in the railway industry that have different backgrounds and skill levels.

The disadvantages of this option are:

- The computing resources are limited compared to local machines or other online platforms such as Google Colab. The deployment might be done with difficulty when the machine learning models or software are complex.
- Hardware configurations can be controlled with limitations because Replit is an online platform. Developers do not have the option to upgrade the hardware and infrastructure.
- The deployment can be accessed using web browsers only. Therefore, the configuration is limited.
- Large datasets may not be suitable for this platform because the storage is limited. In addition, uploading large files or datasets can be time-consuming because this platform is could-based.

10 Conclusions and recommendations

10.1 Research findings

This thesis aims to integrate BIM and machine learning for railway infrastructure predictive maintenance. The results from the study demonstrate the potential of the integration of BIM and machine learning in many aspects and the developed approach can revolutionize the procedure in the railway industry and develop the railway organization to be the data-driven organization that the decision-making is done based on the data and is not affected by human biases and errors. BIM can be used to create a digital model for railway projects that can be used for the whole service life of railway projects. In this thesis, BIM models are developed to meet the requirement of 6D BIM models that contain the information of physical information, time, cost, and maintenance aspect. At the same time, machine learning can utilize the information stored in BIM models for further analysis and railway predictive maintenance. The integration of BIM and machine learning can improve maintenance efficiency, accurate decision-making, reduce maintenance costs, reduce possession time, and reduce carbon emissions.

For the BIM technology, BIM can be used to create digital models for railway projects so that the models' detail can be adjusted according to the needs of the application. For example, BIM models can be integrated with building BIM models to demonstrate the overall landscape of the city level when buildings and railway projects are integrated. On the other hand, BIM models of railway projects can be modelled in detail to include the detail of railway infrastructure components such as rails, sleepers, rail pads, ballast, OCS poles, retaining walls, pipes, or bolts.

Each component of BIM models can contain information or data related to each component and this information and data can be used further for asset management to improve management efficiency. This information and data can also be used to train machine learning models that are developed for different purposes such as defect detection, defect severity classification, defect localization, or maintenance schedule preparation as demonstrated in this thesis.

Machine learning is a part of artificial intelligence that can be used for different purposes. In this thesis, machine learning is used to develop predictive models for different purposes as demonstrated. One benefit of machine learning is it can detect hidden patterns in data that humans cannot recognize. Unsupervised learning and reinforcement learning are examples that clearly confirm this statement. Because BIM models of railway projects can store a lot of data, machine learning is a tool that can create advantages from those data.

The integration of BIM and ML for railway infrastructure predictive maintenance offers several benefits. One of the key advantages is increased efficiency and accuracy in detecting and predicting maintenance needs. By using digital models of infrastructure assets, maintenance needs can be identified more quickly and accurately, reducing the time and resources required for inspection and repair. This can help reduce downtime, increase asset utilization, and improve overall productivity.

Machine learning can improve the efficiency of railway predictive maintenance in different aspects as demonstrated in this thesis. It can be used to identify defects and schedule maintenance. At the same time, railway operators can prioritize critical maintenance because

machine learning models are also used to evaluate the severity of defects. Good predictive maintenance can reduce maintenance costs because defects be managed better and light defects required lower costs compared to severe defects. Railway operations are also less interrupted because maintenance is planned and known in advance. The risk of failure and accidents in the railway system is also reduced because the railway infrastructure is regularly monitored.

Overall, the reliability, availability, maintainability, safety, and sustainability of the railway system will be improved using the integrated approach of BIM and machine learning developed in this thesis. The railway infrastructure can be maintained more efficiently, accurately, and cost-effectively.

10.2 Recommendations for future research

For the recommendations for further research, they can be concluded as follows:

1. More sensors and data collection can be considered to explore their suitability of them for railway predictive maintenance.
2. Advanced machine learning techniques can be tried to see whether they can improve the performance of the prediction.
3. Standards and specifications for integrating BIM and machine learning can be further studied and defined as the guideline or standards for the railway industry.
4. The level of detail for BIM models should be considered to define the optimum detail of BIM models which is sufficient and, at the same time, not excessive.

5. Related procedures for integrating BIM and machine learning can be developed such as data collection, data processing, and model validation to standardize the procedure of integration.
6. Extend the application of BIM and machine learning for other applications such as whole life cycle management, asset management, risk management, and safety management.
7. Collaborate with railway infrastructure stakeholders to evaluate the developed approach and improve it to meet requirements.
8. For the environmental aspects, more detail related to extreme weather events such as landslides and heat waves can be included.
9. Applications of augmented reality (AR) and virtual reality (VR) can be included for better visualization and interpretation of BIM and machine learning integration.
10. Other information technologies can be included such as IoT, blockchain, and cloud computing.
11. Explore the requirement of reskilling and upskilling for the implementation of the integrated BIM and machine learning approach for railway infrastructure predictive maintenance.
12. Additionally, assess the environmental aspect such as considering other greenhouse gas besides carbon dioxide.
13. Increase the complexity of the machine learning models by including limitations and constraints such as the limited resources and time for railway maintenance.
14. Evaluate the economic, social, and environmental benefits that the railway industry obtains from the use of the developed approach.

15. Explore the feasibility to apply the developed approach to other systems such as roads, bridges, or airports.
16. Explore the environmental effects of the use of the developed approach because some studies indicate that the use of machine learning requires more energy which also creates carbon emissions.

11 References

1. Eurostat, *Railway passenger transport statistics - quarterly and annual data*. 2019.

2. Budai-Balke, G.G., R.R. Dekker, and U.U. Kaymak, *Genetic and memetic algorithms for scheduling railway maintenance activities*. 2009.

