UNIVERSITY OF BIRMINGHAM

Enabling Methods for Predictive Digital Twin in Pavement Performance Modelling

by

KUN CHEN

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

DOCTOR OF PHILOSOPHY

School of Civil Engineering

College of Engineering and Physical Sciences

The University of Birmingham

September 2024

ABSTRACT

Roads are vital assets and the backbone for any transportation system and support societal development by providing the foundation for constant mobility of goods and people. However, pavements are experiencing accelerated deterioration in most developed countries due to increased traffic volume and load, combined with rapidly changing climate. The existing reactive road asset management approach cannot keep up with the rate of pavement degradation, due to lack of condition data from infrequent inspection surveys and simple models that do not consider the factors influencing pavement performance holistically.

Digital twins have been popularly utilised in recent years enabled by the increasing capacity in data collection using intelligent sensors, digital innovations with technologies such as internet of things, cloud computing, big data analytics with machine learning, as well as artificial intelligence. Despite the growing interest in applications of digital twins in the built environment such as bridges and buildings, current digital twin research related to roads is still at an early stage.

To this end, this study investigates the development of digital twins for the road sector. Based on the literature, a digital twin-based decision-making support theoretical framework for road lifecycle is presented and discussed. In particular, two case studies, as applications of this framework, are conducted to demonstrate the impact of predictive digital twins on roads in the areas of pavement performance and data collection. As part of the road digital twin framework, it is found that integrating physics-based simulation with machine learning, decreased the root mean squared error by at least 25% compared to traditional machine learning in one year prediction, and reduced the 90^{th} percentile range in multi-year predictions by as much as over 30%. In addition, this research also identifies that a

substantial amount (approx. over 95%) of sensor data collected could be reduced while achieving acceptable prediction accuracy, thereby minimising the data related costs within the same framework. The findings are useful for the understanding and consideration of the on-going road digital twin development.

Keywords:

Digital Twins**,** Sensor Data, Data Collection Frequency, Machine Learning, Physics Enhanced Machine Learning, Physics-based Simulation, Road Asset Management, Pavement Performance Prediction, Uncertainty Quantification

PUBLICATIONS

The results of this PhD study have been published by or submitted to the following peer-reviewed journals:

Chen, K., Torbaghan, M. E., Chu, M., Zhang, L., & Garcia, A. (2022). Identifying the most suitable machine learning approach for a road digital twin. **Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction**, 174(3), 88–101. https://doi.org/10.1680/jsmic.22.00003

Chen, K., Torbaghan, M. E., Thom, N., Garcia-Hernández, A., Faramarzi, A., & Chapman, D. (2024). A Machine Learning based approach to predict road rutting considering uncertainty. **Case Studies in Construction Materials**, e03186. https://doi.org/10.1016/J.CSCM.2024.E03186

Chen, K., Torbaghan, M. E., Thom, N., A., Faramarzi, A. (2024). Physics-Enhanced Machine Learning for Predicting International Roughness Index on Flexible Pavements Considering Uncertainty. **Engineering Applications of Artificial Intelligence** (under review)

DECLARATION

The research reported in this thesis was jointly supervised by the University of Birmingham and the University of Nottingham under the Universitas 21 joint PhD scheme. The research, undertaken between October 2020 and September 2024, was primarily carried out at The University of Birmingham. The University of Nottingham, Nottingham Transportation Engineering Centre hosted the researcher for multiple research visits since 2022 till the end of the PhD.

The contents of this thesis are the original work of the researcher, except where appropriately acknowledged and referenced in the text, and no part of this thesis has been submitted for a degree at another institution.

Kun Chen

ACKNOWLEDGEMENTS

The author would like to acknowledge and sincerely thank his supervisors, Dr Mehran Eskandari Torbaghan (University of Birmingham), and Dr Nick Thom (University of Nottingham), for whom without the success of this research would not have been possible. The author is appreciative of their dedication, in-depth knowledge, inspiration, guidance and continuous support throughout the whole journey of the PhD.

The author would also like to further acknowledge the wider supervision team, Dr Asaad Faramarzi, Prof David Chapman (University of Birmingham), and Prof Alvaro Garcia (RWTH Aachen University) and Dr Darren Prescott (University of Nottingham) and thank their time, inputs and their help at various important stages of the research.

The author wishes to appreciate the annual reviewers from both institutions, Dr Michael Burrow and Dr Jelena Ninic for their most useful comments and encouragements.

The Universitas 21 is also thanked with gratitude for providing the scholarship to financially support this research.

TABLE OF CONTENTS

ABSTRACT i
PUBLICATIONS iii
DECLARATION iv
ACKNOWLEDGEMENTS v
TABLE OF CONTENTS vi
LIST OF FIGURES ix
LIST OF TABLES xi
ABBREVIATIONS xii
1 INTRODUCTION 1
1.1 Digitalisation and Digital Twins 3
1.2 Aim and Objectives 6
1.3 Research Scope and Limitations 7
1.4 Contribution of the Thesis 7
1.5 Structure of the Thesis 8
2 LITERATURE REVIEW 10
2.1 Road Pavement Management and Maintenance 10
2.1.1 Introduction 10
2.1.2 Pavement Performance Prediction and Modelling Concepts 15
2.1.3 Data Used for Pavement Deterioration Modelling 16
2.1.4 Mechanism of Pavement Deterioration 25
2.1.5 Pavement Deterioration Models 30
2.2 DTs and their Impacts on Roads 50
2.2.1 Concepts and Definitions 50
2.2.2 Categorisations and Types 52
2.2.3 System Architectures 55
2.2.4 Enabling Technologies 58
2.2.5 Main Components and Characteristics 60
2.2.6 Applications and Benefits 67
2.2.7 DTs in Construction Industry and Built Environment 69
2.2.8 DT Impacts on Roads 71
2.3 Optimisation methods for data collection frequency 74
2.4 Summary of the Research Gaps 77

3 METHODOLOGY 78
3.1 Introduction 78
3.2 Research Methodology 78
3.3 A DT-based decision-making support theoretical Framework for road lifecycle 80
3.4 Scope of the Study 86
3.5 RDT Framework's Modelling Methodology 87
3.5.1 Introduction to common ML models in pavement performance modelling 89
3.6 Introduction to the Case Studies 92
3.6.1 Case Study I – Pavement Performance Modelling with Historical Data 92
3.6.2 Case Study II – Experiment at UK National Buried Infrastructure Facility (NBIF) Using Sensor Data 118
3.7 Summary of the Methodology 134
3.8 Utilised Software and Packages 135
4 RESULTS AND DISCUSSIONS 136
4.1 Predictions on Rutting 136
4.1.1 Short-Term Predictions 136
4.1.2 Prediction Intervals 139
4.1.3 Multi-Year 90th Percentile Range Predictions 142
4.1.4 Key Findings of Rutting Predictions 146
4.2 Predictions on IRI 147
4.2.1 Short-Term Predictions 147
4.2.2 Prediction Intervals 151
4.2.3 Multi-Year 90th Percentile Range Predictions 153
4.2.4 Key Findings of IRI Predictions 157
4.3 Optimisation of Data Collection Frequency Using NBIF Sensor Data 158
4.4 Overall Discussions 168
4.4.1 RDT for Future PMS 168
4.4.2 Development of RDT 171
4.4.3 Summary of the Research 173
4.4.4 Value of the Research 175
5 CONCLUSION AND FUTURE WORKS 177
5.1 Accomplished Work and Main Findings 177
5.2 Recommendations for Further Research 180
6 REFERENCES 184

Appendix A: Data Pre-processing and Cleaning Python Script..... I
Appendix B: Exhaustive Feature Selection for Rutting and IRI.....V
Appendix C: RF Model Development Including Hyperparameter Tuning.....V
Appendix D: ANN Model Development Including Hyperparameter Tuning.....VI
Appendix E: Script to Generate Prediction Intervals Based on a List of Actual and Predicted Values Produced by Models..... VII
Appendix F: Uncertainty Quantification with 90^{th} percentile range predictions X
Appendix G: Python Scripts for NBIF Data Analysis X
G.1 Reading Sensor Data.....XI
G.2 Sensor Data Fusion XVI
G.3 Data Preparation for Different Prediction Scenarios XVIII
G.4 Feature Selection XIX
G.5 RF Model Training and Testing on Sensor Data XIX

LIST OF FIGURES

Figure 2.1. Concepts of pavement performance prediction (After Haas et al., 1994) 16
Figure 2.2. Potential measurement of rutting growth over time (After Visintine et al., 2018) 22
Figure 2.3. Importance of perceived modes of deterioration in fully flexible pavements (After Atkinson et al., 2006) 27
Figure 2.4. Bump integrators and the IRI (After Thom, 2024) 29
Figure 2.5. Pavement performance prediction models developments and characteristics 30
Figure 2.6. PEML Techniques and Methods 42
Figure 2.7. Data Flow in a (a) Digital Model; (b) Digital Shadow; (c) Digital Twin 53
Figure 2.8. Five-Dimensional Digital Twin Architecture 55
Figure 2.9. Framework of enabling technologies for DTs (After Qi et al., 2021) 59
Figure 2.10. Visualisation of possible fusion operation types in DT ecosystem 63
Figure 3.1. Research methodology 80
Figure 3.2. DT architecture-based decision-making support theoretical framework for road lifecycle 81
Figure 3.3. Key steps taken to conduct the research. The black lines represent the steps taken for pure data-driven ML model development. The box dotted in red is related to the physics-based model, and the coloured lines in red and blue show two different approaches to interact the physics-based model with the data-driven model 88
Figure 3.4. Common ANN structure with two hidden layers 90
Figure 3.5. Locations and number of selected road sections from the US LTPP database 93
Figure 3.6. Data pre-processing procedures on rutting condition data 96
Figure 3.7. a) Pavement model and b) Its structural layers 103
Figure 3.8. RF model training, evaluation, and test iteration process 114
Figure 3.9. Process flow for ANN model training and evaluation 116
Figure 3.10. Selection process of the 15 test sections 116
Figure 3.11. NBIF experiment set up 119
Figure 3.12. NBIF data collection 119
Figure 3.13. NBIF experiment test in progress 120
Figure 3.14. A sketch of the actuator, pavement and sensors, including 2 LVDTs on the actuator (LVDT3 & 4) as well as 1 LVDT on the pavement surface (LVDT1) and a PC in the corner (PC-C) and one in the middle (PC-M), six temperature probes (T1-T6), and three SGs on top of the buried pipe 121
Figure 3.15. SG readings within a) one second and b) one loading cycle 124
Figure 3.16. Process flowchart for selecting sensor data collection frequency 127
Figure 4.1. One-year model deterministic prediction results of test sections using a) LTPP data only and b) LTPP + physics-based simulated deflection data 138
Figure 4.2. Prediction intervals with 90% confidence level on test sections for a) Scenario 1 and b) Scenario 2 140
Figure 4.3. Visualisation of prediction intervals with 90% confidence level on test sections for a) Scenario 1 and b) Scenario 2 after five runs of the model development process with different training data each time. Each colour represents the result of one run 141
Figure 4.4. Standard error vs. number of runs 143

Figure 4.5. Multi-year predictions of one model for Section 12-0566: a) Scenario 1 and b) Scenario 2 considering ML uncertainties 144
Figure 4.6. One-year model deterministic prediction results on test sections for three scenarios a) Scenario 1 b) Scenario 2 c) Scenario 3 148
Figure 4.7. One-year model deterministic prediction results on different test sections for three scenarios a) Scenario 1 b) Scenario 3 (traditional loss function) c) Scenario 3 (physics-informed loss function) 150
Figure 4.8. 90% prediction intervals on sections for three scenarios a) Scenario 1 b) Scenario 2 c) Scenario 3 152
Figure 4.9. Multi-year 90^{th} percentile range predictions of one model for Section 46-0608 for all scenarios a) Scenario 1 b) Scenario 2 c) Scenario 3 154
Figure 4.10. Performance comparisons for 10 times of run across all scenarios a) 90^{th} percentile range prediction for the 12^{th} year b) RMSE from the 2^{nd} year onwards c) RMSE for the 12^{th} year 155
Figure 4.11. Downsampled plots with different sampling rates 159
Figure 4.12. Performance metrics on test data when sampling every 20 minutes to predict the next 20 minutes 160
Figure 4.13. Prediction performance with different ranges in different sampling rates measured by a) R^2 b) RMSE c) MAE 162
Figure 4.14. Performance of predictions using different sampling rates across different ranges, measured by a) R^2 b) RMSE c) MAE 164

LIST OF TABLES

Table 2.1. Optimization techniques for pavement maintenance decision-making 13
Table 2.2. Details on multiple DT types 54
Table 2.3. Fusion operation types and benefit 62
Table 2.4. DT model updating methods 65
Table 3.1. Data issues and applied pre-processing techniques 95
Table 3.2. Descriptions of the features collected in this case study 97
Table 3.3. Top five exhaustive variable selection results for Rutting (t+1) 100
Table 3.4. Top five exhaustive variable selection results for IRI (t+1) 100
Table 3.5. Material codes and characteristics 103
Table 3.6. Assumptions made in the creation of Abaqus models 107
Table 3.7. Results for multiple ML algorithms 108
Table 3.8. Up-sampled data generation process, blue shade represents the original data, and the grey shade shows the generated additional up-sampled data 111
Table 3.9. Scenarios considered in this study for predictions 112
Table 3.10. RF hyperparameter space and tuning results 113
Table 3.11. ANN hyperparameter space and tuning results 115
Table 3.12. Sensor details and frequencies 122
Table 3.13. Experiment details 122
Table 3.14. Sensor data and meanings 125
Table 3.15. Descriptions of the sensor data collected in this case study 130
Table 3.16. Input and output features and results for their correlation coefficient and p-value, with selected features being highlighted in Green 132
Table 3.17. Features after the feature selection process 133
Table 3.18. Utilised software and packages 135
Table 4.1. Model performance for rutting one-year prediction 137
Table 4.2. The 12^{th} year rutting prediction details with 30 models for both scenarios 145
Table 4.3. Performance evaluation of different scenarios 156
Table 4.4. Performance metrics with different sampling rates across multiple prediction ranges 161

ABBREVIATIONS

3-Dimensional	3D
AADT	Annual Average Daily Traffic
AI	Artificial Intelligence
ANN	Artificial Neural Network
BIM	Building Information Model
DT	Digital Twin
ESAL	Equivalent Single Axle Load
FE	Finite Element
GIS	Geographic Information System
GPR	Gaussian Process Regression
HDM	Highway Development and Management
HPD	Hybrid Physics Data
IoT	Internet of Things
IRI	International Roughness Index
kN	Kilonewton
KNN	K-Nearest Neighbours
LiDAR	Light detection and ranging
LTPP	Long Term Pavement Performance
MAE	Mean Absolute Error
ML	Machine Learning
NBIF	National Buried Infrastructure Facility
NN	Neural Network

PEML	Physics-enhanced Machine Learning
PENN	Physics-enhanced Neural Network
PGML	Physics-guided Machine Learning
PIML	Physics-informed Machine Learning
PINN	Physics-informed Neural Network
PMS	Pavement Management System
R^2	Coefficient of Determination
RAM	Road Asset Management
RDT	Road Digital Twin
ReLU	Rectified Linear Unit
RF	Random Forest
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SE	Standard Error
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine

1 INTRODUCTION

Roads are one of the biggest and most valued physical infrastructures of any country, playing a pivotal role in supporting the economy and enabling societal development and productivity (Zhang and Cheng, 2023). However, roads experience continuous deterioration under traffic loads and being exposed to the natural world. Road condition has worsened in recent years in the 21st century due to the increasing demand on highways with rising traffic volume and average vehicle weights as well as more frequent extreme weather events (Gössling et al., 2023). Therefore, for transportation agencies and road management authorities, there is an urgent need to efficiently maintain the road assets to ensure their level of serviceability.

Road asset management (RAM) system has been adopted to help mitigate this problem. According to American Association of State Highway and Transportation Officials (2002), RAM is the systematic and strategic approach of managing and maintaining road infrastructure in order to maximize its value and life span. It includes a range of components and activities such as inventory, condition inspection by collecting data with a pre-defined frequency and condition prediction of road assets; maintenance and rehabilitation strategy prioritisations; budgeting and resource allocations; as well as monitoring and evaluation of performance to support the decision-making process (Zakir et al., 2024). RAM can be applied at both the network level and the project level. Network level activities include long-term planning, develop the optimum strategy for allocating pavement rehabilitation and maintenance funds over the entire network whereas project level activities focus on detailed planning and technical methods for the construction or maintenance of a particular roadway section (De La Garza et al., 2011).

Due to the limited budget available, it has been a challenging task to make the best use of the allocated funds to achieve the maintenance goals within RAM. The current approach when it

comes to road maintenance is still mostly reactive, and according to UK Annual Local Authority Road Maintenance report in 2024 (Asphalt Industry Alliance, 2024), both structural condition and the surface condition of UK local roads continue to decline, reflected by the cost of £16.3 billion to tackle the carriageway repairs backlog, and the fact that 2 million potholes were filled over the last year, up from 1.4 million, 16% and 40% increases since the previous year respectively. The constant degradation of roads and soaring maintenance expenses indicate an ineffective RAM approach that is not economically viable, and hence a more proactive approach is required to enable the maintenance prioritisations.

Proactive maintenance is the opposite to reactive with a focus on averting failure and performance decrease, and it refers to any form of maintenance activity that is done before any significant defects occur, especially considering of different factors such as traffic, materials, environmental and climatic impacts. All of these contribute to the decision-making of maintenance strategies prioritisation under a proactive approach (Fitch, 2013). It includes preventive maintenance (Abdulmohsen et al., 2013) and predictive maintenance (Patiño-Rodriguez and Carazas, 2019).

A proactive maintenance approach enables a reduction in the whole life cycle cost of the asset, as well as eliminates the associated risks posed by defects or failure due to the lack of timely maintenance. For instance, Tran et al. (2024) proposed a risk-based proactive asset management approach leveraging a failure assessment model for urban stormwater pipes, resulting in extended service life and significant cost savings over the whole life cycle. Similarly, Liu et al. (2020) suggested a risk-informed framework integrated with cost-benefit analysis to determine proactive maintenance actions for service life extension of ships leads to financially feasible strategies.

However, there are two main reasons that still hinder the adoption of a proactive approach to manage and maintain road assets. First of all, an infrequent condition inspection results in insufficient condition data to monitor the road deterioration process, and in addition there is a lack of an accurate, reliable condition predictive model that considers the whole picture around the pavement with as many factors as possible that could have an impact on pavement performance from a holistic point of view (Chen et al., 2022).

Pavement performance can be classified mainly into two types, namely Functional Performance and Structural Performance. Functional pavement performance assesses how well a pavement serves the public travelling, such as its availability and level of comfort. It is mostly measured by roughness index (Hussam et al., 2018). Whereas structural pavement performance is how well a pavement can carry traffic loading overtime considering various impacting factors such as structural parameters, environmental conditions and maintenance (Bhandari et al., 2023). This research intends to focus on the structural pavement performance modelling.