3. Gerum, P.C.L., A. Altay, and M. Baykal-Gürsoy, *Data-driven predictive maintenance scheduling policies for railways.* Transportation Research Part C: Emerging Technologies, 2019. **107**: p. 137-154.

4. Du, G. and R. Karoumi, *Life cycle assessment of a railway bridge: comparison of two superstructure designs.* Struct. Infrastruct. Eng., 2013. **9**(11): p. 1149-1160.

5. Bryde, D., M. Broquetas, and J.M. Volm, *The project benefits of Building Information Modelling (BIM).* International Journal of Project Management, 2013. **31**(7): p. 971-980.

6. Hoeber, H. and D. Alsem, *Life-cycle information management using open-standard BIM.* Engineering, Construction and Architectural Management, 2016. **23**(6): p. 696-708.

7. One Model Inc. *AI Academy: What's the difference between forecasting and predictive modeling?* 2019 [cited 2019 23 December]; Available from: https://www.onemodel.co/blog/ai-academy-forecasting-vs-predictive-modeling.

8. Thompson, W., H. Li, and A. Bolen. *Artificial intelligence, machine learning, deep learning and beyond*. 2019 [cited 2019 23 December]; Available from: https://www.sas.com/en_th/insights/articles/big-data/artificial-intelligence-machine-learning-deep-learning-and-beyond.html#/.

9. Dhar, V., *Data science and prediction.* 2012.

10. Zhang, X., N. Feng, Y. Wang, and Y. Shen, *Acoustic emission detection of rail defect based on wavelet transform and Shannon entropy.* J. Sound Vib., 2015. **339**: p. 419-432.

11. Xiong, Z., Q. Li, Q. Mao, and Q. Zou, *A 3D laser profiling system for rail surface defect detection.* Sensors, 2017. **17**(8): p. 1791.

12. Liu, Z., W. Li, F. Xue, J. Xiafang, B. Bu, and Z. Yi, *Electromagnetic Tomography Rail Defect Inspection.* IEEE Transactions on Magnetics, 2015. **51**(10): p. 1-7.

13. Cheng, Y., D. HongGui, and F. YuXin. *Effects of Faster Region-based Convolutional Neural Network on the Detection Efficiency of Rail Defects under Machine Vision*. in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. 2020.

14. Tastimur, C., H. Yetis, M. Karaköse, and E. Akin, *Rail defect detection and classification with real time image processing technique.* IJCSSE, 2016. **5**(12): p. 283.

15. Guler, H., *Prediction of railway track geometry deterioration using artificial neural networks: a case study for Turkish state railways.* Struct. Infrastruct. Eng., 2014. **10**(5): p. 614-626.

16. Letot, C., I. Soleimanmeigouni, A. Ahmadi, and P. Dehombreux, *An adaptive opportunistic maintenance model based on railway track condition prediction.* IFAC-PapersOnLine, 2016. **49**(28): p. 120-125.

17. Lee, J.S., S.H. Hwang, I.Y. Choi, and I.K. Kim, *Prediction of track deterioration using maintenance data and machine learning schemes.* J. Transp. Eng. A: Syst., 2018. **144**(9): p. 04018045.

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

19. Gibała, Ł. and J. Konieczny, *Application of artificial neural networks to predict railway switch durability.* Zeszyty Naukowe. Transport/Politechnika Śląska, 2018.

20. Bukhsh, Z.A., A. Saeed, I. Stipanovic, and A.G. Doree, *Predictive maintenance using tree-based classification techniques: A case of railway switches.* Transp. Res. Part C Emerg. Technol., 2019. **101**: p. 35-54.

21. Gao, R., C. Shang, and H. Jiang, *A fault detection strategy for wheel flat scars with wavelet neural networks and genetic algorithm.* J. Xi'an Jiaotong Univ., 2013. **47**(9): p. 88-91.

22. Krummenacher, G., C.S. Ong, S. Koller, S. Kobayashi, and J.M. Buhmann, *Wheel defect detection with machine learning.* IEEE trans Intell Transp Syst, 2017. **19**(4): p. 1176-1187.

23. Mohammadi, R., Q. He, F. Ghofrani, A. Pathak, and A. Aref, *Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects.* Transp. Res. Part C Emerg. Technol., 2019. **102**: p. 153-172.

24. Sysyn, M., U. Gerber, O. Nabochenko, D. Gruen, and F.J.U.r.t. Kluge, *Prediction of rail contact fatigue on crossings using image processing and machine learning methods.* 2019. **5**(2): p. 123-132.

25. Franca, A.S. and R.F. Vassallo, *A Method of Classifying Railway Sleepers and Surface Defects in Real Environment.* IEEE Sens. J., 2021. **21**(10): p. 11301-11309.

26. Yella, S., A.S. Rahman, and M. Dougherty. *Pattern recognition for classifying the condition of wooden railway sleepers*. in *2010 International Conference on Multimedia Computing and Information Technology (MCIT)*. 2010.

27. Zheng, Y., S. Wu, D. Liu, R. Wei, S. Li, and Z. Tu. *Sleeper Defect Detection Based on Improved YOLO V3 Algorithm*. in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. 2020.

28. Wang, G., Y. Liu, and J. Xiang. *A two-stage algorithm of railway sleeper crack detection based on edge detection and CNN*. in *2020 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM)*. 2020.

29. Sysyn, M., M. Przybylowicz, O. Nabochenko, and L. Kou, *Identification of Sleeper Support Conditions Using Mechanical Model Supported Data-Driven Approach.* Sensors, 2021. **21**(11).

30. Lee, J.S., S.H. Hwang, I.Y. Choi, and Y. Choi, *Deterioration prediction of track geometry using periodic measurement data and incremental support vector regression model.* J. Transp. Eng. A: Syst., 2020. **146**(1): p. 04019057.

31. Hu, C. and X. Liu. *Modeling Track Geometry Degradation Using Support Vector Machine Technique*. in *2016 Joint Rail Conference*. 2016.

32. Sysyn, M., D. Gruen, U. Gerber, O. Nabochenko, and V. Kovalchuk, *Turnout monitoring with vehicle based inertial measurements of operational trains: a machine learning approach.* Communications – Scientific Letters of the University of Zilina, 2019. **21**(1): p. 42-48.

33. Lopes Gerum, P.C., A. Altay, and M. Baykal-Gürsoy, *Data-driven predictive maintenance scheduling policies for railways.* Transportation Research Part C: Emerging Technologies, 2019. **107**: p. 137-154.

34. Consilvio, A., J. Solís-Hernández, N. Jiménez-Redondo, P. Sanetti, F. Papa, and I. Mingolarra-Garaizar, *On Applying Machine Learning and Simulative Approaches to Railway Asset Management: The Earthworks and Track Circuits Case Studies.* Sustainability, 2020. **12**(6).

35. Mohammadi, R. and Q. He, *A deep reinforcement learning approach for rail renewal and maintenance planning.* Reliability Engineering & System Safety, 2022. **225**: p. 108615.

36. Azhar, S., *Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry.* Leader Manag Eng, 2011. **11**(3): p. 241-252.

37. Infrastructure and Projects Authority. *Creating a Digital Built Britain: what you need to know*. 2017; Available from: https://www.gov.uk/guidance/creating-a-digital-built-britain-what-you-need-to-know#what-digital-built-britain-aims-to-do.

38. Kaewunruen, S., J. Sresakoolchai, and Z. Zhou, *Sustainability-Based Lifecycle Management for Bridge Infrastructure Using 6D BIM.* Sustainability, 2020. **12**: p. 2436.

39. Kurwi, S., P. Demian, and T.M. Hassan. *Integrating BIM and GIS in railway projects: A critical review*. 2017. Association of Researchers in Construction Management (AR-COM).

40. Bensalah, M., A. Elouadi, and H. Mharzi, *Integrating BIM in railway projects Review & perspectives for Morocco and MENA.* Int. J. Recent Sci. Res., 2018. **9**(1): p. 23398-23403.

41. Bensalah, M., A. Elouadi, and H. Mharzi, *BIM integration to railway projects-case study.* Mngmt, 2018. **2**: p. 3.

42. Shin, M.H., H.K. Lee, and H.Y. Kim, *Benefit–Cost Analysis of Building Information Modeling (BIM) in a Railway Site.* Sustainability, 2018. **10**(11): p. 4303.

43. Kaewunruen, S. and N. Xu, *Digital twin for sustainability evaluation of railway station buildings.* Front. Built Environ., 2018. **4**: p. 77.

44. Kaewunruen, S. and Q. Lian, *Digital twin aided sustainability-based lifecycle management for railway turnout systems.* J. Clean. Prod., 2019. **228**: p. 1537-1551.

45. Kaewunruen, S., S. Peng, and O. Phil-Ebosie, *Digital twin aided sustainability and vulnerability audit for subway stations.* Sustainability, 2020. **12**(19): p. 7873.

46. Bensalah, M., A. Elouadi, and H. Mharzi, *Overview: the opportunity of BIM in railway.* Smart Sustain. Built Environ., 2019.

47. Aziz, Z., Z. Riaz, and M. Arslan, *Leveraging BIM and big data to deliver well maintained highways.* Facilities, 2017. **35**(13/14): p. 818-832.

48. Davtalab, O., *Benefits of real-time data driven BIM for FM departments in operations control and maintenance*, in *Computing in Civil Engineering 2017*. 2017. p. 202-210.

49. GhaffarianHoseini, A., T. Zhang, O. Nwadigo, A. GhaffarianHoseini, N. Naismith, J. Tookey, and K. Raahemifar, *Application of nD BIM Integrated Knowledge-based Building Management System (BIM-IKBMS) for inspecting post-construction energy efficiency.* Renew. Sust. Energ. Rev., 2017. **72**: p. 935-949.

50. Peng, Y., J.-R. Lin, J.-P. Zhang, and Z.-Z. Hu, *A hybrid data mining approach on BIM-based building operation and maintenance.* Build Environ, 2017. **126**: p. 483-495.

51. Liu, Z., N. Meyendorf, and N. Mrad. *The role of data fusion in predictive maintenance using Digital Twin*. in *AIP Conference Proceedings*. 2018. AIP Publishing.

52. McArthur, J., N. Shahbazi, R. Fok, C. Raghubar, B. Bortoluzzi, and A. An, *Machine learning and BIM visualization for maintenance issue classification and enhanced data collection.* Adv. Eng. Inform., 2018. **38**: p. 101-112.

53. Chen, W., J.C. Cheng, and Y. Tan, *BIM-and IoT-Based Data-Driven Decision Support System for Predictive Maintenance of Building Facilities.* Innovative Production And Construction: Transforming Construction Through Emerging Technologies, 2019: p. 429.

54. Martinelli, D.R. and H. Teng, *Optimization of railway operations using neural networks.* Transp. Res. Part C Emerg. Technol., 1996. **4**(1): p. 33-49.