1.1 Digitalisation and Digital Twins

The advancement of digital technologies may help addressing this issue by enabling continuous monitoring of road condition, as well as accurate predictive modelling using the large amount of data available. Looking broadly, entering the industry 4.0 era triggered by the wave of digitalisation with technologies such as Internet of Things (IoT), Big Data Analytics, Sensor technologies, Machine Learning (ML) or Deep Learning, and Artificial Intelligence (AI), transformational impacts have been brought into the operation of various industries such as manufacturing (Felsberger et al., 2022), healthcare (Ricciardi, 2019), supply chain (Aarasse and Idelhakkar, 2023), automotive (Viale and Zouari, 2020), logistics and transportation (Loske and Klumpp, 2022; Muntaka et al., 2023). The generalised benefits

that come with it can be summarised as faster outcomes through automated processes, reduced costs, improved operational efficiency and productivity (Arsić, 2020; Fähndrich, 2023).

There has also been a great effort in digitalisation when it comes to the construction industry, despite it being one of the oldest industries and often known as very conservative in facing changes and technology innovations. In the pavement domain, there has been an increasing amount of attention to resolve pavement engineering problems by researchers (Karimzadeh, 2020). Among both academics and practitioners in industry, pavement engineers have recognised that a large amount of data from sensor networks can provide continuous and useful information on pavement behaviour and performance and is able to provide a more comprehensive understanding of the pavement status if combined well with visualisation on the sensor data. For example, a Smart Winter project in Kent demonstrated the benefits of installing real-time surface temperature sensors to allow operational managers to see how their roads fluctuate in temperature enabling an immediate improvement in gritting schedules (Trousdale, 2019). Similarly, pavement temperature sensor data can be analysed in combination with material sensors to understand the effects of sudden environmental changes on the behaviour and performance expected from the pavement structure (Steyn, 2020).

In addition, it provides the opportunity to identify the optimised data collection frequency when it comes to sensor instrumentations and configurations. Specifically, with the growing amount of real-time or sensor data from various sources, data analytics approaches have been leveraged to perform improved asset deterioration modelling and to enable prediction with high accuracy for the performance of road assets (Piryonesi, 2019). With regards to pavement performance prediction, multiple different ML, neural network (NN) and deep learning

architectures as well as algorithms have been explored to predict pavement performance based on a data-driven approach (Amirhossein Hosseini, 2020; Choi and Do, 2020).

As a result of the evolution of these digital technologies, one novel concept in particular, Digital Twin (DT), has recently become a popular research area for architecture, engineering and construction management and has shown a great potential to support intelligent and automated decision-making as a tool for asset whole lifecycle management and therefore optimising the operation and maintenance strategies (Macchi et al., 2018).

Initially, the DT concept was first introduced by Professor Michael Grieves in 2002 who later published a white paper, providing a formal definition that a digital twin mainly comprises of three parts: 1) physical entities in real space; 2) virtual models in virtual space; 3) data that connects physical entities and virtual models together (Grieves, 2014). Then different organisations within various industries started to produce their own definitions and interpretations on DT concept based on domain-specific functions and characteristics due to the broad applicability of DTs (Kritzinger et al., 2018). However, a common understanding and study of DTs done by Kritzinger et al. (2018) has classified it into three categories depending on the level of integration between the physical and virtual entities, namely Digital Moel, Digital Shadow, and the fully integrated Digital Twin. In addition, multiple research projects have described different levels of DT maturity including 1) descriptive twin; 2) informative or diagnostic twin; 3) predictive twin; 4) prescriptive twin (Madni et al., 2019; Babanagar et al., 2025). More details on DTs are provided in the Literature Review chapter.

DT can provide an array of benefits for infrastructure lifecycle management such as real-time data monitoring and diagnostics, proactive maintenance, maintenance prioritisation as well as automation and optimisation (Arisekola and Madson, 2023). Given all infrastructure deteriorates over time, a DT that incorporates data from various factors that would have an

impact on the deterioration, such as climate, ambient environment, material, underlying science, available budget, potential risks, and stakeholders involved, can contribute to a better understanding of the degradation and the decision-making process of maintenance prioritisation leveraging data analytics, optimisation algorithms and visualisation capabilities enabled by ML, AI and 3-dimensional (3D) modelling and simulation techniques as part of realising a proactive maintenance approach (Waqar, 2024).

Within the construction industry, the application of DTs across different infrastructure assets have been researched across multiple levels, for example, at city and building level (Lu et al., 2020), bridge level (Ye et al., 2019) and tunnel level (Yu et al., 2020). All these studies focused on the adoption of DTs for an enhanced operation and management phase of the lifecycle of the underlying physical assets. However, limited amount of previous doctoral research thoroughly explored the potential enabling methods for a DT at the road level asset management, and hence this doctoral research project addresses this significant shortfall by investigating and evaluating different aspects that would enable a DT for modelling pavement performance as well as the road management overall.

1.2 Aim and Objectives

The aim of the research is to demonstrate the potential enabling methods for a predictive road DT (RDT), in the areas of pavement performance prediction and data collection. To achieve this, the research has the following key objectives:

1. Develop a DT-based decision-making support framework for road lifecycle which includes the key DT layers, (sub)components, their interactions and technologies.
2. Implement and assess the performance of an RDT-based predictive model in comparison to that of traditional ML models on road condition using historical data.

3. Investigate data collection frequency optimisation as part of an RDT leveraging real time sensor data.

4. Discuss the perspectives and provide recommendations for the use of an RDT.

1.3 Research Scope and Limitations

This research produces a DT-based decision-making support theoretical framework for road lifecycle described in Chapter 3 with five layers, and part of the framework is then applied to demonstrate the enabling methods and optimisation for a predictive DT for roads through two case studies. The other parts which were not directly implemented are not in the scope of this study and therefore are included as limitations of the research. The scope and limitations of this study can be described as follows:

- Only the methods for a predictive DT on pavement performance modelling is concretely evaluated through case studies using historical and real-time data. The maintenance prioritisation and decision-making process based on the predictive DT outputs will only be discussed.
- Only a limited discussion will be provided on the use of RDT on other stages of the road management life cycle such as design, construction and operation phases.
- The focus of the research is on the analysis and the predictability of a DT with the elements such as data and predictive modelling, rather than a full-scale of a DT such as the generation of digital 3D model visualisations.

1.4 Contribution of the Thesis

This study will contribute to current research as follows:

1. Provide a DT-based decision-making support theoretical framework for assessing the enabling methods for an DT in road lifecycle management across their design,

construction, operation and maintenance. Such a framework that integrates key components of DTs with RAM was not found elsewhere in the existing literature.

2. Advance the pavement performance modelling research area by moving the state of the art on this research topic further from a pure data driven approach such as ML modelling for pavement defects to combining physics with ML such as physics-enhanced ML to improve the model's stability, rationality and reliability.
3. Assess the physics-enhanced ML modelling performance compared to that of ML only as part of a predictive RDT.
4. Continuously monitor a pavement through an experiment with sensor instrumentation.
5. Optimise sensor data collection frequency as part of a predictive RDT consideration.

1.5 Structure of the Thesis

To achieve the objectives laid out above, this thesis is structured as follows:

1. The introduction in Chapter 1 describes the problem statement and contextual background of the research topic. The research aims and objectives are clearly defined. The scope, limitation, and contribution of the thesis is presented.
2. Chapter 2 reviews the concept, components, applications and benefits of DT technology as well as RAM. This chapter also provides a thorough review on existing pavement performance models which could be an essential part of a DT.
3. Chapter 3 describes the developed DT-based decision-making support theoretical framework for road lifecycle informed by the literature review outcome. In addition, the chapter presents a methodological approach on pavement performance modelling based on the RDT framework, especially combining ML, physics-based modelling as well as uncertainty quantification.

4. Chapter 4 presents the results of two case studies on the application of the DT-based decision-making support theoretical framework using both historical data for road rutting and roughness prediction, and real-time sensor data for the investigation and discussion on data collection frequency optimisation. The enabling methods for a predictive DT on the pavement performance modelling are clearly presented and discussed. Its use on the whole pavement lifecycle including pavement design, construction, operation and maintenance is also discussed.
5. Chapter 5 summarises the results from Chapter 4 and draws conclusions about the value of this research and presents the potential future works on the continuing developments and applications of the theoretical RDT framework.

2 LITERATURE REVIEW

This chapter provides a critical review of the literature base in relevance to the following areas: 1) Road pavement management; 2) Road deterioration mechanisms and defects; 3) Various types of pavement performance models; 4) DT and its impact on roads. Details are provided in Section 2.1 on road pavement management and Section 2.2 on DTs.

2.1 Road Pavement Management and Maintenance

2.1.1 Introduction

Road transport systems are seen by most countries as the essential foundations for general economic and social development, in facilitating trade both nationally and internationally. According to Robinson et al. (1998), a small improvement in the costs of operating and maintaining the physical infrastructure can result in huge economic benefits. On average, it has been calculated as three times savings return for investment on maintenance expenditure (Heggie, 1995). Burningham and Stankevich (2005) also emphasised the need for and the importance of road maintenance in every nation with regards to its position to sustain national transportation, due to the large amount of direct and indirect costs when it is not done appropriately. Therefore, given the value roads provide, a significant number of resources and budgets have been devoted to managing roads as an asset effectively, to achieve the optimal road pavement management.

Kulkarni and Miller (2003) described eight key elements of pavement management systems: 1. Functions or Scopes, 2. Data collection and management, 3. Pavement performance prediction, 4. Economic analysis, 5. Priority evaluation, 6. Optimization, 7. Institutional issues, 8. Information technology. Among the components, at the centre of RAM lies information collection and management. According to Hosseini and Smadi (2021), the performance predictive capability of road network deterioration has a direct impact on

subsequent activities within a management system such as maintenance prioritisation and economical optimisation as well as decision making.

Thus, this chapter provides a review of different pavement performance approaches and models, and various pavement condition data that are available for model development, as well as potential pavement management practices in the future. Section 2.1.2 introduces the pavement performance prediction and modelling concept, which is followed by available data sources for pavement deterioration modelling in Section 2.1.3. Section 2.1.4 describes road pavement deterioration mechanisms, factors impacting the deterioration and different pavement defects and their mechanisms. Section 2.1.5 provides a comprehensive summary on multiple types of pavement degradation models ranging from traditional types to the most recent developments in the pavement performance modelling area.

2.1.1.1 Pavement Management System

One key component of RAM is the pavement management system (PMS). PMS was introduced as a result of increasingly large road networks since the 1960s which require preservation, management of limited budgets, awareness of road user cost and more capacity to monitor the road condition, as well as the availability of information technology systems (OECD, 1987). Since then, the process of pavement management has been applied in national and local transportation agencies worldwide.

There have been continuous improvement needs, technical advancements and innovations over time in pavement management. More recent developments in this area have been reviewed by Pérez-Acebo et al. (2018), including technical implementation of automated data collection methods such as crack classification, multi-objective optimisation models with generic algorithms, pavement performance modelling and its adaptations. Parida et al. (2005) and Nodrat and Kang (2017) enhanced the PMS with the integration of a geographic

information system (GIS) to help with decision making support. This was done by leveraging its spatial analysis capabilities which enables PMS to provide features such as graphical display of the pavement condition, which could be of help in visualising decision-making and budget allocation. It is worth mentioning that the advantages GIS-integrated PMS has over the traditional PMS is not quantified concretely but it does provide a user-friendly graphic interface. Apart from the GIS system, given the development of digitalisation technologies, the use of Building Information Modelling (BIM) has been increasingly pervasive within the construction industry across the globe (Singh, 2019; Tang et al., 2019). There have also been attempts in combining PMSs with the BIM approach as discussed in Biancardo et al. (2023) and Oreto et al. (2023). The combined PMS-BIM acts as a tool to support an improvised pavement management and optimisation for maintenance with integration of road design and pavement analysis. It uses the BIM approach by modelling a digital road network, connecting and representing material models from external databases based on shared parameters, to apply selection ranking algorithms to evaluate the durability of the pavement materials to help prioritise the needs of maintenance. However, the authors did not mention how this approach would be more beneficial concretely compared to the existing prioritisation functions within PMS apart from visualisation, and therefore the quantified benefits BIM provides for PMS functionalities are still yet to be understood.

2.1.1.2 Maintenance Strategy Decision-Making Element within PMS

Data collection and pavement prediction models are essential elements of any PMS, which have a direct impact on maintenance scheme prioritisation and strategy decision-making (Haider et al., 2011), which is another important element within the PMS. This section succinctly presents this function. According to Lazic (2003), an effective road maintenance scheme helps to allocate the limited amount of funding to the roads that provides maximum

return on investment, and minimizes cost spent on preserving the road conditions on a long-term basis. Specifically, based on the research conducted by Hosseini and Smadi (2021) there is a strong correlation between the accuracy of pavement prediction models and their associated cost for maintenance and rehabilitation activities. Therefore, support tools or methods for prioritization and optimization of pavement maintenance activities decision-making take into consideration pavement distress level, budget constraints, and predicted pavement performance, and this has been another focused research area in pavement management and engineering (Peraka and Biligiri, 2020). Table 2.1 shows optimization techniques researched for pavement maintenance strategy decision-making.

Table 2.1. Optimization techniques for pavement maintenance decision-making

Author(s)	Optimization technique	Category	Advantage	Disadvantage
(Hajek and Phang, 1988)	Linear programming	Mathematical programming	Able to achieve optimum benefits	Require domain expert knowledge
(Ibraheem and Atia, 2016)	analytic hierarchy process model	Mathematical programming	Able to rank a list of decision options	Not mentioned
(Flintsch et al., 1996) (Fwa et al., 1998) (Taha and Hanna, 1995)	Artificial Neural Network & Genetic algorithms	AI	Enable automation of selection process of optimum maintenance strategy	Only focus on single treatment
(Yao et al., 2020)	Reinforcement learning	ML	ML-based trial and error to identify the best action to for the best outcome	New approach to the field, more research effort needed

Various optimization techniques have been employed and explored in the road sector to assist roadway agencies. These techniques include dynamic programming, linear and nonlinear programming, integer programming, optimal control theory, heuristic methods, Markovian and semi-Markovian methods (Berthelot, 2020). Hajek and Phang (1988) found that the

linear programming technique is helpful when allocating pavement investment strategies to achieve optimum technical benefits for the entire pavement network given a limited budget. However, the action plans as part of the strategies are produced manually by experienced regional staff whereas several other methods have been adopted to identify the right maintenance treatment types and timing. For example, Ibraheem and Atia (2016) have conducted a case study to understand the efficiency of the analytic hierarchy process model and concluded the benefit of this technique is its capability to rank choices of treatment types in the order of their effectiveness and hence decide the best treatment for pavement damage. Since the late 1980s, AI techniques have been developed to support the pavement maintenance decision making process and they include expert systems, artificial neural network (ANN), genetic algorithms, and hybrid approaches (Sundin and Braban-Ledoux, 2001). In particular, Flintsch et al. (1996) have used ANN to automate the selection process of roadway sections recommended for pavement preservation. Genetic algorithms were used by Fwa et al. (1998) to solve maintenance planning at the network level considering the maintenance time and type, and in addition, Taha and Hanna (1995) have combined both ANN and genetic algorithms for an optimum maintenance strategy selection for flexible pavements. However, these aforementioned maintenance strategies mostly focused on the effectiveness and cost of one single treatment and the mathematical programming methods make the solution space grow exponentially (Yao et al., 2020). Hence more recently, studies have been carried out by researchers to explore data analytics, and deep learning (Roberts et al., 2021) to support pavement maintenance decisions making and especially, one type of deep learning called reinforcement learning (RL), in order to better learn maintenance strategies (Yao et al., 2020). These authors concluded that the model could learn an appropriate strategy for 15-year maintenance planning. As it is one of the first deep RL

algorithms for pavement maintenance decision-making, further areas could be explored and included into the algorithm such as budget constraints and more advanced machine learning pavement performance prediction models for future research.

2.1.2 Pavement Performance Prediction and Modelling Concepts

Pavement performance prediction and modelling is an integral part of pavement maintenance management at both the network and project level (Anyala, 2011). At the network level, pavement performance modelling is used to provide long-term maintenance forecasts, development needs for the whole road network within different economical scenarios, and the prioritisation of road sections which need maintenance under budget constraints (Haas and Hudson, 1987; Robinson et al., 1998). Meanwhile at the project level, pavement performance predictions can be used to choose the best rehabilitation strategy amongst a list of alternatives (Kerali, 2001). Therefore, pavement performance modelling is arguably one of the most important elements to be able to manage the pavement efficiently and holistically. It also has been demonstrated that its accuracy has a direct impact on the maintenance decision-making process in terms of cost (Hosseini and Smadi, 2021).

In essence, the problem being discussed here is “when” and “what”, that is to identify when the best time is to maintain a road and what is the best maintenance type. To be able to find the solution, it is vital if the future condition of pavements could be predicted given the information collected about the road. Figure 2.1 illustrates the standard pavement deterioration curve and the concept of pavement performance prediction with maintenance decision making (Haas et al., 1994).

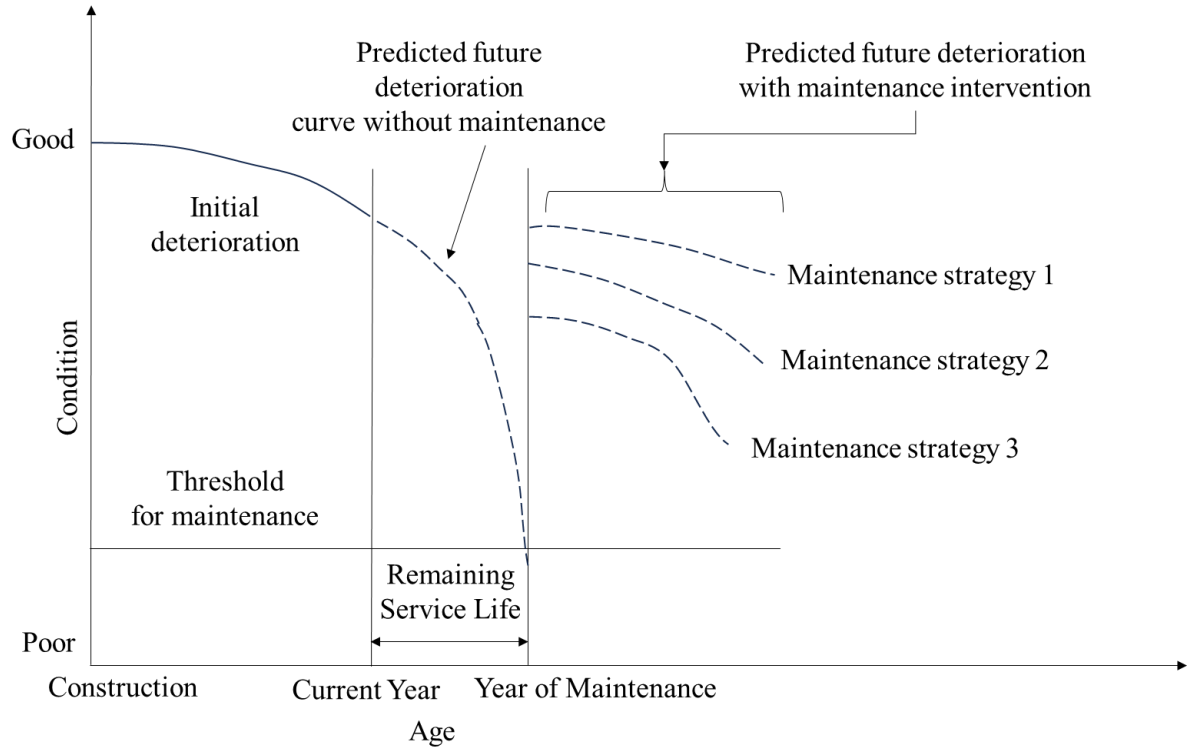


Figure 2.1. Concepts of pavement performance prediction (After Haas et al., 1994)

From Figure 2.1, the past deterioration trend which is determined based on existing inventory and collected condition data, is followed by a dotted curve showing the future predicted values made by pavement performance models within a pavement management decision-support tool. Upon the prediction, different maintenance options could be considered. The choice of a proper strategy for maintenance can lead to roads being kept in a good condition for a long period of time, ensuring the serviceability of the road networks (Robinson et al., 1998).