55. Zhuo, W., J. Li-Min, Q. Yong, and W. Yan-Hui, *Railway passenger traffic volume prediction based on neural network.* Appl Artif Intell, 2007. **21**(1): p. 1-10.

56. Zheng, D. and Y. Wang. *Application of an artificial neural network on railway passenger flow prediction*. in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*. 2011. IEEE.

57. Dündar, S. and İ. Şahin, *Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways.* Transp. Res. Part C Emerg. Technol., 2013. **27**: p. 1-15.

58. Bergmeir, C., G. Sáinz, C.M. Bertrand, and J.M. Benítez. *A study on the use of machine learning methods for incidence prediction in high-speed train tracks*. in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. 2013. Springer.

59. Feng, J., X.-y. Wu, Q.-x. Yang, and B.-l. Zhu, *Prediction of Railway Foundation Settlement Based on the BP Neural Network Model.* 2014.

60. Wang, Z., T. Kan, Y. Wang, Y. Zuo, D. Xia, D. Chen, A. Sun, and P. Han. *A study on structural deformation of platform screen door based on artificial neural network*. in *2015 International Conference on Fluid Power and Mechatronics (FPM)*. 2015. IEEE.

61. Domingo, L.M., J.A.V. Fernández-Villa, C.M. Sendra, and J.I.R. Herráiz, *An artificial neural network model as a preliminary track design tool.* Proc Inst Mech Eng F J Rail Rapid Transit, 2016. **230**(4): p. 1105-1117.

62. Chalouhi, E.K., I. Gonzalez, C. Gentile, and R. Karoumi, *Damage detection in railway bridges using Machine Learning: application to a historic structure.* Procedia Eng., 2017. **199**: p. 1931-1936.

63. Poyhonen, S., M. Conti, A. Cavallini, G.C. Montanari, and F. Filippetti. *Insulation defect localization through partial discharge measurements and numerical classification*. 2004. IEEE.

64. Ferguson, M., R. Ak, Y.-T.T. Lee, and K.H. Law. *Automatic localization of casting defects with convolutional neural networks*. 2005. IEEE.

65. Lasisi, A. and N. Attoh-Okine, *Principal components analysis and track quality index: A machine learning approach.* Transp. Res. Part C Emerg. Technol., 2018. **91**: p. 230-248.

66. !!! INVALID CITATION !!! [68].

67. Soleimanmeigouni, I., A. Ahmadi, A. Nissen, and X. Xiao, *Prediction of railway track geometry defects: a case study.* Struct. Infrastruct. Eng., 2020. **16**(7): p. 987-1001.

68. Huang, K., J. Wu, X. Yang, Z. Gao, F. Liu, and Y.J.J.o.A.T. Zhu, *Discrete train speed profile optimization for urban rail transit: A data-driven model and integrated algorithms based on machine learning.* J. Adv. Transp., 2019. **2019**.

69. Alawad, H., S. Kaewunruen, and M. An, *Learning From Accidents: Machine Learning for Safety at Railway Stations.* IEEE Access, 2020. **8**: p. 633-648.

70. Núñez, A., J. Hendriks, Z. Li, B.D. Schutter, and R. Dollevoet. *Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study*. in *2014 IEEE International Conference on Big Data (Big Data)*. 2014.

71. Li, Z., M. Molodova, A. Núñez, and R. Dollevoet, *Improvements in Axle Box Acceleration Measurements for the Detection of Light Squats in Railway Infrastructure.* IEEE Trans. Ind. Electron., 2015. **62**(7): p. 4385-4397.

72. Al Sehrawy, R., B. Kumar, and O. Amoudi, *Exploitation of BIM in Planning and Controlling the Construction Phase On-site Carbon Emissions: A 6D BIM Case Study.* Innovative Production And Construction: Transforming Construction Through Emerging Technologies, 2019: p. 279.

73. Tchana, Y., G. Ducellier, and S. Remy, *Designing a unique Digital Twin for linear infrastructures lifecycle management.* Procedia CIRP, 2019. **84**: p. 545-549.

74. Heaton, J., A.K. Parlikad, and J. Schooling, *Design and development of BIM models to support operations and maintenance.* Comput Ind, 2019. **111**: p. 172-186.

75. Fargnoli, M., A. Lleshaj, M. Lombardi, N. Sciarretta, and G. Di Gravio, *A BIM-based PSS Approach for the Management of Maintenance Operations of Building Equipment.* bldg, 2019. **9**(6): p. 139.

76. Shim, C.-S., N.-S. Dang, S. Lon, and C.-H. Jeon, *Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model.* Struct. Infrastruct. Eng., 2019: p. 1-14.

77. Dang, N. and C. Shim. *BIM authoring for an image-based bridge maintenance system of existing cable-supported bridges*. in *IOP Conference Series: Earth and Environmental Science*. 2018. IOP Publishing.

78. Chen, W., K. Chen, J.C. Cheng, Q. Wang, and V.J. Gan, *BIM-based framework for automatic scheduling of facility maintenance work orders.* Autom. Constr., 2018. **91**: p. 15-30.

79. Wanigarathna, N., K. Jones, A. Bell, and G. Kapogiannis, *Building information modelling to support maintenance management of healthcare built assets.* Facilities, 2019. **37**(7/8): p. 415-434.

80. Suchocki, M., *The BIM-for-rail opportunity.* WIT Trans. Built Environ., 2017. **169**: p. 37-44.

81. Wetzel, E.M. and W.Y. Thabet, *A case study towards transferring relevant safety information for facilities maintenance using BIM.* 2018.