2.1.3 Data Used for Pavement Deterioration Modelling

Before any modelling work for pavement deterioration, it is necessary to collect relevant pavement data. They can be typically put into following categories: 1) Pavement inventory data (road geometrics, pavement sections, drainage and other amenities, if any); 2) Pavement structural data (number of layers, thickness of the layers); 3) Pavement material data

(material properties); 4) Pavement condition data (road defects from inspections, roughness, deflection); 5) Traffic volume data (annual average daily traffic, axle load distribution); 6) Environmental data (temperature, precipitation, and humidity) and 7) Historical maintenance records (Peraka and Biligiri, 2020). The following sub-sections describe different data sources and types in detail.

2.1.3.1 Data from Existing Databases

Existing pavement data providing information about the pavement inventory, structure, material and condition is accessible mostly from public databases such as open-source datasets created for research or commercial purposes, and transportation agencies' data management tools as well as the specialised databases (e.g., the US Long Term Pavement Performance (LTPP)). Such data is categorised as historical data throughout the years collected by various equipment and road pavement condition survey vehicles with a fixed frequency, e.g., annually or twice a year or every two years. Apart from data holding numerical values, multiple pavement distress image data is also collected using specialised vehicles equipped with high-resolution cameras and subsequently the data is stored in such traditional databases, providing annotated images with different types of pavement distresses like cracks, potholes, and fatigue cracks (Kheradmandi and Mehranfar, 2022). From a research point of view, many research effort have utilised the data that is publicly available in the US LTPP database to model the progression of defects (Abdelaziz et al. 2018; Fathi et al. 2019; Gong et al. 2018; Marcelino et al. 2019; Ziari et al. 2015, 2016). In addition to this database, researchers have also been able to use data stored in governmental or local transportation agencies' PMS across the globe such as the California Department of Transportation in the US (Shu et al., 2022), National Highways in the UK (Corker et al., 2023), Austroads in Australia (Martin and Choummanivong, 2016), State Highways in New

Zealand (Stevens et al., 2009), Road Administration Bureau of Shanghai (Wang et al., 2020) to understand road deterioration and model pavement condition.

2.1.3.2 Data from Sensors and Maps

More recently after entering the industry 4.0 era where a large amount of data is available due to the advancement in technologies such as Industrial Internet of Thing networks, Big Data, Robotics, Automation and AI, it has been suggested that the abundance of data from sensor networks can provide continuous information on the performance and behaviour of the physical infrastructure, which should be adopted in the pavement realm to ensure a more efficient and effective transportation system (Steyn, 2020). Pavement practitioners have started leveraging from multiple different sensors available to collect pavement condition data and to evaluate the potential benefits. For example, making use of mobile phones, Wang (2019) and Souza et al. (2018) demonstrated that the data generated from accelerometers in smartphone devices can be used to measure road roughness with an average error of less than ± 0.3 m/km, and predict the International Roughness Index (IRI) accurately using random forest (RF) algorithm with the accuracy of 97.3% in terms of Coefficient of Determination (R^2), thereby classifying the condition of roads.

In addition to data generated from smartphones, low-cost monitoring sensors installed on the pavement surface or within its structure are going to become another data source that helps road authorities to acquire real-world and real-time data, enabling the continuous monitoring of roads. This is aimed to achieve a better understanding of road structural health and essentially a superior pavement management, which has been demonstrated in field trials (Ye et al., 2024). For instance, the flexible pavement research at the Virginia Smart Road instrumented environmental sensors such as thermocouples to measure temperature, time domain reflectometry probes for moisture content in the base layer and resistivity probes, to

measure frost penetration as well as dynamic sensors to evaluate stress and strain across different layers under truck loading (Loulizi et al., 2001). Moreover, Karimi and Mallick (2023) provided a comprehensive state-of-the-art review on the instrumentation of flexible pavements where various sensors such as strain gauges, earth pressure cells, linear variable differential transformers and their applications in pavement have been presented. Similarly, a review of the application of fibre optics sensors in asphalt pavement monitoring systems worldwide has also been provided by Kara De Maeijer et al. (2019).

Additionally, imagery and point cloud are among the sources of pavement geometric data for creating digital presentation of the pavement. Geospatial point cloud data sets are generated by sensors such as 3D scanner, Light detection and ranging (LiDAR), and Photogrammetry software that can provide a detailed surface topography, which can also be integrated with existing GIS software (Inzerillo et al., 2018). The recent work from Marie d'Avigneau et al. (2025), introduced the CAMHighways - Cambridge Highways dataset, where a versatile and comprehensive dataset is provided using the mobile mapping data surveyed over 40 km of UK Highways. The dataset has been prepared and automated to produce segmented and classified point clouds, annotations, labelling and georeferenced for the development of geometrical road digital twins. Instead of data collected from the field, various forms of map data on the public online platforms can also be retrieved, such as aerial photography, high-resolution imagery, digital surface and elevation models as well as from different kinds of map downloading platforms and software. Take a recent work from Jiang et al. (2022) as an example, a systematic approach was proposed and tested to make a highway digital twin based on map data using Digimap (https://digimap.edina.ac.uk/) in the UK.

2.1.3.3 Data Collection Frequency

Pavement surface condition data plays a pivotal role in pavement management and performance modelling. Transportation agencies around the globe collect pavement condition data on a regular basis but not at a very frequent rate and at the same time it varies between different countries or states. Some authorities collect condition data only once a year, or twice in a year whereas others may only collect data once in two, or sometimes three years. The frequency of road condition data collection plays a very important role in pavement management and performance prediction. As reported by multiple studies (Haider et al., 2010; Xu et al., 2018; Hosseini and Smadi, 2021), the frequency of monitoring the road condition will have a significant impact on model performance prediction, cost planning and maintenance strategy planning. Furthermore, it has been recommended by Wang et al. (2020) that agencies should understand how the differences in data collection frequencies might impact their pavement performance models. With this knowledge, agencies can optimize the data collection plans. Therefore, they will be able to reduce the frequency without compromising the precision and accuracy of analysis.

With the fast development of IoT technology, its applications in road management have gained attention and momentum. These sensors enable a constant real-time monitoring of the pavement functional and structural health measured by a range parameters such as deflection, stress, strain, vibrations, pressure, as well as temperature and humidity. This in turn helps to analyse pavement behaviour, causes of pavement deformation and different road defects. However, instrumentation in pavement also poses multiple challenges. Firstly, such sensors are often integrated with data collection devices for large amount of data which has high energy demands and can hardly possess the collected data in real-time. Another challenge when applying IoT to road structural health monitoring is the redundancy in the large number

of real-time sensor data, thereby generating a substantial amount of unnecessary energy and cost (Ye et al., 2024). Therefore, there is a need to acquire this knowledge on the optimal frequency prior to the actual data collection.

2.1.3.4 Accuracy and Uncertainty

For transportation agencies, various pavement condition data is collected, and then used for model development that facilitates pavement management. However, there is inevitable level of inaccuracy and uncertainty associated with the collected data and the built models, which in turn would impact the decision-making process. This sub-section reviews the sources of these uncertainties and the techniques to tackle them. For pavement data, they can be largely categorised into two groups: 1) Pavement surface distress such as cracking and patching, which is measured using images, videos or manual observations; 2) Defects measured using vehicle sensors such as rutting, roughness and faulting. More recently, real-time data from embedded sensors has also been collected in laboratory and field applications. Regardless of the data collected using condition inspection survey vehicles or instrumented sensors, it is unavoidable that there is error or uncertainty in the accuracy of the measurement data obtained. This has been seen as a general issue in pavement data collection (Barbedo, 2019). For example, according to the US federal highway administration guideline (Visintine et al., 2018), a scenario was illustrated where the pavement rutting condition may not deteriorate at all or even improve itself based on the data collected in the absence of any maintenance or rehabilitation because of measurement error. This may not reflect the reality of the pavement deterioration, as demonstrated in Figure 2.2.

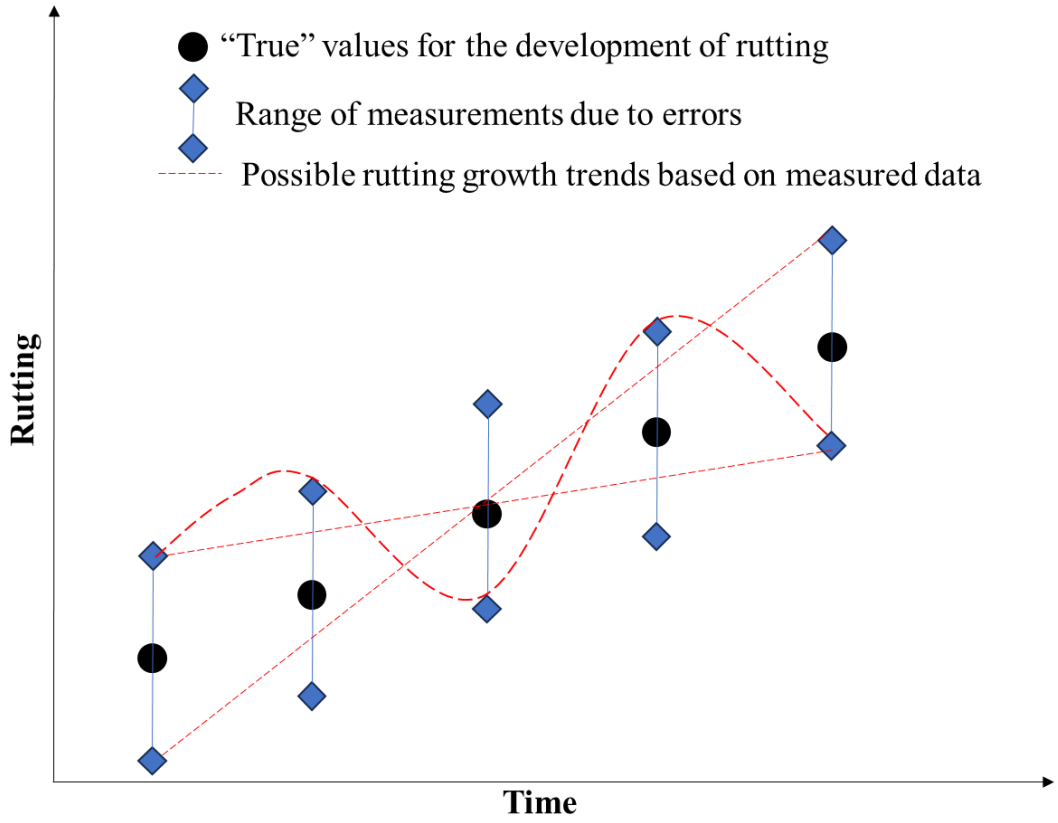


Figure 2.2. Potential measurement of rutting growth over time (After Visintine et al., 2018)

Therefore, the inconsistency between the predicted pavement performance and the observed field performance is a function of the prediction model and the inherent uncertainty in the measured pavement distresses. There are multiple different sources of uncertainty in pavement data (Wu et al., 2023). It can often be due to factors such as inherent variations in the physical quantity or material properties (e.g., aggregation sizes, inconsistent binder properties and variations in subgrade soil), and inaccurate measurements due to human error or sensor reading errors from improper equipment, as well as the pavement spatial variability such as sampling and data collection frequency and the changing traffic as well as ambient climatic environment (Noshadravan et al., 2013). This aspect of uncertainty stemming from data has also been highlighted by several studies (Amin & Amador-Jiménez, 2017; Gogoi et

al., 2020; Luo et al., 2022; McNeil & Humplick, 1991) and should be accounted for when it comes to pavement performance modelling.

Data pre-processing is a key step to address the inherent data quality issues, and the uncertainties associated with the collected data. Several key pre-processing techniques for traditional pavement condition data include: 1) dynamic segmentation for spatial variability; 2) outlier removal for measurement errors or unreasonable data; 3) smoothing / moving average for noises in the data and temporal variability; and 4) Interpolation and bootstrapping for missing data (Kargah-Ostadi et al., 2019). Different techniques are also applied in addressing similar issues in big data such as data from sensors. Khoei & Singh (2024) summarised these methods into two categories: data transformation and data cleaning with techniques such as normalisation aggregation, discretisation, imputation, outlier detection and removal that could help streamline the analysis of large and complex datasets (Hancock et al., 2024).

Another critical and influential source of uncertainty for pavement deterioration modelling process is the inherent errors to the model itself, due to the complexity and large-scale sizes for pavement structures (Song et al., 2020). The errors may arise from the simplifications made in representing the real-world processes, and the inaccurate assumptions made during model's development on the underlying system. In addition, missing key factors or the unknown functional relationships between variables may also result in model errors (Refsgaard et al., 2006; Simmonds et al., 2022). Moreover, it can be attributed to the limitations in estimating model parameters and the corresponding values (Zhang et al., 2024).

There are multiple different strategies to mitigate model inherent errors. For numerical models, process of validation and calibration, mesh refinement that improves discretisation, and the use of more complex numerical representation are among the common approaches to

reduce the errors in the predictions (Freitas, 2002; Eça and Hoekstra, 2014). For empirical, statistical and machine learning models, the most frequently used techniques include cross-validation, regularisation, model parameter sensitivity analysis as well as uncertainty quantification such as producing confidence intervals to reflect the level of uncertainty (De Muth, 2019; Berber and Srećković, 2024).

For pavement management decision making, it is of paramount importance to incorporate uncertainty into the process (Gregory et al., 2017). The uncertainties derived from data and model errors can be dealt by adopting several methods, especially by the validation of data quality and the use of probabilistic pavement performance models (Karanam et al., 2023). For example, Chang et al. (2024) introduced a systematic statistical method with quality control procedures to analyse the reliability of pavement condition data from field surveys to enhance pavement management decision-making. In addition, different probabilistic pavement deterioration models can be used to help support decision-making, such as Bayesian network, Markov chain and Monte Carlo. Bayesian probabilistic models produce predicted probabilities and multiple deterioration scenarios under different factors influencing road degradation, based on which decision-makers can choose the most appropriate maintenance strategy (Cui and Wang, 2024; Philip and AlJassmi, 2024). Markov chain models also provide probabilities for future road condition predictions, allowing decision-makers to understand the risk associated with the maintenance schemes. However, unlike Bayesian models, it only takes the current condition state into consideration to predict the next state through transition matrix (Sati et al., 2020; Seites-Rundlett et al., 2022; Wasiq and Golroo, 2024). Another method to deal with uncertainty in decision-making process is to use Monte Carlo simulations. It incorporates probability distributions to represent the inherent variability in the input variables that impact road deterioration. Then the most cost-effective

maintenance type and timing can be identified (Guo et al., 2020; Heidari et al., 2020; Vagdatli & Petroutsatou, 2022). More details on pavement probabilistic predictive models can be found in Section 2.1.5.2.

2.1.4 Mechanism of Pavement Deterioration

2.1.4.1 Factors Affecting Pavement Deterioration

As soon as a pavement section has been constructed and opened for traffic flow, the process of deterioration starts (Jafari Ahangari, 2014). There are multiple different factors that could have an impact on this. According to Haas et al. (2001), the road pavement's performance deterioration is influenced by the interactions between construction quality, traffic volume, road structure, surrounding environment, vehicle speeds, load axle configuration, types of tyres and pressure as well as the maintenance policy.

Similarly, in a study investigating the pavement structural behaviour, Salour & Erlingsson (2013) also concluded that the factors that are affecting pavement condition can be pavement age, traffic, environment, pavement thickness, materials, as well as the strength and the mechanical properties of the pavement. Knowledge about the factors affecting pavement performance would help in understanding the mechanism and the prediction of the pavement future state. The next sub-sections summarise the potential defects comprising deterioration.

2.1.4.2 Modes of Pavement Deterioration

As the pavement deteriorates, various forms of distresses would occur in asphalt pavements, each due to a unique reason, such as poor mix design, construction, or environmental conditions, traffic and/or a combination of each. Mallick and El-Korchi (2013) have summarised a list of common distresses in asphalt pavements.

- Bleeding: Shiny asphalt surface on the road caused by upward asphalt movement.

- Block cracking: Interconnected cracks dividing pavement into rectangular pieces.
- Delamination: The separation of asphalt layers due to loosened bond.
- Edge crack: Long cracks along the edges of asphalt.
- Fatigue crack: Interconnected cracks caused by fatigue failure of the asphalt surface layer due to repeated traffic loading.
- Longitudinal crack: cracking occurring in parallel to the centreline of the pavement.
- Polished aggregate: Worn down aggregate particles and binder due to traffic over time.
- Pothole: Bowl-shaped depressions in the pavement surface.
- Ravelling: Disintegration of an asphalt road surface due to the dislodgment of aggregate materials (gravel, sand, and crushed stone).
- Reflective cracking: Cracks in a flexible overlay layer that match existing cracks in the underlying pavement.
- Slippage crack: Crescent- or half-moon-shaped cracks having two ends pointed away from the direction of traffic.
- Thermal cracking: Cracks caused by the thermal stress within the pavement material due to fluctuating temperatures.
- Rutting: A longitudinal surface deformation usually on the wheel path.

A typical pattern of deterioration in asphalt pavements is rutting, which is also the most basic form of pavement distress (Thom, 2024). It initiates and progresses rapidly during the first few years since initiation, but then slows down to a lower rate (Huang, 2004). Rutting is of great significance for road management. A study of the mechanisms of deterioration on trunk roads in Europe compared several modes of pavement distress as shown in Figure 2.3 and

concluded that rutting occurring in the asphalt layer of pavements is the most important mode (Atkinson et al., 2006).

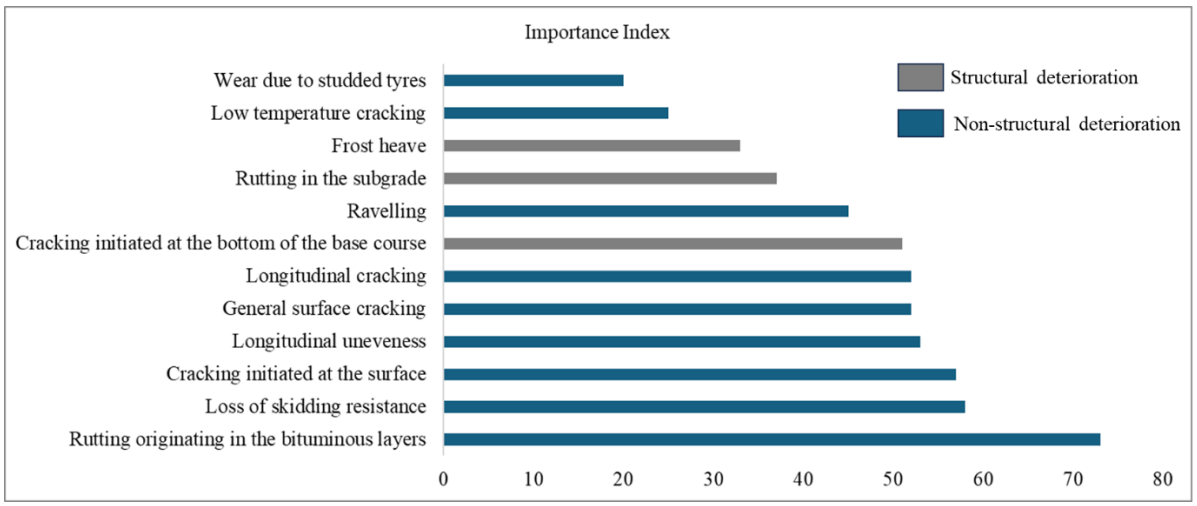


Figure 2.3. Importance of perceived modes of deterioration in fully flexible pavements (After Atkinson et al., 2006)

In addition, the fourth important mode in the same list, longitudinal unevenness or roughness, has also been regarded as a key indicator for road condition to trigger road maintenance. Reported as IRI, it is used to measure ride quality, comfort level for road users and users operating costs (Sayers et al., 1986). Therefore, this study focuses on these two pavement deterioration modes: rutting and roughness.

2.1.4.3 Mechanism of Road Rutting Formation

Pavement deterioration can be put into two categories: 1) surface deterioration 2) structural deterioration, both of which can cause the rutting defect phenomenon to appear (Mehdi et al., 2022). Surface rutting is described as a result of excessive stress induced by traffic load which is beyond the shear strength of the material (Paterson, 1987). The main scenarios that would cause the development and growth of surface rutting have been stated by TRL (1993) and are listed below:

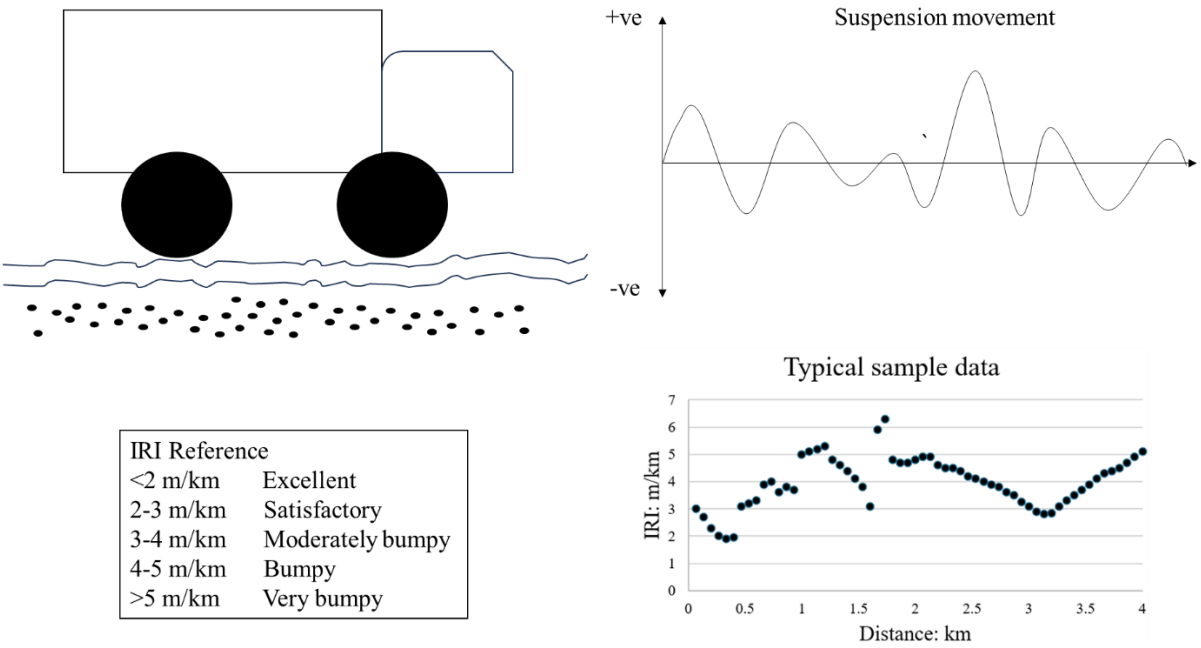
- Very heavy axle loads;

- High maximum temperatures;
- Channelised traffic;
- Stopping or slow-moving traffic.

Structural rutting, on the other hand, mostly stems from the permanent vertical deformation of the underlying pavement structure under repeated traffic loads. For example, it could be a reflection of the permanent deformation within the subgrade layer (Albayati, 2023).

2.1.4.4 Mechanism of Road Roughness Formation

Traditionally, roughness is used as a metric for road pavement quality evaluation (Tamagusko and Ferreira, 2023) and therefore used by road agencies to identify which highway networks require maintenance and rehabilitation and the allocation of budgets (Damirchilo et al., 2021). Roughness is measured by the IRI which is one of the few pavement condition measures that have international acceptance. It is defined by the amplitude of motion of a standard vehicle suspension system as it travels along the road, measured in metres of the suspension system movement per kilometre of travel cumulatively (m/km or mm/m) (Thom, 2024). Bump integrators measure the IRI when driving on the highways with average traffic speed. The idea is illustrated in Figure 2.4 (Thom, 2024).



IRI Reference	
<2 m/km	Excellent
2-3 m/km	Satisfactory
3-4 m/km	Moderately bumpy
4-5 m/km	Bumpy
>5 m/km	Very bumpy

Figure 2.4. Bump integrators and the IRI (After Thom, 2024)

Roughness can be caused by either longitudinal or transverse distortions in the roadway, where the former could be a result of consolidation of the foundation material of the pavement, and the latter could be a result of rutting or settlement within the road (Mallick and El-Korchi, 2008). A summary of the factors affecting roughness progression has been presented by Hunt and Bunker (2002) and listed as follows:

- Construction quality: Strength/Moduli of pavement, dependent on type of soil, rock, gravel, asphalt and compaction.
- Environmental impact: Precipitation and temperature variances causing the drying, evaporation and oxidation of bitumen.
- Pavement ageing: strength of pavement across all layers.
- Traffic loading: a mixture of vehicles loads.
- Pavement drainage conditions: Permeability varies by type of cross sections and long sections.

The mechanisms of rutting and roughness discussed in this section help to understand and identify key variables which are of importance for the prediction model development. The next section reviews the state-of-the-art on the different types of existing deterioration models.

2.1.5 Pavement Deterioration Models

Pavement deterioration models are important in predicting the degradation of pavement and the progression of different defects. The models reviewed in this section are discussed under the following headings: Deterministic models; Probabilistic models; and ML models.

Within road pavement management, one of the most important components is the prediction model for pavement performance deterioration, as it aims to forecast the future conditions of the pavement accurately. It serves as a fundamental input to design and plan maintenance activities, to prioritise scheme selections and to complete associated treatments to distresses, which supports the final decision-making. Figure 2.5 describes the development of these models and their relevant characteristics.

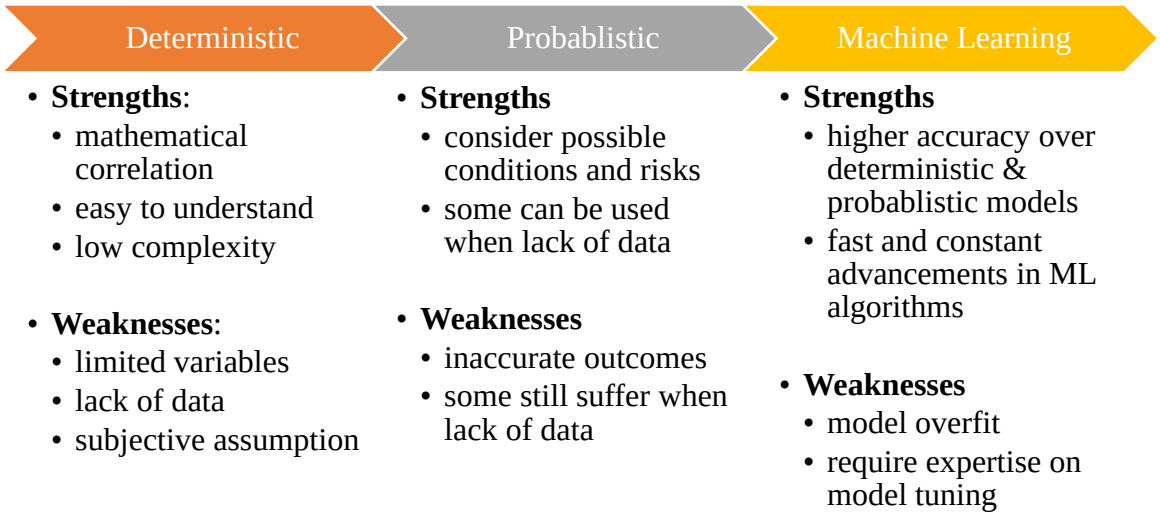


Figure 2.5. Pavement performance prediction models developments and characteristics

Deterministic models are composed of a series of mathematical functions, and probabilistic models predict a range of probability of a particular condition based on the distributions of probability in the future, whereas machine learning models use AI techniques to identify correlations between variables and predict the pavement future condition based on a learning process using historical data (Anyala et al., 2014; Sanabria et al., 2017; Wang et al., 2017).

2.1.5.1 Deterministic Models

The deterministic models typically encompass four key components: primary response, structural performance, functional performance and damage models (George et al., 1989). The preceding models can be categorised into mechanistic models (stem from physical properties and interactions), empirical models (developed from regression analysis, which usually have specific formulas) and mechanistic-empirical models. Theoretically, the mechanistic models are drawn based on the relationship between response parameters such as stress, strain, and deflection (Li et al., 1996). The mechanistic-empirical models draw the relationship between road defects and the loading of the traffic whereas the empirical models provide the correlations between a performance indicator (e.g., IRI) and a series of predicting parameters such as pavement number of layers and thickness, structural number, material properties, pavement age and traffic loading (Shohel and Amin, 2015). A general function of a deterministic pavement performance model can be expressed by Eq. 2-1 proposed by Li et al. (1996).

$$PCS_t = F\ (P_0, ESALs_t, H_e \text{ or } SN, M_R, C, W, I) \qquad \text{Eq. 2-1}$$

Where the PCS_t represents the generalised pavement condition state at a given year t, P_0 is the initial pavement condition state, $ESALs_t$ means the equivalent single axle loads (ESALs) accumulated at age t, H_e is the total granular pavement structure thickness, SN is the structural number index of the total pavement, also known as the pavement strength, M_R is

the subgrade soil resilient modulus, C means construction effects to be considered, W accounts for a set of climatic or environmental effects, and I is the interactions effects. Based on this, the American Association State Highway and Transportation Officials developed the equation for the present serviceability index (PSI) of the flexible pavement where 18-kip ESALs, material properties, drainage and environmental conditions are considered as main factors, as presented in Eq. 2-2 (Abaza et al., 2001).

$$Log_{10}(ESAL_s) = Z_R \times S_0 + 9.36 \times \log_{10}(SN + 1) - 0.2 + \frac{\log 10\left[\frac{\Delta PSI}{4.2-1.5}\right]}{0.4+\frac{1094}{(SN+1)^{5.19}}} + 2.32 \times \log_{10}(M_R) - 8.27 \quad \text{Eq. 2-2}$$

Where ΔPSI is the difference between the initial design serviceability index and the serviceability index at the year t, Z_R is the standard normal deviate, and S_0 is the combined standard error of the traffic and performance prediction.

Similarly, developed a mechanistic roughness model relating the roughness with the number of load repetitions, axle load, and the thickness of asphalt layer. The statistical relationship is depicted in Eq. 2-3 where IRI_0 is the initial roughness which is the most significant factor that affects roughness in road deterioration, and other factors are axle load expressed as P, asphalt thickness as T, and the number of load repetitions as ESALs.

$$IRI = -1.415 + 2.923 \times \sqrt{IRI_0} + 0.00129 \times \sqrt{ESALs} + 0.000113 \times T - 5.485 \times 10^{-10} \times P^4 - 10^{-3} \times T \times \sqrt{ESALs} + 5.777 \times 10^{-12} \times P^4 \times \sqrt{ESALs} \quad \text{Eq. 2-3}$$

At present, deterministic models are largely implemented by regional or local PMSs in most transportation agencies across the globe (Hicks and Groeger, 2001; Ferreira et al., 2010). For example, Highway Development and Management (HDM) model four, they provide a mathematical correlation between the condition and the variables that impact pavement deterioration, which is easy to understand and has low level of complexity to use (Karimzadeh, 2020). However, these models only cover a limited number of variables that

contribute to the pavement degradation because of lack of available data and deterioration knowledge, which limits the models' applicability and flexibility. In addition, the extensive use of expert-based subjective assessment with relevant assumptions as well as simplification cause challenges to the accuracies of the models in general (Karimzadeh, 2020). Furthermore, deterministic models often would require calibration and structural data, which puts a huge challenge on its development and application (Anyala et al., 2014).

2.1.5.2 Probabilistic Models

Given the stochastic nature of the pavement deterioration process, its nonlinear behaviour, as well as the influence of unexplained variables, it was suggested that more complex models are required to capture this deterioration process (Justo-Silva et al., 2021). Therefore, probabilistic models were developed by taking into consideration a range of possible conditions leveraging risks and its associated probability. They have been adopted prevalently in the US and Europe. Probabilistic models are purely empirical and are defined by transition probability matrices with probabilities of transition between quality states of the pavement with or without application of maintenance and rehabilitation actions.

One of the other common probabilistic modelling approaches is Bayesian methodology or Bayesian models, which works differently to Markov chain. Specifically, it uses a probability distribution for prediction by considering a variety of factors. As a result, these models also are impacted by insufficient data in most cases, but they use probability distributions by simplification applied to mitigate these issues caused by the lack of information. As a result, they could help forecasting the pavement future condition. Nonetheless, these distribution models might not be able to provide an exact fit on the actual performance observations and consequently lead to inaccurate results (Karimzadeh, 2020). Markov chain and Monte Carlo methods have been recognised to facilitate this analysis (Lunn et al., 2000).

Markov chain is a typical representative of the probabilistic models. It can predict the next state of the pavement based on the knowledge of the previous or current state of pavement (George et al., 1989). The main challenge for Markov chain process is the development of the transition probability matrices. Several pavement degradation functions by means of Markov process modelling were developed for PMS of Arizona Department of Transportation (Eq. 2-4) using observed historical pavement performance data and Waterloo (Ontario) regional road network (Eq. 2-5) based on pavement age.

$P_{ij}^{n} = \sum_{k=0}^{M} P_{kj}^{1} P_{kj}^{(n-1)} \forall n \leq v$ and $P_{ij}^{n} = \sum_{i=0}^{M} \sum_{k=0}^{M} (P_{ik}^{(v)} \times P_{kl}^{(1)a}) P_{lj}^{(n-v-1)} \forall n > v$ Eq. 2-4

Where P_{ij}^{n} is the n-step transition probability from the condition state i to j for the entire period (N), M + 1 is the total number of pavement condition states, v is the period within which the rehabilitation is applied; $P_{ik}^{(v)}$ is the v-step probability of transition from condition i to k under the routine maintenance; $P_{kl}^{(1)a}$is the one-step transition probability from the condition k to l at period v; and $P_{lj}^{(n-v-1)}$ is the (n-v-1) step transition probability from the condition l to j under the same routine maintenance (Wang et al., 1994).

$V(n) = V(0) \times M^{n}$ Eq. 2-5

Where V(n) is the predicted condition state matrix at year n, V (0) represents the initial condition state matrix at year 0, and M is the 1-step transition probability matrix (Karan, 1978; Wang et al., 1994).

One of its benefits is that this method does not consider the actual factors that contribute to road deterioration, and that it can be useful especially when there is not enough data for all variables and factors (Jin and Mukherjee, 2014). However, one of the assumptions of such models is to assume all transition probabilities are constant which would cause inaccurate outcomes. Even with some variants of these models which do use flexible transition probabilities, the imprecision is still found to be high and therefore inaccurate results are

produced (Chen and Mastin, 2016). When Markov chain models are combined with Monte Carlo simulations, the model could reflect the stochastic transition of the pavement condition over time and such models can be developed without the need of a large historical database. Probabilistic pavement deterioration model's parameters can also be derived using Monte Carlo simulations for estimating the uncertainty band of the parameters and the model predictions (Karanam et al., 2023).

2.1.5.3 Machine Learning Models

Starting with a basic understanding, unlike the mathematical or statistical models introduced in previous sections, ML is a branch of AI, which is driven by data and learns patterns directly from the data without relying on explicit, pre-defined assumptions. ML can be categorised into classification or regression problems. Classification algorithms are used to classify or predict the classification of certain values such as true or false, spam or not spam etc. However, regression algorithms are used to predict continuous values such as condition index, age, and salaries. ML algorithms are used to fit the data to understand the relationship between input(s) and outputs. For example, given a very simple dataset $(x, y) = \{(x_1, y_1), (x_2, y_2), (x_3, y_3) .. (x_n, y_n)\}$, where (x_i, y_i) is a pair of data input and output. It is defined that for point $x_{i,}$ there is a corresponding output value $y_{i.}$ From a mathematical point of view, a model needs to be built to identify the relationship; this is where a ML algorithm can be applied (Badillo et al., 2020).