82. Kim, T. and S. Kim, *A Study on BEM Application for the Efficient Maintenance Management of Bridge Structures.* International Information Institute (Tokyo). Information, 2017. **20**(5B): p. 3691-3698.

83. Hamzah, N., Z.-A. Ismail, S.N. Kamaruzzaman, A.A. Mutalib, and A.I.C. Ani. *A CMMS Expert using BIM for IBS Building Maintenance*. in *MATEC Web of Conferences*. 2016. EDP Sciences.

84. Meadati, P. and J. Irizarry, *BIM and QR Code for Operation and Maintenance*, in *Computing in Civil Engineering 2015*. 2015. p. 556-563.

85. Ilter, D.A. and E. Ergen, *BIM for building refurbishment and maintenance: current status and research directions.* Struct. surv., 2015. **33**: p. 228-256.

86. Alwan, Z., *BIM performance framework for the maintenance and refurbishment of housing stock.* Structural Survey, 2016. **34**: p. 242-255.

87. Fregonese, L., C. Achille, A. Adami, F. Fassi, A. Spezzoni, and L. Taffurelli, *BIM: An integrated model for planned and preventive maintenance of architectural heritage*. 2015. 77-80.

88. Sanchez, A., J. Kraatz, K. Hampson, and S. Loganathan, *BIM for Sustainable Whole-of-life Transport Infrastructure Asset Management*. 2014.

89. Lee, S.H., S. Park, J. Park, and K.W. Seo, *Open BIM-Based Information Modeling of Railway Bridges and its Application Concept*. 2014. 504-511.

90. Bosch, A., L. Volker, A. Koutamanis, M. Kumaraswamy, and P. Love, *BIM in the operations stage: bottlenecks and implications for owners.* Built Environ. Proj. Asset Manag., 2015. **5**.

91. Motawa, D.I. and A. Almarshad, *A knowledge-based BIM system for building maintenance.* Autom. Constr., 2013. **29**: p. 173–182.

92. Bynum, P., R.R.A. Issa, and S. Olbina, *Building Information Modeling in Support of Sustainable Design and Construction.* J. Constr. Eng. Manag., 2013. **139**(1): p. 24-34.

93. Cho, H., K. Lee, S. Lee, T. Lee, H. Cho, S. Kim, and S. Nam. *Introduction of Construction management integrated system using BIM in the Honam High-speed railway lot No. 4-2*. in *Proceedings of the 28th ISARC*. 2011. Seoul, Korea

94. Cai, Z., *Modelling of rail track dynamics and wheel/rail interaction.* 1994.

95. Steffens, D.M., *Identification and development of a model of railway track dynamic behaviour.* 2005.

96. Leong, J., *Development of a limit state design methodology for railway track.* 2007.

97. Kaewunruen, S. and C. Chiengson, *Railway track inspection and maintenance priorities due to dynamic coupling effects of dipped rails and differential track settlements.* Eng. Fail. Anal., 2018. **93**: p. 157-171.

98. Jain, M. *Data Science, Best Data Science Practices, Difference between AI, ML and DL and Technology stack !* 2018; Available from: https://medium.com/analytics-vidhya/data-science-best-data-science-practices-difference-between-ai-ml-and-dl-and-technology-stack-f8bc899e3791.

99. International Organization for Standardization, *ISO 15186-3:2002*, in *Acoustics — Measurement of sound insulation in buildings and of building elements using sound intensity — Part 3: Laboratory measurements at low frequencies*. 2002.

100. Buber, E. and D. Banu. *Performance analysis and CPU vs GPU comparison for deep learning*. in *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*. 2018. IEEE.

101. Autodesk Inc. *BIM*. 2019 [cited 2019 25 December]; Available from: https://www.autodesk.com/solutions/bim.

102. Jergéus, J., C. Odenmarck, R. Lundén, P. Sotkovszki, B. Karlsson, and P. Gullers, *Full-scale railway wheel flat experiments.* Proc. Inst. Mech. Eng. F: J. Rail Rapid Transit, 1999. **213**(1): p. 1-13.

103. Iwnicki, S., *Handbook of railway vehicle dynamics*. 2006: CRC press.

104. Brizuela, J., C. Fritsch, and A. Ibáñez, *Railway wheel-flat detection and measurement by ultrasound.* Transp. Res. Part C Emerg. Technol., 2011. **19**(6): p. 975-984.

105. Dong, R.G., S. Sankar, and R.V. Dukkipati, *A Finite Element Model of Railway Track and its Application to the Wheel Flat Problem.* Proc. Inst. Mech. Eng. F: J. Rail Rapid Transit, 1994. **208**(1): p. 61-72.

106. Bracciali, A. and G. Cascini, *Detection of corrugation and wheelflats of railway wheels using energy and cepstrum analysis of rail acceleration.* Proc. Inst. Mech. Eng. F: J. Rail Rapid Transit, 1997. **211**(2): p. 109-116.

107. Yue, J., Z. Qiu, and B. Chen. *Application of wavelet transform to defect detection of wheelflats of railway wheels*. in *6th International Conference on Signal Processing, 2002.* 2002.

108. Belotti, V., F. Crenna, R.C. Michelini, and G.B. Rossi. *Wavelet signal processing applied to railway wheelflat detection*. in *Proc. XVII IMEKO World Congress Metrology in the Third Millennium, Dubrovnik, Croatia*. 2003.

109. Filograno, M.L., P. Corredera, M. Rodríguez-Plaza, A. Andrés-Alguacil, and M. González-Herráez, *Wheel Flat Detection in High-Speed Railway Systems Using Fiber Bragg Gratings.* IEEE Sens. J., 2013. **13**(12): p. 4808-4816.

110. Jia, S. and M. Dhanasekar, *Detection of Rail Wheel Flats using Wavelet Approaches.* Struct. Health Monit., 2007. **6**(2): p. 121-131.

111. Li, Y., M.J. Zuo, J. Lin, and J. Liu, *Fault detection method for railway wheel flat using an adaptive multiscale morphological filter.* Mech. Syst. Signal Process, 2017. **84**: p. 642-658.

112. Krummenacher, G., C.S. Ong, S. Koller, S. Kobayashi, and J.M. Buhmann, *Wheel defect detection with machine learning.* IEEE trans Intell Transp Syst, 2017. **19**(4): p. 1176-1187.

113. Vyas, N. and A. Gupta, *Modeling rail wheel-flat dynamics*, in *Engineering Asset Management*. 2006, Springer. p. 1222-1231.

114. Zhai, W., *Vehicle–track coupled dynamics: theory and applications*. 2019: Springer Nature.

115. Zeng, M., L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, and J. Zhang. *Convolutional neural networks for human activity recognition using mobile sensors*. in *6th International Conference on Mobile Computing, Applications and Services*. 2014. IEEE.

116. Network Rail. *Visual Track Inspection*. 2019 [cited 2022 5 May]; Available from: https://www.networkrail.co.uk/wp-content/uploads/2019/06/Visual-Track-Inspections-v1.pdf.

117. Network Rail. *Ultrasonic Rail Inspection*. 2019 [cited 2022 5 May]; Available from: https://www.networkrail.co.uk/wp-content/uploads/2019/06/Ultrasonic-Rail-Inspection-v1.pdf.

118. Bray, D.E. *Historical review of technology development in NDE*. in *Proceedings of the 15th World Conference on NDT, Roma, Italy*. 2000.

119. Rockstroh, B., W. Kappes, F. Walte, M. Kröning, S. Bessert, W. Schäfer, R. Schallert, W. Bähr, D. Joneit, and A. Montnacher. *Ultrasonic and eddy-current inspection of rail wheels and wheel set axles*. in *17th World Conference on Nondestructive Testing*. 2008.

120. Deutschl, E., C. Gasser, A. Niel, and J. Werschonig. *Defect detection on rail surfaces by a vision based system*. in *IEEE Intelligent Vehicles Symposium, 2004*. 2004. IEEE.

121. Mandriota, C., M. Nitti, N. Ancona, E. Stella, A.J.M.V. Distante, and Applications, *Filter-based feature selection for rail defect detection.* 2004. **15**(4): p. 179-185.

122. Jie, L., L. Siwei, L. Qingyong, Z. Hanqing, and R. Shengwei. *Real-time rail head surface defect detection: A geometrical approach*. in *2009 IEEE International Symposium on Industrial Electronics*. 2009. IEEE.

123. Feng, H., Z. Jiang, F. Xie, P. Yang, J. Shi, and L. Chen, *Automatic fastener classification and defect detection in vision-based railway inspection systems.* IEEE Trans Instrum Meas, 2013. **63**(4): p. 877-888.

124. Kang, G., S. Gao, L. Yu, and D. Zhang, *Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning.* IEEE Trans Instrum Meas, 2018. **68**(8): p. 2679-2690.

125. Yu, H., Q. Li, Y. Tan, J. Gan, J. Wang, Y.-a. Geng, and L. Jia, *A coarse-to-fine model for rail surface defect detection.* IEEE Trans Instrum Meas, 2018. **68**(3): p. 656-666.

126. Wei, X., Z. Yang, Y. Liu, D. Wei, L. Jia, and Y. Li, *Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study.* Eng. Appl. Artif. Intell., 2019. **80**: p. 66-81.

127. Martey, E.N., L. Ahmed, and N. Attoh-Okine. *Track geometry big data analysis: A machine learning approach*. in *2017 IEEE International Conference on Big Data (Big Data)*. 2017.

128. Sadeghi, J. and H. Askarinejad, *An investigation into the effects of track structural conditions on railway track geometry deviations.* Proc. Inst. Mech. Eng. F: J. Rail Rapid Transit, 2009. **223**(4): p. 415-425.

129. Zarembski, A.M., D. Einbinder, N.J.C. Attoh-Okine, and B. Materials, *Using multiple adaptive regression to address the impact of track geometry on development of rail defects.* 2016. **127**: p. 546-555.

130. Berggren, E., *Railway track stiffness: dynamic measurements and evaluation for efficient maintenance.* 2009.

131. Sussmann Jr, T.R., T.D. Stark, S.T. Wilk, and H.B. Thompson, *Track support measurements for improved resiliency of railway infrastructure.* Transp. Res. Rec., 2017. **2607**(1): p. 54-61.

132. Kim, H., *Trackside measurement of critical zones in railway tracks.* 2016.

133. Zhu, J.Y., D.J. Thompson, and C.J.C. Jones, *On the effect of unsupported sleepers on the dynamic behaviour of a railway track.* Veh. Syst. Dyn., 2011. **49**(9): p. 1389-1408.

134. Augustin, S., G. Gudehus, G. Huber, and A. Schnemann, *System Dynamics and Long-Term Behaviour of Railway Vehicles, Track and Subgrade, volume 6, chapter Numerical Model and Laboratory Tests on Settlement of Ballast Track*. 2002, Springer.