In more recent years, according to Morales et al. (2017), ML has become a popular approach adopted by researchers and practitioners in the prediction of future pavement performance. Based on the review conducted by Sundin and Braban-Ledoux (2001), as well as the proposal from Efe and Shokouhian (2020), ML techniques such as ANNs are able to serve as an alternative way to perform prediction modelling for pavement management systems. Multiple

machine learning algorithms and techniques have been researched and implemented for pavement performance prediction. For example, Hamdi et al. (2017) used ANN to predict surface distress index with a high correlation factor (R^2 = 99.6%) using data from an integrated pavement management system following a data-driven approach. In addition, ANN have been proven by various researchers regarding its prediction accuracy over previous traditional model types. For instance, Sanabria et al. (2017) conducted a comparative analysis between the prediction performance on distress rate using ANN and ordered-probit which is one of probabilistic models utilising the same traffic data, and the study suggested that ANN produces higher prediction capacities over ordered-probit. Similar studies have been carried out by Yang et al. (2003) and Saghafi et al. (2009) who concluded that the ANN models resulted in more prediction accuracy in comparison to traditional regression models. As the field of computer science and machine learning continues to develop, multiple improved and advanced versions of ANN as well as other types of NN have also been produced and subsequently researched for investigating their suitability in pavement performance prediction. One significant factor that impacts the performance of ANNs is the implemented backpropagation neural network algorithm (Leung and Haykin, 1991), and several research studies have been conducted with a focus on the use of back propagation neural network to establish prediction modelling. According to the study done by Amin and Amador-Jiménez (2017), Back Propagation Neural Networks with a generalised delta rule learning algorithm reduce the measurement errors experienced in most deterministic and stochastic models. The same algorithm has also been proven to produce highly accurate prediction of rutting progression in bituminous pavements and to define the error percentage contributed by input factors (Ajakaiye and Amin, 2020). More recently, given that pavement maintenance data is time-series based, an increasing amount of research has been focused on the use of another

type of NN called Recurrent Neural Network (RNN) which is essentially ANN with recurrent connections, and such a network is capable of modelling sequential time to learn features and long-term dependencies for sequence recognition and prediction (Salehinejad et al., 2017). Amirhossein Hosseini (2020) and Choi and Do (2020) also explored the usage of RNN to predict cracking, rut depth and the IRI based on 10-year historical data from the Korean National Highway PMS, taking into consideration data on pavement type, traffic, and environmental factors, as well as hyperparameter configurations. In addition to NNs, methods such as support vector machine (SVM) and ensemble algorithms such as RF and gradient boosting regressor as well as k-nearest neighbours (KNN) have also been applied in predicting pavement performance and achieving promising results (Marcelino et al., 2021; Shtayat et al., 2022; Chen, 2023). Despite the achievements of these studies adopting various ML techniques in predicting road defects and performance, there is a common limitation and problem faced by ML which is explained in the following sub-sections.

2.1.5.4 ML Limitations on Rutting Prediction

As for pavement performance modelling, including rutting and IRI deterioration modelling, remarkable developments in the application of different types of ML algorithms, have been observed such as NNs, Decision Trees, SVM, combinations of them and other advanced ML algorithms. However, despite using these encouraging ML techniques, the same drawback applies when using "black-box" ML models that are agnostic to the existing underlying physics. This issue has been acknowledged by researchers in the field, e.g., (Deng et al., 2024). For instance, in a study conducted by Alnaqbi et al. (2023), the authors collected 1584 records from the US LTPP database and compared the ML modelling performances across ANN, SVM, decision trees and Gaussian Process Regression (GPR) with the results showing GPR achieving the highest accuracy (R^2 of 0.989). Despite the promising result, the authors

also acknowledged the need for combining the ML approach with domain specific knowledge to yield more interpretable predictions. Similar suggestions have also been made by Haddad et al. (2022), in which a fully connected three-layered feedforward deep neural network (128-32-8) was developed and obtained R^2 of 0.82. However, the authors admitted the black-box nature without any direct explanation on how NNs produced a certain output, indicating the need for the model's interpretability and the integration of physical domain knowledge. Some progress has been made in this regard in pavement modelling. For example, the SHapley Additive exPlanations (SHAP) approach for the interpretation on models' rutting predictions has been adopted by Yao et al. (2021) and Guo et al. (2022) with Bayesian Neural Network and Gradient Boosting Decision Tree. While this approach has led to further enhancing of the ML model's capacity compared to traditional ML techniques as well as the ability to understand the model's prediction output on pavement performance, the SHAP value is a model-agnostic tool to explain the individual prediction output which only happens after the model development rather than considering physical characteristics (Chen et al., 2023). This study therefore addresses the gap regarding the integration of domain knowledge based on physics into the ML modelling process for pavement rutting prediction.

2.1.5.5 ML Limitations on IRI Prediction

Similarly, various attempts have been made in recent years to predict IRI applying different types of ML algorithms with promising results. For example, Ziari et al. (2015) used an ANN to predict IRI in the short and long term, with an average prediction accuracy, R^2, above 90%, with various network architectures. Despite the high accuracy, the ANN model was largely dependent on and optimised by the homogeneous data used, which is not the case in practice. Similarly, the findings of Kargah-Ostadi et al. (2010) demonstrated that the successful forecasting capacity of ANN with R^2 of 95.8% on the test data set makes it helpful in

network-level pavement management system decision making but it had the same data limitation where the data used for modelling was homogeneous. Several researchers have performed comparative studies revealing the superiority of ANN model performance on the same datasets from either the commonly used US LTPP database or unique state local agency databases (e.g., low volume road pavement sections in India) over traditional methods such as linear regression models and the HDM-4 model (DT Thube, 2012; Abdelaziz et al., 2018). Apart from ANN, the RF algorithm has also been suggested as a powerful ML algorithm resulting in relatively high long-term IRI prediction accuracy on test data (e.g., approx. R^2 98% for 5-year prediction, and R^2 93% for 10-year prediction) while dealing well with both overfitting issues and generalisation ability (Gong et al., 2018; Bashar and Torres-Machi, 2021; Marcelino et al., 2021). However, the lack of consideration of the physical deterioration mechanism over the years and the fact that the multi-year predictions were only based on data, reduce the reliability and accountability of the model. In addition, SVM is another well-known method that has been widely used for predicting the IRI. For instance, Ziari et al. (2016) have formed and analysed five SVM algorithm kernel types and identified Pearson VII Universal kernel being the one producing the most accurate results with R^2 over 92%. In another study using SVM regression for IRI forecast with a different data source (experimental pavement roughness data collected annually for seven years for a high-volume motorway), radial basis function kernel was found to yield the highest prediction performance with R^2 over 90% in training data and Mean Absolute Error (MAE) less than 8% in the testing data (Georgiou et al., 2018). These research outputs denote the performance variability of ML algorithms which is fundamentally bound by the characteristics of the data collected itself. More recent efforts have also been made to investigate other advanced and novel ML algorithms indicating even further improvement in prediction such as gradient

boosting or extreme gradient boosting (Damirchilo et al., 2021), TrAdaboosting (Marcelino et al., 2019) as well as CatBooster (Bral et al., 2024). For example, Bral et al. (2024) presented the superior performance of CatBooster over both RF and ANN by 119% and 129% for the training data, 5.2% and 3.4% for the test data. However, the training data having over 99% R^2 accuracy and the fact that it was significantly higher than that of test data seems to indicate an overfitting issue during the development of the model.

According to Karpatne et al. (2017) and Deng et al. (2024), the common issues of black box ML models can be summarised as follows:

1. Potential imbalance between the utilized data and the structure of the model, causing overfitting and a lack of model interpretability.
2. Heavy reliance on the quality and quantity of the utilized data, leading to unreliable predictions with input variables with different data distributions, reduced stability.
3. Tendency to produce scientifically inconsistent results that are not aligned with existing scientific theories and understanding as well as their inability to provide a mechanistic understanding of discovered patterns and relationship from data. This shows the need to introduce scientific consistency as an essential component for learning generalisable ML models in pavement performance prediction problems.

2.1.5.6 Integrate Physics with ML

In recent years, a new paradigm in the AI research area, integrating the aspect of existing known physics with ML have demonstrated potential capacity in various disciplines to account for some of the common ML issues and limitations such as uncertainties and imprecision (Willard et al., 2022). This fusion has been referred to using the term “physics-enhanced machine learning (PEML)” (Cicirello, 2024), which is adopted herein the thesis. This is achieved by denoting that prior physics knowledge is embedded, in some form, to the

learner, and guiding the ML or neural network training process through the incorporation of existing first principles or laws of physics before producing an output (He et al., 2020; Robinson et al., 2022). Therefore, it could generate a more consistent result and thereby improving model's stability and generalisability. This technique has already been applied across diverse disciplines such as climate science (Faghmous and Kumar, 2014), earth systems (Reichstein et al., 2019), engineering and environmental systems (Raymond and Camarillo, 2021; Daw et al., 2022), and structural engineering (Gu et al., 2022). Within structural mechanics, Haywood-Alexander et al. (2024) has focused on the application of PEML in this domain and proposed a two-dimensional spectrum contextualising the different model types of combinations driven by both the amount of data available, and the level of physics constraints that are applied.

Willard et al. (2020) and Haywood-Alexander et al. (2024) conducted detailed surveys on the overview of multiple PEML schemes as well as enabling methods. Based on a combination of the reliance on the physics model prescription and physics embedding methods, PEML techniques can be categorised as 1) physics-guided ML techniques; 2) physics-informed ML methods; and 3) physics-encoded ML schemes. Technique 1) can be achieved by hybrid physics-ML modelling and residual modelling methods; Technique 2) could be realised by using dictionary and physics-informed loss function methods whereas technique 3) can be approached using constrained Gaussian processes. physics-encoded initialisation and physics-encoded design of architecture. Figure 2.6 provides a diagram summarising the PEML schemes, techniques and implementation methods. The following subsections also review these in detail.

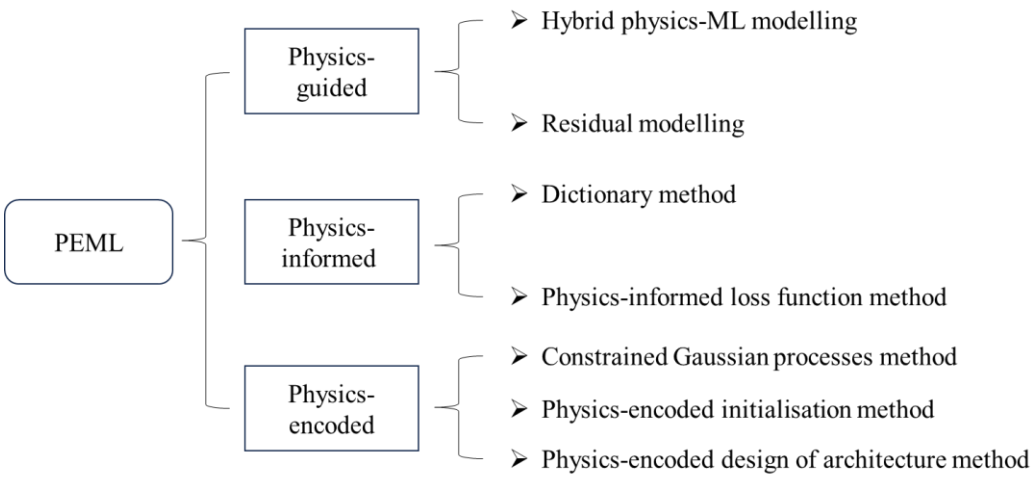


Figure 2.6. PEML Techniques and Methods

Physics-guided ML (PGML) techniques

This is the technique where the physics model prescriptions are embedded as proposed solutions, and act in parallel to the data-driven learner in the full PEML model architecture. It steers the ML learner by the high degree of strictness in the prescribing physical models. Two approaches have been used to implement them.

A. Hybrid physics-ML modelling

One straightforward method to ingest physics with ML is fusing the output of a physics-based model as an extra input to an ML model. For example, Daw et al. (2022) presented a framework for lake temperature modelling using PGML where the outputs from the physics-based model were used as an additional input, reducing the Root Mean Squared Error (RMSE) from 1.18 to 0.73 °C. Duran et al. (2022) used a similar approach where the physics-based simulation was adopted as an extra input for ANN to predict the mechanical properties of high carbon pearlitic steel and steel connection stiffness, which yielded a 38.5% accuracy improvement in RMSE. This method can also be seen as creating extra data to enlarge the training set, especially useful

in a controlled environment. Data generated by physical simulations has been also used as synthetic data for learning tasks such as autonomous driving (Dosovitskiy et al., 2017) and adversarial test generation (Tuncali et al., 2018).

B. Residual modelling

This approach differs from the previous one in that both ML and physics-based models operate simultaneously instead of one after another. Residual modelling is one of the most common scientific approaches to address the imperfections of physics-based models, where ML learns to predict the errors or residuals made by a physics-based model. This then can be used to correct the physical models' predictions. One key area where this approach has been applied is in reduced order models of engineering systems. For example, neural networks that were used to model the error in models due to reduction, demonstrated sharp error reduction when applied to known differential equations (San and Maulik, 2017, 2018) as well as in prediction of extreme weather events compared to observational data (Wan et al., 2018).

Physics-informed ML (PIML) techniques

This technique corresponds to a heavier reliance on data, but at the same time, retains still a moderate level of dependence on the physics. Physics information is embedded as prior information from which a loss function is constructed, which prompts the learning process. Two such approaches are reviewed and discussed.

A. Dictionary method

This method selects a suitably sparse representation of the physics model via linear superposition from a dictionary of candidate functions or atoms by extracting key features from

the data based on the underlying physical principles. This enables to more efficient and accurate predictions (Zemouri et al., 2023).

B. Physics-informed loss function method

This method is aimed to guide the neural network training process so that the training is guided by known underlying physics (Tianren Zhang et al., 2024). Most of the work reported under physics-informed neural network (PINN) have adopted this approach especially for solving engineering and scientific problems because of its ability to handle complex relationships between many physical variables across time and space. Physical knowledge has been incorporated into loss functions to help develop ML models that can capture dynamic patterns consistent with established physical models or laws. Thereby, increasing the capacity of models' generalisation when facing scenarios not seen in training data. One common technique to realise this is to add physics constraints into the ML loss function using Eq. 2-6 (Karpatne et al., 2017).

$$Loss = Loss_{TRN}(Y_{true}, Y_{pred}) + \lambda R(W) + \gamma Loss_{PHY}(Y_{pred}) \quad \text{Eq. 2-6}$$

where the LossTRN is the training loss that measures a supervised data error based on RMSE or cross-entropy between the predictions Y_{pred} and actual values Y_{true}. λ is a hyperparameter that controls the weight of the model complexity loss expressed as R(W). The additional loss function $Loss_{PHY}$ is physics-based that measures the consistency between the predictions Y_{pred} and physical laws, which is then weighted by a hyperparameter γ.

Daw et al. (2022) in the case of predicting lake temperature, in addition to leveraging hybrid physics-ML modelling approach, also modified the loss function based on existing knowledge of physics with the known physical monotonic relationship between temperature, density, and

the depth of water. This was done to inform the training of the ANN to be physically consistent (predictions of denser water are at lower depths than predictions of less dense water). This has resulted in a substantial decrease in reducing the physical inconsistency, expressed as the violation of a density and depth relationship equation, to nearly zero while improving RMSE. In addition to this, if the underlying physics of the system being modelled is known or can be estimated in the form of ordinary or partial differential equations, then this can also be embedded into the loss function to inform the ML training process (Haywood-Alexander et al., 2024). Wang and Yu (2021) proposed a PIML model that included an extra divergence-free regularizer during training that ensures consistency with physical laws, to perform the task of forecasting 2D raw velocity fields of an incompressible turbulent flow. In the case of governing PDEs not being available or being computationally expensive, Raymond and Camarillo (2021) introduced a simpler loss-function approach by utilising only simple but universally applicable laws (conservation of energy) as the informing principle in the error function. This was used to build a neural network to predict the motion of a pendulum, which achieved better performance when compared with traditional, data-only approaches. Several similar studies adopted this method and produced both physically meaningful results as well as improved model generalizability (Li et al., 2021; Elhamod et al., 2022; de N Santos et al., 2023; Faroughi et al., 2024).

Physics-encoded ML techniques

Physics-encoded techniques directly integrate the imposed physics with the architecture of the learner, via selection of operators, kernels, or transforms. Such techniques rely less on the form of the model, but they heavily rely on the physics model it's adhered to. Three examples of physics-encoded learners are described.

A. Constrained Gaussian processes method

Physics information can be embedded in multiple levels into the Gaussian process modelling by incorporating a variety of kernels to enable more accurate and reliable predictions, especially when data is limited and noisy (Chang and Zeng, 2023). Petersen et al. (2022) applied a novel physics-informed GP method to a bridge problem, but with the aim of estimating wind load from acceleration data. In their work, they developed a novel infusion of gaussian process latent-force model with a Kalman filter-based approach. This inclusion allowed for characterization of the evolution of the wind-load, and this is enriched with prior physical knowledge in the form of stochastic information on wind-loads taken from wind-tunnel tests. This work provides an excellent demonstration of how physics information can be embedded to allow the transfer of information from scaled structures.

B. Physics-encoded initialisation method

Physical knowledge can also be leveraged when generating the initial values of ML model parameters before training (Bousmalis et al., 2018). This approach can accelerate or improve the model training process as the initialisation of the weights is informed by contextual physical knowledge rather than a random distribution which would result in local minima issue (Jia et al., 2021). A specific ML technique known as transfer learning can be used to pre-train a model based on physics-based model's simulated data and then refine the model with observed data (Tajbakhsh et al., 2016). For example, Jia et al. (2018) used this strategy to model lake temperature by pre-training a physics encoded recurrent neural network on data from physics-based model and fine-tuned the model with actual data. Read et al. (2019) also demonstrated a better generalisability of such models on unseen scenarios compared to pure physics-based models.

C. Physics-encoded design of architecture method

Different from the physics-informed loss function and the weight initialisation, this method looks at ways to encode physical consistency or physical properties into the ML architecture (e.g., ANN architecture) otherwise often has been seen as a black box (Hassija et al., 2024). Domain knowledge or physics information can be used to specify node connections that capture physics-based dependencies among variables, thereby producing more interpretable results (Sun et al., 2020). The techniques to achieve this can be ascribing physical meaning for certain neuron in the neural network (Daw et al., 2019; Muralidhar et al., 2020). In this method one or more weights are fixed within during neural network forward propagation process according to known physical governing equations (Sun et al., 2020), as well as incorporating symmetries and novel architectures into the ANNs (Ling et al., 2016; Anderson et al., 2019; Wang et al., 2020).

Within the pavement engineering domain, although there exists an abundant amount of existing physical understanding and knowledge on roads, limited research effort has been made to integrate them into the ML modelling process to enhance its overall performance. The following sub-section reviews existing research in pavement engineering leveraging the relevant PEML approach.

Relevant PEML Research Effort in Pavement Engineering

With regards to pavement management, various studies highlighted the limitations and the issues of both traditional and novel ML techniques including the lack of physics information. It has been indicated that more research is required to integrate pavement engineering domain knowledge into the ML process to enhance model's prediction stability, reliability and generalisability (Yao et al., 2021; Song et al., 2022). For instance, Kargah-Ostadi et al. (2024)

has advocated for and discussed several potential applications of PEML to enhance pavement engineering practices, in the areas of pavement analysis and design, pavement condition evaluation and performance prediction. Several studies have attempted to combine physics information with ML to model pavement performance leveraging one or more of the previously mentioned five classes of methodologies.