135. Mosayebi, S.-A., M. Esmaeili, and J.-A. Zakeri, *Numerical investigation of the effects of unsupported railway sleepers on train-induced environmental vibrations.* J. Low Freq. Noise Vib. Act. Control, 2017. **36**(2): p. 160-176.

136. Zhang, S., X. Xiao, Z. Wen, and X. Jin, *Effect of unsupported sleepers on wheel/rail normal load.* Soil Dyn. Earthq. Eng., 2008. **28**(8): p. 662-673.

137. Sadeghi, J., J.-A. Zakeri, and A.R. Tolou Kian. *Effect of unsupported sleepers on rail track dynamic behaviour*. in *Proceedings of the Institution of Civil Engineers-Transport*. 2018. Thomas Telford Ltd.

138. Zakeri, J.A., M. Fattahi, and M.M. Ghanimoghadam, *Influence of unsupported and partially supported sleepers on dynamic responses of train–track interaction.* J. Mech. Sci. Technol, 2015. **29**(6): p. 2289-2295.

139. Kim, D.-S., S.D. Kim, and J. Lee. *Easy detection and dynamic behavior of unsupported sleepers in high speed ballasted track*. 2006.

140. Li, T., Q. Su, and S. Kaewunruen, *Influences of dynamic material properties of slab track components on the train-track vibration interactions.* Eng. Fail. Anal., 2020. **115**: p. 104633-104648.

141. Cai, C.-b., W.-m. Zhai, and K.-y. Wang, *遂渝线路基上板式轨道动力性能计算及评估分析[Calculation and assessment analysis of the dynamic performance for slab track on Sui-Yu railway].* China Railway Science, 2006. **27**(4): p. 17-21.

142. Ngamkhanong, C., Q.Y. Ming, T. Li, and S. Kaewunruen, *Dynamic train-track interactions over railway track stiffness transition zones using baseplate fastening systems.* Eng. Fail. Anal., 2020. **118**: p. 104866.

143. Kassa, E., C. Andersson, and J.C.O. Nielsen, *Simulation of dynamic interaction between train and railway turnout.* Veh. Syst. Dyn., 2006. **44**(3): p. 247-258.

144. Andersson, C. and T. Dahlberg, *Wheel/rail impacts at a railway turnout crossing.* Proc Inst Mech Eng F J, 1998. **212**(2): p. 123-134.

145. Dindar, S., S. Kaewunruen, and M. An, *Identification of appropriate risk analysis techniques for railway turnout systems.* J. Risk Res., 2018. **21**(8): p. 974-995.

146. Benn, J., *Railway bridge failure during flooding in the UK and Ireland.* Proc. Inst. Civ. Eng.: Forensic Eng., 2013. **166**(4): p. 163-170.

147. Dindar, S. and S. Kaewunruen. *Assessment of turnout-related derailments by various causes*. in *Sustainable Civil Infrastructures: Innovative Infrastructure Geotechnology*. 2017. Springer.

148. Xin, L., V.L. Markine, and I.Y. Shevtsov, *Analysis approach of turnout crossing performance by field measurements and finite element modeling*, in *The Dynamics of Vehicles on Roads and Tracks*. 2016, CRC Press. p. 1576-1583.

149. Hasnayn, M.M., W.J. McCarter, P.K. Woodward, D.P. Connolly, and G. Starrs, *Railway subgrade performance during flooding and the post-flooding (recovery) period.* Transp. Geotech., 2017. **11**: p. 57-68.

150. Hamarat, M., M. Papaelias, and S. Kaewunruen, *Vulnerability of railway switches and crossings exposed to flooding conditions*, in *Resilient Infrastructure*. 2022, Springer. p. 337-348.

151. Li, M.X.D. and E.G. Berggren, *A Study of the Effect of Global Track Stiffness and Its Variations on Track Performance: Simulation and Measurement.* Proc Inst Mech Eng F J Rail Rapid Transit, 2010. **224**(5): p. 375-382.

152. Kaewunruen, S., *Monitoring structural deterioration of railway turnout systems via dynamic wheel/rail interaction.* Case Stud. Nondestruct. Test. Evaluation, 2014. **1**: p. 19-24.

153. Cao, W.-J., S. Zhang, N.J. Bertola, I.F.C. Smith, and C.G. Koh, *Time series data interpretation for 'wheel-flat' identification including uncertainties.* Struct. Health Monit., 2019: p. 1475921719887117.

154. Sysyn, M., D. Gruen, U. Gerber, O. Nabochenko, and V. Kovalchuk, *Turnout monitoring with vehicle based inertial measurements of operational trains: a machine learning approach.* Commun. - Sci. Lett. Univ. Zilina., 2019. **21**(1): p. 42-48.

155. Hamarat, M., S. Kaewunruen, M. Papaelias, and M. Silvast, *New insights from multibody dynamic analyses of a turnout system under impact loads.* Appl. Sci., 2019. **9**(19): p. 4080.

156. Wan, C., V.L. Markine, and I.Y. Shevtsov, *Analysis of train/turnout vertical interaction using a fast numerical model and validation of that model.* Proc Inst Mech Eng F J Rail Rapid Transit, 2014. **228**(7): p. 730-743.

157. Wei, Z., A. Núñez, Z. Li, and R. Dollevoet, *Evaluating Degradation at Railway Crossings Using Axle Box Acceleration Measurements.* Sensors, 2017. **17**(10).

158. Sawley, K. and R.P. Reiff, *Rail Failure Assessment for the Office of the Rail Regulator: An Assessment of Railtrack's Methods for Managing Broken and Defective Rails*. 2000: Transportation Technology Center.