For example, Chen et al. (2024) used a simple hybrid physics-ML model approach where the output of the pavement responses under load from a physics-based finite element (FE) model was augmented as an extra input for a RF model to predict road rutting using data collected from 99 sections in LTPP database. This approach has demonstrated both one-year prediction improvement by 4.4% and the reduction of prediction uncertainty by 6.76% for multiple years' rutting forecast, indicating an improved model stability and reliability. In addition, several research also focused on developing physics-guided loss function to constraint the training of ANN by obeying pre-defined pavement behaviour rules so that it enables the model's predictions to be more consistent with the theoretical understanding of pavement mechanics.

For instance, for predicting rutting Deng et al. (2024) incorporated existing physical knowledge by following two steps: 1) Generation of synthetic input vectors for the model inputs (number of wheel passes, temperature) which both are known to the accumulation of rutting in asphalt mixture, 2) by sorting each one of the two inputs in an ascending order while keeping the rest of inputs constant at their mean values. This allows for the examination of the individual effects of the monotonic increase of these two model inputs on rutting development. The findings showed that while PINN reduced the accuracy by less than 3%, the model stability measured by averaged cosine similarity, improved from 0.977 to 0.999, and the average coefficient of variation based on repeatedly constructed models reduced by 79% from 0.272 to 0.056.

Narrower confidence intervals demonstrated model's rationality improvement by reducing unreasonable rutting trends with respect to corresponding model inputs.

Pasupunuri et al. (2024) used an IRI equation from the Mechanistic-Empirical Pavement Design Guide (Titus-Glover and Darter, 2001) to inform the loss function in addition to the standard data loss in predicting roughness in concrete pavement. The authors have illustrated PINN's capacity in predicting IRI in concrete pavement with producing acceptable results with a Mean Absolute Error (MAE) of 0.134 m/km, and a coefficient of determination of 90%. A sensitivity analysis was also performed to showcase the reliability and robustness of the developed PINN model. However, the authors did not intend to compare the PINN model with that of a standard ANN.

Another research aimed at supplementing infrequent standard survey data for pavement roughness/IRI with continuous data crowdsourced from connected vehicles using PINN Kargah-Ostadi et al. (2024). The authors chose the physics-guided initialisation (transfer learning) and physics-guided design of architecture (fixing parameters of ANN inner layers parameters to preserve the knowledge of the suspension behaviour) approaches for reconciling two data sources. The model was pre-trained based on quarter-car simulation data and some of its parameters were fine-tuned based on actual standard IRI measurements. Similar to the findings of Deng et al. (2024), the developed PINN model showed that despite an increase in MAE from the physics-based model, the error is still at acceptable range for practical purposes. Notably, the model's stability has improved from 81.82% to 95.18% in the agreement between training and testing accuracy.

To the authors' best understanding, no prior research has focused on developing a PEML model on predicting the defects for asphalt pavement while evaluating and comparing its performance

against traditional ML methods. This could provide valuable insights in implementations of PEML in the enhancement of pavement engineering and management practices as an enabler for future predictive RDT applications.

2.2 DTs and their Impacts on Roads

Triggered by the industry 4.0 initiative, digital transformation has been on-going across a broad spectrum of industries and as a result, new concepts and technologies start to emerge and surface. Amongst many, one of these novel concepts that is gaining an increasing amount of attention and momentum is the DT. It is related to creating a virtual entity of the physical system, providing a connection between the real and virtual systems to collect, analyse, and simulate data in the virtual model to reflect and improve the performance of the real system. It has brought considerable values and benefits across different industries over the last decades, attracting a significant amount of interest from both research and industry communities, and growing in its importance for the years to come (Pires et al., 2019). Hence, the origin, the definitions, the implemented applications and the benefits of DTs are reviewed in this chapter. In addition, the main DT components and required characteristics have also been listed out.

2.2.1 Concepts and Definitions

Before the DT term was coined, the idea of a DT was first brought up back in the 1960s at National Aeronautics and Space Administration where a living model was created for the Apollo 13 mission (Allen, 2021). A network of high-fidelity simulators, represented as a set of virtual assets which were supposed to mimic the structure and behaviour of the corresponding physical spacecraft was used to test scenarios of failure and to refine instructions that are of great importance in critical moments, and eventually bring the

astronauts back from space after an explosion happened in the physical craft (Editorial, 2024).

The term DT was only then proposed during a presentation at the University of Michigan in the early 21st century by Michael Grieves who stated that a DT should be a virtual representation of what has been produced, and should contain three components, namely real space, a virtual representation of the product, and the connections of the data and information in between for optimizing product life-cycle management in manufacturing systems (Grieves, 2014). Since it was introduced in manufacturing systems until today, there has been a variety of definitions of a DT under different contexts in which it is applied in specific industries and fields.

A relatively generalised and consolidated definition for DT is proposed by VanDerHorn and Mahadevan (2021) after systematically reviewing 46 articles where DT definitions have been provided. The authors have defined DT as "A virtual representation of a physical system (and its associated environment and processes) that is updated through the exchange of information between the physical and virtual systems". Similarly, Singh et al. (2021) also put forward a more comprehensive and specific DT definition that can be applied irrespective of the industry or its application: "A Digital Twin is a dynamic and self-evolving digital/virtual model or simulation of a real-life subject or object (part, machine, process, human, etc.) representing the exact state of its physical twin at any given point of time via exchanging the real-time data as well as keeping the historical data. It is not just the Digital Twin which mimics its physical twin but any changes in the Digital Twin are mimicked by the physical twin too".

Different industries where DT is applied may have varied definitions respectively based on their own contextual scenarios. For instance, the DT definition from an official programme as

part of the Digital Built Britain refers to DT as “a realistic digital representation of assets, processes or systems in the built or natural environment” (Bolton et al., 2018). This is a National DT initiative from the UK government to develop DTs for the built environment and infrastructure industry as a whole.

2.2.2 Categorisations and Types

Given the diverse definitions of DTs and its wide-ranging applications within different contexts, this section reviews the common understanding of DTs in terms of the categories of DTs defined as well as various DT types that have been reported in the literature.

According to Kritzinger et al. (2018), the authors classified DTs into three subcategories on the basis of the level of data integration: 1) Digital Model; 2) Digital Shadow; 3) Digital Twin. The lowest level is Digital Model where there is no self-initiated data interaction between the physical and the digital entity. Digital Shadow differs itself from Digital Model as the physical object is connected automatically with the virtual representation. Digital Twin is fully integrated, providing the highest level of connectivity between both spaces (Trauer et al., 2020; Liu et al., 2021). However, most research claimed digital twin despite the actual work being digital model or digital shadow (Liu et al., 2021). The communication level differences are demonstrated in Figure 2.7.

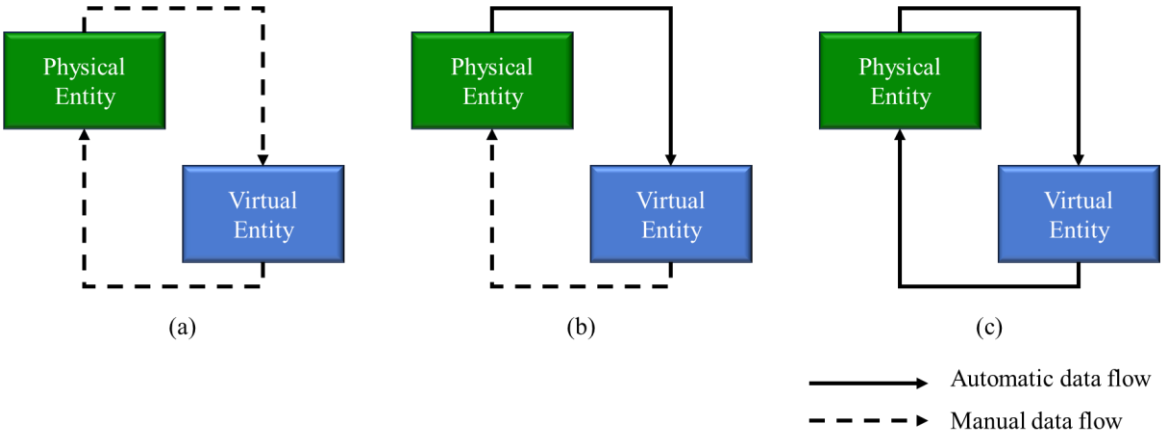


Figure 2.7. Data Flow in a (a) Digital Model; (b) Digital Shadow; (c) Digital Twin

Depending on the intended purposes of a DT, there has been generally five different types of DTs ranging from Descriptive DT, Informative DT which can also be called Reflective DT (Babanagar et al., 2025) or Diagnostic DT (Kibira and Shao, 2023), Predictive DT, Comprehensive or Prescriptive DT and Autonomous DT as described by Wang et al. (2024). Table 2.2 provides a summary on each type of DTs with regards to the definition, benefits and the potential use cases.

Table 2.2. Details on multiple DT types

Type	Definition	Benefits	Use Cases	References
Descriptive	A virtual representation of a detailed and accurate description of the physical asset.	▪ Centralised data repository ▪ Track asset performance over time	▪ 3D Virtual model creation ▪ BIM ▪ Performance visualisation	(Singh et al., 2021)
Informative	A virtual representation of the data and information associated with a physical asset	▪ Real time monitoring ▪ Real time understanding and insight of the asset's behaviour and performance	▪ Track the physical asset's performance over time ▪ Support predictive maintenance and decision-making	(Parmar et al., 2020; Sacks et al., 2020; Bado et al., 2022)
Predictive	A virtual representation of a physical asset that is used to model and predict its future behaviour and performance	▪ Detect and identify potential issues and anomalies before they occur	▪ Predictive modelling for the physical entity	(Sahal et al., 2021; Tu et al., 2022)
Comprehensive	A virtual representation of a physical asset that integrates all the data and information associated with the asset	▪ Provide a complete and holistic view of the asset ▪ Prescriptive analytics and recommendations for future performance	▪ Recommendations for maintenance scheduling ▪ Recommendation for product designs	(Callcut et al., 2021)
Autonomous	A virtual representation of a physical asset that is designed to operate and make decisions without human intervention	▪ Automate tasks and processes ▪ Free up time for other important human manual tasks	▪ Automate process of maintenance activities scheduling ▪ AI-driven decision-making using future-based simulations	(Deryabin et al., 2020)

2.2.3 System Architectures

A DT can be defined as “a formal digital representation of some asset, process or system that captures attributes and behaviours of that entity suitable for communication, storage, interpretation or processing within a certain context” (Steindl et al., 2020). Various system architectures have been proposed for the construction of different DTs. A very basic DT system composes three different aspects: the physical space, the virtual space, and the connection between them to exchange data and information (Grieves, 2014). As an evolution of the initial DT system, the so-called Five-Dimensional DT was then suggested, adding two extra aspects namely data and service aspects (Tao et al., 2019). These five dimensions and their relations are outlined in Figure 2.8. This can be seen as the foundation of more sophisticated DT architectures.

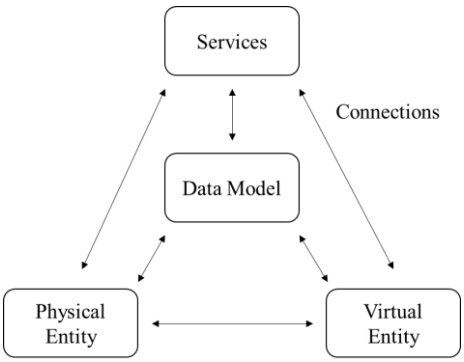


Figure 2.8. Five-Dimensional Digital Twin Architecture

The details of a DT architecture are dependent on the use case and the purpose of the DT. Within the context of Industry 4.0, Aheleroff et al. (2021) proposed a conceptual DT reference model consisting of four parts: Physical layer, Digital layer, Cyber layer and communication for exchanging data across the three layers. Based on this, the authors further expanded the reference model by including an extra application layer as part of the Digital Twin layers, agile value life cycle and digital twin’s integration hierarchy.

Various DT architectures have been proposed across multiple sectors such as manufacturing (Redelinghuys et al., 2019), energy (Steindl et al., 2020), construction (Tuhaise et al., 2023), building and city (Lu et al., 2020), as well as underground (Babanagar et al., 2025). Despite the differences, the architectures share a high level of similarity. For instance, Steindl et al. (2020) suggested a general digital twin architecture model that consists of five layers namely Asset Layer, Integration Layer, Communication Layer, Information Layer and Functional Layer. Similarly, in manufacturing domain, a six-layer digital twin architecture was proposed and evaluated by Redelinghuys et al. (2019) giving a more specific functionality definition within each layer. However, it is largely similar to the five-layer architecture previously mentioned, with dividing the Asset Layer into the physical devices layer and the local controllers layer separately. Moving onto the construction sector, the authors have mostly excluded the physical entities themselves as a separate layer but emphasised the data generated and collected from the physical environment can be the first layer. For example, both Lu et al. (2020) and Tuhaise et al. (2023) presented a five-layered system for DT model in the context of building and city level applied within the construction industry in general. The first layer is Data acquisition, followed by Data transmission layer, digital modelling layer, data/model integration and fusion layer as well as the service layer. Quite comparably to the previous proposals, Babanagar et al. (2025) conceptualised an underground DT architecture with six layers that cover comprehensive details especially on the physical entities and process layer as well as data management layer and their interactions with other layers. The six layers are: Physical Entities and Processes Layer; Sensing and Data Acquisition Layer; Communication / Transmission Layer; Data Management Layer; Data integration / Modelling Layer; and Application Layer. A detailed description, meaning, purpose of each layer and how they are inter-connected are discussed in the next paragraph.

Given there is currently not yet a clear DT architecture for road management, the underground DT system framework provides valuable insights considering the proximity between pavement and underground.

The Physical Entities and Processes Layer (Layer 1) encapsulates the actual physical entities and process involved in different phases from design, construction, operation and end-of-life, as well as the contextual environmental information that is available.

Multiple types of data generated from Layer 1 can be captured by the Sensing and Data Acquisition Layer (Layer 2) by utilising different data collection methods utilising various sensing equipment. For example, contactless data collection such as Radio-Frequency Identification, distributed sensor systems using IoT devices or Quick Response code can be used in this layer.

These data acquired from Layer 2 are then transmitted to the upper layers by the Communication / Transmission Layer (Layer 3) enabled by a broad spectrum of communication technologies such as short-range coverage access network technologies, and 3G, 4G, Long-term evolution, 5G, low-power wide-area networks as well as Wi-Fis (Silva et al., 2018). More detailed on the enabling technologies for DTs are provided in Section 2.2.4.

In the Data Management Layer (Layer 4), collected and transmitted data are stored and processed. This layer fuses and processes data from multiple sources, performing preliminary analysis. Mostly cloud-based database servers act as information repositories for the large amount of data that has been generated. And Layer 4 is closely tied to Data integration / Modelling Layer (Layer 5) where virtual models can be created using technologies like BIM, and where digital twin data can be processed and analysed using data mining and AI techniques that involves a series of machine learning algorithms. Given the importance of

data, model, their fusion and integration, more details are provided in Section 2.2.5 on data fusion and model updating within DT's main components and characteristics context.

The Application Layer (Layer 6) provides a range of actual services and solutions, from descriptive applications to more advanced prescriptive phases depending on the DT maturity to the users and stakeholders. Common services include data-driven modelling, structural health monitoring, early anomaly detection and prediction, optimisation simulations as well as automated decision-making.

2.2.4 Enabling Technologies

Although most of the time DT technology has been viewed as a single technology, it can be more seen as a system of systems, a combination of multiple enabling technologies that construct an intelligent virtual representation of a physical entity and support a continuous two-way feedback loop between the physical and virtual twins (Mihai et al., 2022). That is to say, the popularity of DTs can be largely attributed to the major developments in multiple technologies that surround it in parallel.

In a DT survey conducted by Fuller et al. (2020), it was concluded that the development of DT technology is possible with the same growth experienced in AI and IoT domains which are highly important key enablers for DTs. Mihai et al. (2022) also highlighted the development in IoT or Industrial IoT devices as well as data analytics capacity. The study summarised more explicitly six technologies for DT enablement: 1) ML 2) Cloud, Fog, and Edge Computing, 3) Internet of Things, 4) Cyber-Physical Systems; 5) Virtual Reality and Augmented Reality, and 6) Modelling Methodologies. Qi et al. (2021) formulated the DT model into five dimensions which are physical entities, virtual models, services, DT data and the connections. On the basis of this, a comprehensive list of enabling technologies was presented under each dimension, which is shown in Figure 2.9.

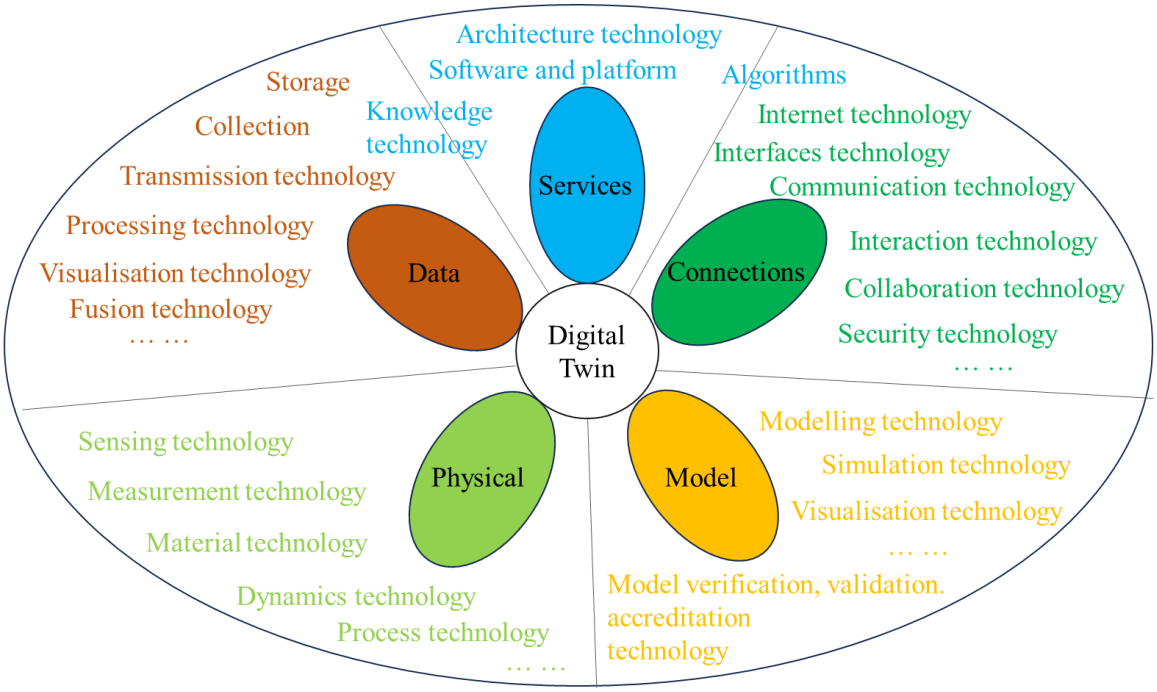


Figure 2.9. Framework of enabling technologies for DTs (After Qi et al., 2021)

In another study, Hu et al. (2021) built on top of the 5-dimension DT model reiterated the DT key enabling technologies such as sensor technology, virtual modelling technologies, data processing and transmission technologies. At the same time, the paper proposed a 6-dimension model by integrating the environment factors in the DT model overall. When it comes to the construction industry, Zhang (2023) demonstrated that the same 5-dimension DT model is feasible within the architecture, engineering and construction industry using the formula and provided technical guidance for DT technologies and applications in the operation and management stage of the building sector.

In addition, although there are multiple enabling technologies that build up a DT, arguably modelling and twinning methods are of key importance to integrate all different technologies in one DT. Therefore, Thelen et al. (2022) provided a comprehensive and focused examination on the element of modelling methods for various aspects of a physical system,

including geometric modelling, physics-based modelling, data-driven modelling, physics-enhanced ML, and system modelling.

2.2.5 Main Components and Characteristics

It has been emphasised by several studies that data is the key or core of a DT (Kaur et al., 2020; Zhang et al., 2022). A practical implementation of DT was conducted for bridge structure health monitoring by Ye et al. (2019) where the authors summarised the key features and capabilities for a bridge DT, which are 1) a digital replica; 2) data; 3) connection to the physical bridge; 4) whole life cycle coverage; 5) common data environment for storage; 6) a visualisation tool; 7) a simulation tool; 8) learning from real measurement data for better future predictions. Regarding data, multiple research projects proposed the full usage of both historical data and real-time data (structured, semi-structured and unstructured) as part of a DT framework to enable the DT to be reflective, comprehensible, useful, and self-evolving (Laborie et al., 2019; Zhu et al., 2019).

Another important component is how the data obtained is going to be used or processed. As mentioned by Chakraborty and Adhikari (2021) one key aspect of the DT technology is to update the DT using the sensor data collected from the physical system and then use the same data for making predictions on the future state. For this, data driven modelling has been revealed as an important element when data is safely collected and stored, and the role of data mining and ML has become enormous, and their advancement directly enables DT development. The same findings have been concluded in another study (Teng et al., 2021) where the authors pointed out the significance of data processing and data-driven modelling within DTs due to their unparallel advantages in adaptability to uncertainty and changes, accuracy, predictivity and simplicity.

In addition to data driven modelling which purely looks at the data, in many domains such as engineering, aircraft systems, manufacturing processes and robotics where there are existing physics-based models with expert knowledge and advanced modelling techniques based on physical laws, DTs could also well be developed based on these models which do not require any data as input for the model to work (Aivaliotis et al., 2019; Sun and Shi, 2021). For example, Yang and Özel (2021) presented a physics-based simulation model that predicted the thermal field solution for the development of a DT in the metal additive manufacturing industry. The Finite Element Analysis method has been widely used to simulate different processes to build DTs for behaviour simulation and prediction purposes (Hinchy et al., 2020; Funari et al., 2021; Sisson et al., 2022). Furthermore, a physics-based model DT could help simulate the physical entity at different scales such as macro and micro, as well as the interactions between objects (Wook Heo et al., 2021; Erdogdu et al., 2022).

What makes the DT technology unique is the fact that it could combine the best parts of both data-driven and physics-based models to enable an improved predictive performance and allow robust decision making for asset owners and operators (Gardner et al., 2020). There are increasing numbers of studies focusing on embedding ML together with high-fidelity physics-based models to improve outcomes within a dynamic DT environment (Kapteyn and Willcox, 2020; Ritto and Rochinha, 2021; Gong et al., 2022). To provide an example, Srikonda et al. (2020) has demonstrated that physics-based models can be first calibrated to field sensor data if there is any and then be used to produce synthetic data for ML model training resulting in improved quality results in the engineering, oil and gas industry.

Establishing the reliability and trust in DTs is vital for their adoption in practice, and therefore to address or quantify uncertainty(ies) is another necessary component in a DT framework, especially given the fact that it is built based on aforementioned elements such as

data, ML model and physics-based models. Each of these has its own potential source of errors and hence uncertainties (Kochunas and Huan, 2021). Multiple methods: Bayesian Inference, Interval Analysis, Fuzzy Theory and Monte Carlo methods have been applied for this task in DTs (Al & Sin, 2021; Lin et al., 2021).

As indicated from the studies, digital twin data is diverse and coming from multiple sources, collected using different types of sensors, resulting in heterogeneous datasets such as dynamic time-series data, image data, video data, and static environmental data, mechanical as well as geometrical data. Data fusion is therefore a critical component to facilitate the flow of information from raw sensory data to high-level understanding and insights within a DT system (Tuhaise et al., 2023). As summarised by Boström et al. (2007), the general data fusion concept is defined as "*the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making*". Liu et al. (2018) summarised the possible fusion operations in a DT environment and their benefits. Table 2.3 and Figure 2.10 demonstrate more details.

Table 2.3. Fusion operation types and benefit

No.	Type of data fusion operation	Benefit
1	Sensor fusion	Better signal quality
2	Physics-based model fusion	Better model performance
3	Data-driven model fusion	Better model performance
4	Sensor and physics-based model fusion	Adaptive physics model
5	Sensor and data-driven model fusion	Robust data-driven model
6	Physics-based model and data driven model fusion	Improved prediction
7	Sensor, physics-based model, and data-driven model fusion	Reliable decision making

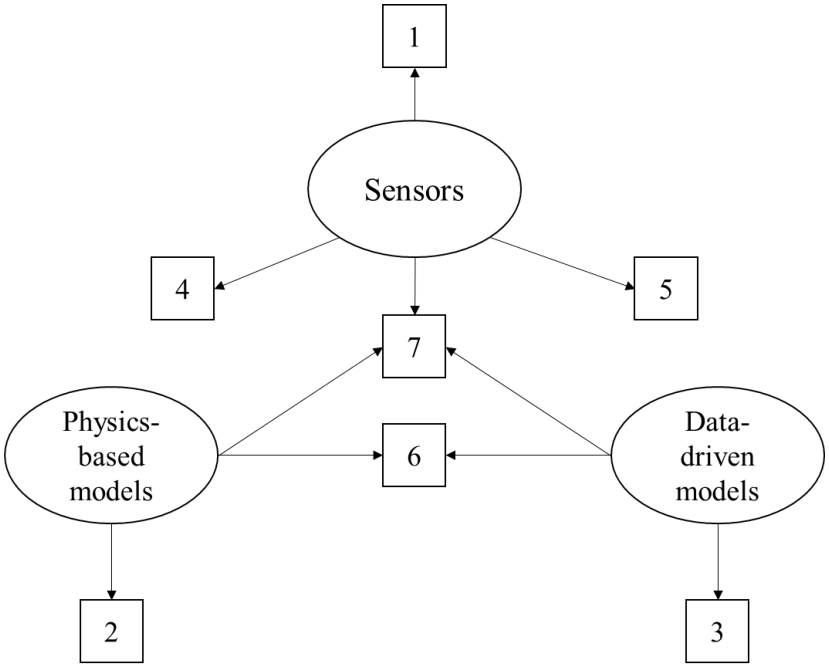


Figure 2.10. Visualisation of possible fusion operation types in DT ecosystem

To enable the fusion of different data types, multiple methods have been suggested to process the data. Firstly, considering the noises and the potential errors experienced in sensors, probabilistic fusion models using Bayesian networks are helpful in dealing with inherent data uncertainty, combined with varying levels of confidence to make predictions (Macías et al., 2024). Secondly, semantic data fusion method is also important when the meaning and context behind data are of significance in the case of semantic annotation for meaningful extraction and analysis in images and videos (Li et al., 2024). Thirdly, ontologies and knowledge graphs methods provide a formal structure for defining different concepts, relationship and rules within a domain such as complex infrastructure management (Yu et al., 2021).

Model updating is another key aspect within a DT system to enable the constant and real-time synchronisation between digital twin models and the corresponding physical objects. It results in a continuous refinement and adjustment of the virtual representation of a physical system that reflecting accurately the state of the object by incorporating data from sensors in

real-time to enable informed prediction, informed decision-making. There have been several techniques proposed within the research community on digital twins to achieve this. They are summarised in Table 2.4.

Table 2.4. DT model updating methods

Names	Characteristics	References
3D Point Cloud Reconstruction method	Uses algorithms to update the geometry of different surface types for physical counterparts	(Dawes et al., 2019) (Xu et al., 2020)
Fractal Theory	Captures complex geometric structures and handle nonlinear relationships	(Chen et al., 2022) (Zheng and Guo, 2024)
Iterative optimisation methods	Compares the difference between DT output and actual output, design optimisation algorithm and find optimal model parameters through iterations	(Zhang et al., 2023) (Hao et al., 2023)
Least squares method	Extracts the most important parameters from the DT model and searches for the parameters that have a greater impact on the model through sensitivity analysis	(Wang et al., 2023) (Coburg et al., 2024)
Bayesian updating	Utilizes time series data to analyse and determine the state of the system, updating the DT model with both historical and real-time data to ensure its synchronization with the physical entity	(Kapteyn et al., 2022) (Titscher et al., 2023)
Neural network method	Updates the weight parameters of the neural network model through an evolutionary algorithm to keep the synchronisation	(Yoon et al., 2022) (Wang et al., 2024)

Additionally, DTs enable the simulation and prediction for the whole lifecycle management of physical entities by the digital entities based on different types of data and algorithms. The DT can reflect the design, construction, operation, and maintenance of the physical entity, e.g., linear infrastructure (Tchana et al., 2019). According to Wang et al. (2022), the implementation of a DT that simulates the whole network lifecycle reduced its lifecycle cost by minimising repeating operations and avoiding risk points and therefore improving the efficiency. The full-lifecycle management can be embedded within ML models to provide enhanced predictive maintenance services on complex equipment (Ren et al., 2022). Visualisation is equally important in helping to present, and in interpreting the data coming to the DT; Technologies such as BIM could have a great potential to present the digital model. For instance, in the rail sector, Kaewunruen and Lian (2019) developed a 6D BIM for the lifecycle management of a railway turnout system. And this DT was beneficial for thorough visualisation purposes, prioritisation of maintenance options, stakeholder collaboration promotion and cost estimation. Apart from data visualisation, real-time updates on model behaviour are also important. A city and building DT demonstrator of the West Cambridge site of the University of Cambridge, UK, has been developed based on a multitier architecture to enhance the operation and maintenance of the assets (Lu et al., 2020). Meanwhile, Yu et al. (2020) also demonstrated a DT to predict future tunnel performance with three key parts 1) data collection module including tunnel highway structure, performance, maintenance records as well as vehicle-mounted laser sensors and accelerometers; 2) prediction module using integrated learning framework; 3) parametric analysis module leveraging a BIM model using the software Autodesk Revit and Dynamo.

2.2.6 Applications and Benefits

DTs have been applied in various industries. To facilitate this section, it was intended to provide a review of DT applications amongst various industries according to the lifecycle of the physical entity, which are namely design, creation/construction/manufacturing, operation, maintenance with decision making. Therefore, this section provides a broad review on DT applications and next section will narrow down to review in detail the application of DTs in the construction industry and built environment.

From the perspective of system design and development, DTs are used to simulate, test, and refine or optimise new products and processes (Jones et al., 2020). They enable the users to digitalise, visualise, and materialise the intangible concepts of complicated systems with numerous components and interconnections such as ships, aircraft and factories. In addition, the design qualities can be evaluated within a DT without going through expensive physical prototypes (Tao et al., 2019). It has been used for the purpose of conceptual design, detailed design, design verification, and re-design as well as to increase the virtualisation of the design concept (Lo et al., 2021). Guo et al. (2019) and Zhang et al. (2017) demonstrated that a DT could enable designers to simulate a whole factory design process considering the layout of the factory, the equipment configuration, material handling and buffer capacity.

During the operation phase, DTs are often used to collect operational data over time. This provides data-driven insights into the performance and distribution of the product that can be also shared across different disciplines within an organisation, so the same data can be fully utilised to make improved decisions (Attaran and Celik, 2023). It also has been used in monitoring operational status, diagnosing faults, optimising power plant performance as well as energy grid development (Kim et al., 2019), achieving enhanced efficiency through automation and a streamlined process as well as reduction in operational cost. In addition,

with regards to sustainability, DT results in better resource management and reduced environmental impact (Kim et al., 2019; Abadías et al., 2020).

At the maintenance stage, one of DT's application is smart asset management (Lu et al., 2020). Macchi et al. (2018) investigated the role of DTs in asset lifecycle management for asset management in general, especially on the maintenance decision-making support based on data-driven insights and predictive analytics, producing economic development with improved assets supporting economic growth. They presented five use cases where DT technology had been applied leveraging its capabilities, for example, sensor data and advanced analytics, to achieve performance predictions and informed decision making. However, the research did not specify what exact improvement DT technology contributed compared to existing systems and technologies, as it only aimed to bring a better understanding of benefits brought by the DT. From the perspective of the maintenance phase within asset lifecycle management, Errandonea et al. (2020) conducted a comprehensive literature review on DTs for maintenance where DT applications of various types of maintenance (reactive, preventive, condition-based, predictive and prescriptive) are discussed. The authors also concluded that DT for maintenance has been the focus of research in recent years and suggested one of the future research areas is to use DT to obtain a more reliable maintenance recommendation system based on the progress made in the calculation of the maintenance impact and its well-defined process. Broo and Schooling (2021) highlighted two key benefits of DTs in general: firstly, their capacity to utilise the large amount of data generated in the connected world enabled by IoT devices. In particular, DTs convert raw data into useful information which in turn is transformed to valuable insights that could lead to better decision-making support. The second benefit is that DTs can

be a useful tool for all stages across asset lifecycle that enables the asset owner to understand the asset condition and to determine the appropriate intervention.

With the aforementioned advantages, Enders and Hoßbach (2019) has suggested that DT applications can serve primarily three purposes: 1) Monitoring where the current state of the physical asset is represented by data; 2) Simulation where behaviour of the physical asset can be reproduced, planned, forecasted, and optimised; 3) Control in which DT applications influence the decisions taken over the management of the physical assets . Therefore, it can be envisaged that the implementations of DTs for roads could demonstrate similar benefits for road management practice, integrated with its own contexts, limitations, and challenges.

2.2.7 DTs in Construction Industry and Built Environment

As mentioned above, DT technology has become increasingly popular and been applied across a large spectrum of industries, ranging from manufacturing, aerospace, automotive, urban design, engineering, medicine, medical patient care, sustainability, as well as IT and transport (Singh et al., 2022; Javaid et al., 2023; Kanaga Priya and Reethika, 2024; Mythily et al., 2024). Given the main aim of the thesis is to investigate DT's impact on pavement performance modelling which falls within the civil engineering domain, this section reviews the application of DTs within the built environment to understand its benefits. DTs have already been implemented in recent years in the construction sector and the built environment. Specifically, there are multiple examples of DT applications for different types of assets (buildings, bridges, roads or even a whole city). Firstly, Lu et al. (2019) developed a dynamic DT in building levels and demonstrated a use case with a present asset monitoring function and a predictive maintenance function which analyses real-time data and forecasts the remaining lifecycle of the asset. This research presented a five-layer DT architecture to realise the true value of data and the integration models involved, and the authors also

suggested that integrating AI supported decision-making and data analysis functions would highly improve the whole DT system as it is mainly based on data. In addition, with regard to bridge assets, DT approaches have been recently applied as demonstrated in Shim et al. (2019), Sofia et al. (2020) and Ye et al. (2019). Shim et al. (2019) proposed a new generation of bridge maintenance system by using a DT model concept for more reliable decision-making, especially by introducing automatic bridge inspection and the computation of a time-dependent performance indicator for reliability index evaluation and an associated prediction model, which successfully helped to avoid duplicated work and data loss compared to the traditional methods, and this can be implemented in other similar assets, for instance, road. Shim et al. (2019) also addressed the advantage of a DT and its capacity to visualise by presenting actual performance data via a live sensor. Ye et al. (2019) discussed an exploratory study towards creating a DT of bridges for structural health monitoring purposes, which solved the issues experienced in the existing systems, for example lack of storage for large and heterogeneous datasets, low efficiency of data query and data source incompatibility. Moreover, at the city level, Mohammadi and Taylor (2018) proposed a smart city DT paradigm that enables real-time visualisation of spatiotemporal fluctuations of the city because of human-infrastructure-technology interactions, which demonstrated its capacity to provide predictive insights into the growth and performance of the city. In addition, Lu et al. (2020) presented architecture and explored the development of a DT at building and city levels through implementing a DT demonstrator based on the proposed system architecture. The DT demonstrator offers intelligent anomaly detection, environment monitoring, maintenance optimisation and prioritisation services which are provided via IoT sensors, data integration, data analytics, and ML algorithms. However, the performance analysis and evaluation of the demonstrated DT system were not conducted partly because it

was one of the first few exploratory pilot projects of city level DT and the DT technology itself was still in its early stage of development. Finally, for road tunnelling assets, Yu et al. (2020) improved highway tunnel pavement performance prediction with an accuracy of 94.90% based on the DT concept, which is an improvement on existing performance prediction models.

Each element of a DT is experiencing unprecedented technological development and could be explored to understand where and how they could be used to digitalise and improve the road asset management. The next sub-section provides a review on the existing work on road DTs, and the current state of road lifecycle management where the DT relevant technologies have been applied as well as the impact.

2.2.8 DT Impacts on Roads

The research on DTs for roads is still at its beginning, so there has only been a handful studies directly related to DT for roads despite the benefits shown in similar linear infrastructures. However, it has been suggested by Steyn (2020) that the abundance of data generated from sensor networks in this industry 4.0 age should be managed in the pavement realm to help pavement engineers understand what would be made possible and potential benefits for pavement infrastructure management. The same vision has been shared by Kaliske et al. (2021) where the authors have emphasised the need for research and innovation for a “road of the future” that is suitable for future mobility. Matchett & Wium (2022) reiterated that DTs will have a high potential to support and extend the functionalities within the road infrastructure asset management processes. Chen et al. (2022) presented basic Road DT requirements after reviewing the requirements for DT developments generally.

From late 2021 to 2024, more practical research has been conducted to explore the generation of road DTs and demonstrate their benefits and impacts. Generally, Steyn and Broekman (2022) presented a case study for a DT development of a local road network where the benefits are clearly articulated, such as the continuous data sharing objectively with the infrastructure owner to support efficient and timely remedial and maintenance planning as well as for improved sustainability and life cycle costs. Moreover, Chen & Brilakis (2024) developed a proof-of-concept DT data structure and an integrated cloud architecture for roads that could support the application of DT technologies to facilitate road lifecycle management and improve decision-making processes. In addition, multiple different computer vision technologies and advancements have been used and proposed in converting physical roads into a digital entity or geometric road DT, such as Google images or point cloud, photogrammetry (Ding and Brilakis, 2023). For example, Jiang et al. (2022) have proposed a systematic method for generating a DT for an old highway using existing online map data based on road engineering expertise and it has been successfully tested in one section of the A1 motorway in the UK to produce a DT model with a relatively good accuracy. Pan et al. (2024) also proposed a framework for efficiently and automatically creating a graphical representation of highways based on point cloud data, which has shown the potential to improve road management processes such as road inspection, maintenance and upgrading, reducing the amount of traditional survey work a road agency would normally perform taking consideration of labour and time. Based on this work, Davletshina et al. (2024) further developed a method to detect and apply geometric changes to road geometric DTs automatically.