159. Fay, M., *Rail Grinding*. 2007.

160. Sresakoolchai, J. and S. Kaewunruen, *Wheelflat detection and severity classification using deep learning techniques.* Insight: Non-Destr. Test. Cond. Monit., 2021.

161. Australian Rail Track Corporation, *Rail Defects Handbook*. 2006.

162. ENGINEER SUPPLY. *HOW ACCURATE ARE LASER DISTANCE MEASUREMENTS TOOLS?* 2020; Available from: https://www.engineersupply.com/How-Accurate-Are-Laser-Distance-Measurements-Tools.aspx.

163. Senix Distance and Level Sensors. *Ultrasonic Sensor Accuracy*. 2020; Available from: https://senix.com/ultrasonic-sensor-accuracy/.

164. Desai, A., *Defects in Cast-in-situ Ballastless track.* IRJET, 2016. **3**(8): p. 130-134.

165. Guo, Y. and W. Zhai, *Long-term prediction of track geometry degradation in high-speed vehicle–ballastless track system due to differential subgrade settlement.* Soil Dyn. Earthq. Eng., 2018. **113**: p. 1-11.

166. Kaewunruen, S., C. Ngamkhanong, and J. Ng, *Influence of time-dependent material degradation on life cycle serviceability of interspersed railway tracks due to moving train loads.* Eng. Struct., 2019. **199**: p. 109625-109638.

167. McCarthy, W.T. *Track geometry measurement on Burlington Northern Railroad*. in *Nondestructive Evaluation of Aging Railroads*. 1995. Oakland, CA, United States: International Society for Optics and Photonics.

168. Transport for New South Wales, *TMC 203 - Track inspection*. 2011.

169. Weston, P., C. Roberts, G. Yeo, and E. Stewart, *Perspectives on railway track geometry condition monitoring from in-service railway vehicles.* Veh. Syst. Dyn., 2015. **53**(7): p. 1063-1091.

170. Ágh, C., *Comparative analysis of axlebox accelerations in correlation with track geometry irregularities.* Acta tech. Jaurinensis., 2019. **12**(2): p. 161-177.

171. Li, D., A. Meddah, K. Hass, and S. Kalay, *Relating track geometry to vehicle performance using neural network approach.* P I MECH ENG F-J RAI, 2006. **220**(3): p. 273-281.

172. Tsunashima, H., Y. Naganuma, and T. Kobayashi, *Track geometry estimation from car-body vibration.* Veh. Syst. Dyn., 2014. **52**(sup1): p. 207-219.

173. Zaiontz, C. *K-means Cluster Analysis*. 2015; Available from: https://www.real-statistics.com/multivariate-statistics/cluster-analysis/k-means-cluster-analysis/.

174. Alibabaei, K., P.D. Gaspar, E. Assunção, S. Alirezazadeh, T.M. Lima, V.N.G.J. Soares, and J.M.L.P. Caldeira, *Comparison of on-policy deep reinforcement learning A2C with off-policy DQN in irrigation optimization: A case study at a site in Portugal.* Computers, 2022. **11**(7): p. 104.

175. Sedghi, M., O. Kauppila, B. Bergquist, E. Vanhatalo, and M. Kulahci, *A taxonomy of railway track maintenance planning and scheduling: A review and research trends.* Reliab. Eng. Syst. Saf., 2021. **215**: p. 107827.

176. Lopes Gerum, P.C., A. Altay, and M. Baykal-Gürsoy, *Data-driven predictive maintenance scheduling policies for railways.* Transp. Res. Part C Emerg. Technol., 2019. **107**: p. 137-154.

177. Mohammadi, R. and Q. He, *A deep reinforcement learning approach for rail renewal and maintenance planning.* Reliab. Eng. Syst. Saf., 2022. **225**: p. 108615.

178. Sutton, R.S. and A.G. Barto, *Reinforcement learning: An introduction*. 2018: MIT press.

179. Rungskunroch, P., Z.-J. Shen, and S. Kaewunruen, *Getting it right on the policy prioritization for rail decarbonization: Evidence from whole-life CO2e emissions of railway systems.* Front. Built Environ., 2021: p. 60.

180. Yarram, S. and V.V. Krishna *COMPARISON OF REINFORCEMENT LEARNING ALGORITHMS*. 2022.

181. OpenAI. *Proximal Policy Optimization*. 2017; Available from: https://openai.com/blog/openai-baselines-ppo/.

182. Kaewunruen, S., J. Sresakoolchai, and Y.-h. Lin, *Digital twins for managing railway maintenance and resilience.* Open Research Europe, 2021. **1**(91): p. 91.

183. Meddah, M.S. *Recycled aggregates in concrete production: engineering properties and environmental impact*. 2017. EDP Sciences.

184. Rempelos, G., J. Preston, and S. Blainey, *A carbon footprint analysis of railway sleepers in the United Kingdom.* Transp Res D Transp Environ, 2020. **81**: p. 102285.

185. Hall, J., *Cleaning Up The Steel Industry: Reducing CO2 Emissions with CCUS.* 2021.

186. Network Rail, *Track treatment fleet*. 2022.

187. Pouget, *MACHINES FOR TRACK WORK.* 2022.

188. Progressrail, *KERSHAW® KSC2000 SHOULDER CLEANER.* 2022.

189. Geismar, *SLEEPER CHANGING MACHINE.* 2022.

190. Vossloh, *Rail Replacement Wagon SWW.* 2022.

191. Anyang General International Co., L., *Rail Fastening System.* 2022.

192. European Patent Office, *Wagon for unloading ballast on a railway.* 2007.