From the road lifecycle perspective, various research studies, though still in their infancy, have been carried out to investigate the applications of DT across multiple stages in the lifecycle of a roadway infrastructure. For example, during the road design and construction phrase, Meža et al. (2021) explored the development of a functioning DT for monitoring road construction progress using secondary raw materials. Marai et al. (2020) proposed a methodology to create a DT of the roads' infrastructure for roads in operation, by deploying a DT box composed of a 360° camera, GPS device and other IoT devices for sensing environmental measurements such as ambient temperature and humidity to provide a better understanding of the contextual circumstances on the roads. Similarly, Niaz et al. (2022) explored DT technology with different enabling technologies to track and control transportation systems of a road in operation online and demonstrated a framework to ensure that data can be understood and processed in real time between the real and the virtual world. In terms of pavement maintenance, Chen et al. (2022) explored suitable ML approaches for a road DT and proposed a DT-based framework for road condition prediction that inputs historical and real-time data from the whole road lifecycle into machine learning algorithms to predict future road performance. Comparably, Consilvio et al. (2022) presented an architecture of a DT-based decision support system for road maintenance, demonstrating its application to road pavement condition evaluation. The study revealed that by using this system, a 10% decrease in the volume of major interventions and 12% drop in maintenance cost is expected, indicating better decision-making, and efficient cost management overall. Further research efforts have been made on understanding how DTs can practically be implemented in highway maintenance (Yin and Kumar Reja, 2024) as well as developing a personalized maintenance alert generation system for road infrastructure management (Luo et al., 2024).

2.3 Optimisation methods for data collection frequency

The process of digitalisation across multiple industries is rapidly advancing and as a result, large number of sensor data is produced at an unprecedented level. These require increased data storage, computing power and resource as well as transmission bandwidth resources to manage, process and analyse data with a much higher financial expenditure. In addition, the digitalisation requires enormous support from physical infrastructure such as power grid and energy supply, which has a negative impact on net zero carbon emissions goals. Therefore, there has been a popular research field to investigate data collection frequency optimisation methods to identify the use of a smaller amount of training data instead of the whole training data that would still be able to produce a ML model that achieves an acceptable prediction accuracy (Silva et al., 2024). This section reviews the methods and techniques used for optimising sensor data sampling interval.

For most industrial applications and processes, the use of wireless sensor network enables continuous monitoring and analysing. Harb & Makhoul (2017) presented a data collection mechanism that allowed sensors to adjust their sampling rates based on the variations of its environment by three approaches: 1) analysing data variances via statistical tests; 2) set-similarity functions; 3) distance functions. The authors demonstrated a reduction of up to 80% in the number of acquired samples. Comparably, Al-Qurabat & Kadhum Idrees (2017) also adapted the data collection frequency based on similarities of the data in the consecutive periods by using Euclidean distance in the dynamic modification of the monitored environment conditions. In the field of manufacturing process modelling, Lipp et al. (2020) also investigated when to collect what using sensors and proposed an optimised data load via flexible process-driven methods. An amount of 39% decrease of data load was achieved compared to traditional and less flexible monitoring methods. Optimisation of data

acquisition strategy is also of paramount importance in environment monitoring, Chen et al. (2020) proposed an adaptive genetic algorithm that can dynamically change the frequency to collect data with a smaller acquisition frequency, thereby reducing the sensor energy consumption. In agriculture, Kar et al. (2020) demonstrated an extensive method for optimising data collection frequency using Autoregressive Integrated Moving Average modelling by changing the sampling rate of the entire time series data set from 15 to 180 minutes, thereby achieving the suitable interval of 60 minutes with minimum redundancy and randomness in the data.

Another common approach among existing research to update or decide the sensor data capturing cadence is based on models built based on the data collected using adaptive sampling rates and evaluating their performance by comparing against a pre-defined threshold. This has also been applied in multiple different sectors. For example, Wang et al. (2016) proposed an optimised obtaining strategy for acquiring sensor data based on the characteristics of the regular changes in sensed data in large-scale monitoring networks connected to the IoT, reducing data collection and transmission quantity requirements. Specifically, it used a linear regression model for sensor data to regulate acquisition frequency adaptively based on whether predictions made from linear regression models are within a pre-defined error range. Amongst other IoT applications, Čulić Gambiroža et al. (2022) stated that the majority of IoT sensors would collect data in short equally spaced periods resulting with large amount of redundant or irrelevant data and developed a dynamic monitoring frequency algorithm that ensures a sensor only collects data when a change in monitored phenomenon value exceeds a predefined threshold. In addition, in a smart building study conducted by Haidar et al. (2019), the authors developed a building occupancy prediction model with satisfying accuracy using data from multiple sensors, and they used the

RF algorithm and compared the models' performance under a range of data collection frequencies, ranging from 1 minute to 60 minutes. The research concluded that an interval of 15 minutes and 20 minutes produced the satisfying accuracy of at least 90% in R^2 which was set up as the threshold, instead of every 1 minute. Van Wyk et al. (2017) took a similar approach in the medical industry where different ML classification models were tested with different sampling rates and their performance accuracies were compared. It was found that despite an initial sharp decrease in classification accuracy when changing the data collection frequency from 1 minute to 10 minutes, the accuracy decreased only marginally when the data collection interval was set to between 15 to 60 minutes, thereby providing insights that would lead to the optimal data collection frequency as well as the design of data acquisition systems at hospitals. In fuel consumption prediction, Almér (2015) also tested and compared two different collection frequencies by assessing the accuracies of several ML models.

For civil infrastructure management, there has recently been large amount of research on instrumenting sensors in the infrastructure for detecting defects and structural health monitoring (Bhatta and Dang, 2024). For example, various types of IoT sensors and devices have been used in buildings to obtain data and achieve multiple monitoring purposes such as building vibration (Ibrahim et al., 2019), safety (Lin et al., 2021), earthquake warning (Won et al., 2020), ground shaking (Duggal et al., 2022), and crack width prediction (Lee and Lee, 2017) as well as structural discontinuities (Zabielski and Srokosz, 2020). For bridges as well, research on structural health monitoring by embedding sensors have been prevalent. They can be used extensively to monitor real-time strain distribution under load (Mohapatra et al., 2022), monitor bridge displacement (Hou and Wu, 2019; Shrestha et al., 2020), and classify bridge vibration (Shrestha and Dang, 2020), The similar work has also been increasingly conducted in the pavement domain for pavement health monitoring (Ye et al., 2022, 2024;

Al-Sabaeei et al., 2024). However, to the author's best understanding, as also acknowledged by Sun et al. (2024) and Yang et al. (2015), there is generally a lack of research on sensor data collection frequency to infer the appropriate sampling rate that ensures optimal data amount for relevant data storage and processing, especially within the context of a road digital twin.

2.4 Summary of the Research Gaps

The gaps identified as part of the literature review process, which this study will address, can be summarised below.

1. Research on RDTs is still at its beginning stage and most recent work primarily focused on the generation of a descriptive and informative RDT based on images and point cloud data, a holistic RDT-based decision-making support theoretical framework with layered structure illustrating main components for various types of RDTs and their inter-connections does not yet exist.
2. Regarding the pavement performance modelling research area, the state-of-the-art has been developing prediction models using ML approaches. There is a huge research gap in integrating the existing pavement knowledge domain expressed by physics with ML to overcome the limitations and drawbacks of ML models.
3. Current research on RDTs has focused on the aspect of the automatic generation of geometric RDT and road maintenance using image data, point cloud data and maintenance text log data. Limited study has investigated predictive RDT with the use of numerical road condition data for performance modelling and identification of suitable data collection frequency.

3 METHODOLOGY

3.1 Introduction

As the literature review chapter revealed that there has been insufficient research to investigate DTs on improving RAM. There are also gaps in DT's capability in pavement performance predictions as well as the appropriate RDT data collection frequency with the provision of historical and real-time data. There is no definitive piece of research that defines a DT-based decision-making support theoretical framework that could be used for road lifecycle application developments. To address these gaps, this chapter presents the research methodology developed to establish a DT-based decision-making support theoretical framework for roads. Its key components and their inter-connections are also described. This chapter is composed of three discrete parts:

(1) Research methodology (Section 3.2) which entails the end-to-end methodology used to conduct this piece of research;

(2) DT-based decision-making support theoretical framework (Section 3.3) explaining the key layers, components for the development of an RDT and their expected interactions as well as applications;

(3) Application of the DT-based decision-making support theoretical framework (Section 3.6) which illustrates the methods or processes used to apply the developed framework in two case studies.

3.2 Research Methodology

First, the research aims and objectives were identified, initially, to develop a generalised methodological approach for the research. Next, a comprehensive literature review was carried out to; i) explore and understand the definitions, concepts, applications of DTs across

different industries and how they could be implemented in the domain of RAM; and ii) to improve the existing approaches in pavement performance prediction and to identify the optimal data collection for different future prediction ranges with the use of sensor data (both described in Chapter 2). According to the findings from the literature, a DT-based decision-making support theoretical framework for road lifecycle was then developed. The framework is described in detail below in this chapter. Based on this framework, two case studies were conducted as specific applications of the proposed framework to test its practicality and suitability.

The first case study was used to evaluate the performance of the developed framework using historical data from a US public database. In particular, the performance of a DT-based pavement performance prediction model was assessed and compared against the existing ML approaches. The details are described in this chapter, and the results are presented in Chapter 4.

The second case study was to investigate the aspect of sensor data and the optimisation of the data collection frequency within the RDT framework. This is achieved by utilising data generated from a laboratory experiment instrumented with sensors at the University of Birmingham National Buried Infrastructure Facility (NBIF). Details are shown in this chapter while the results are provided in Chapter 4. An overview of the research methodology is presented in Figure 3.1.

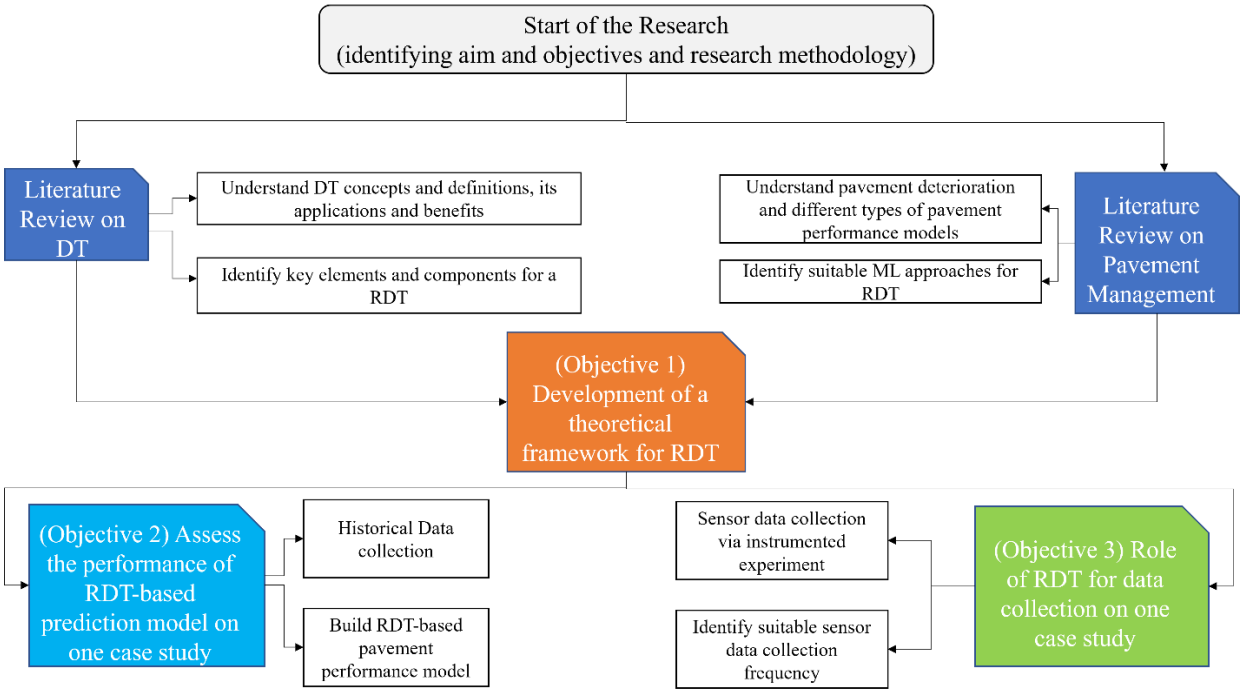


Figure 3.1. Research methodology

3.3 A DT-based decision-making support theoretical Framework for road lifecycle

Following the extensive review of literature on DTs and RAM in Sections 2.1 and 2.2, a few established DT architecture and framework were identified (Lu et al., 2020; Tuhaise et al., 2023; Babanagar et al., 2025), and based on which, a DT-based decision-making support theoretical framework for road whole lifecycle was produced as demonstrated in Figure 3.2.

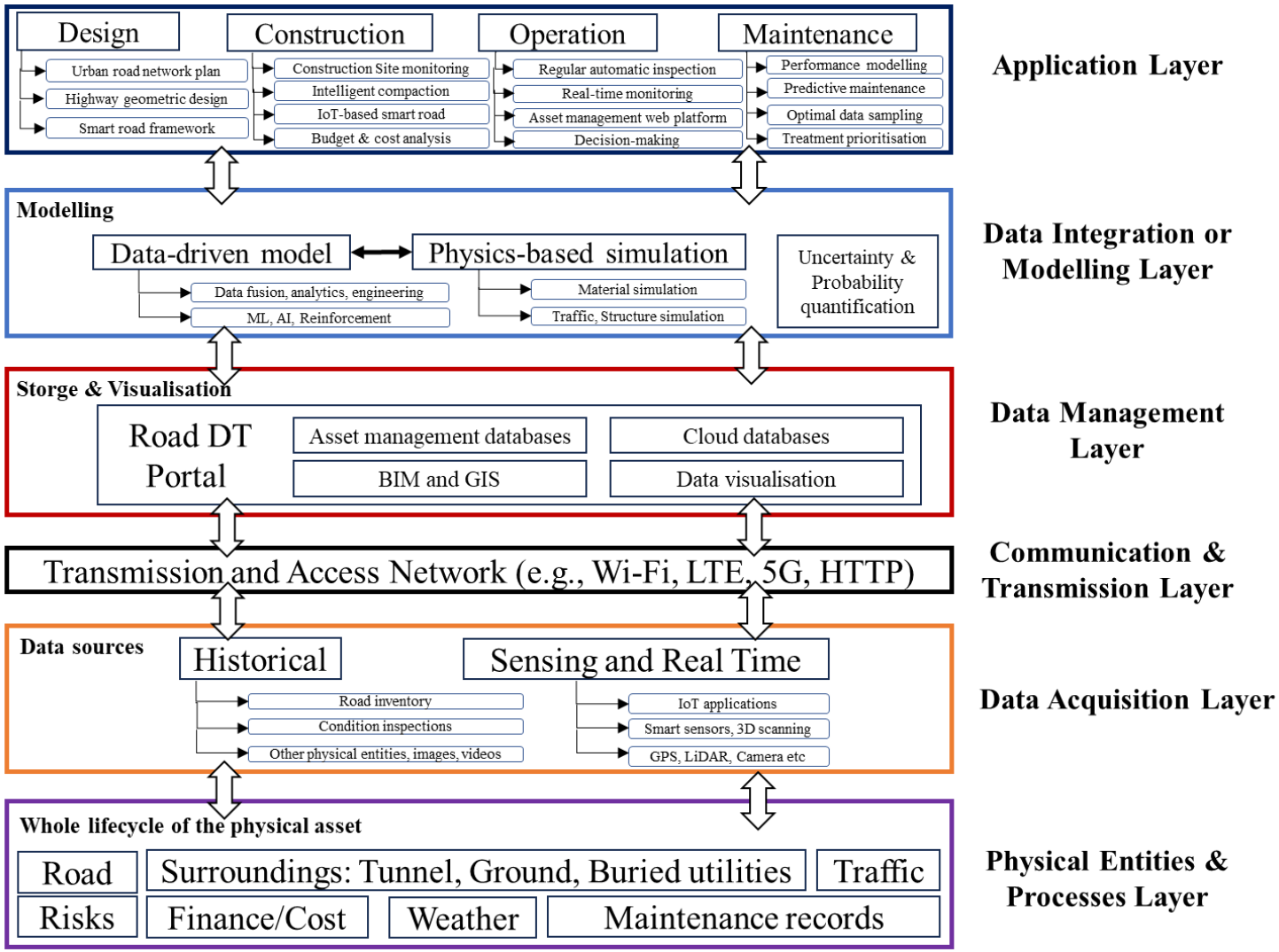


Figure 3.2. DT architecture-based decision-making support theoretical framework for road lifecycle

The developed decision-making support framework for the whole lifecycle of roads is based on various DT architectures reviewed in Section 2.2.3. This framework covers six architectural layers 1) Physical Entities & Processes Layer; 2) Data Acquisition Layer; 3) Communication & Transmission Layer; 4) Data Management Layer; 5) Data Integration or Modelling Layer; 6) Application Layer, with key tasks and enabled technologies, as described below:

1) A physical layer that captures various entities and processes integral to the whole lifecycle of a road is a necessity for creating an RDT. It can include a variety of information such as (i) road site geometries, and geologies, as well as the surrounding environment of the road site, and the inter-dependent assets such as ground, tunnel and pipes; (ii) Information about traffic, climate and weather of the site; (iii) Maintenance records and various costs in utilities and services.
2) Data Acquisition Layer captures the data from the physical layer, with various resources. Data sources are the inputs to an RDT, which can be generally categorised into two components: historical data stored in databases and real-time data from various sensing equipment. Both data types can be used to start the development of the RDT. Historical data includes data on road inventory, condition of the defects, traffic information as well as weather and climate. Real-time data could be generated from mobile phones, IoT applications, sensors embedded in vehicles, as well as 3D scanning, GPS, LiDAR and cameras. A majority of pavement performance studies have used historical data available from a public repository or transportation agencies. For a DT, multiple types of data would be expected, such as numerical, categorical, time series, images, videos and other variations (Dihan et al., 2024).

3) The Communication & Transmission Layer is aimed at transferring the acquired data to the higher layers for management, analysis, especially for modelling and simulation. As presented in the framework, various communication technologies could be used in this layer, such as short-range coverage access network technologies such as Wi-Fi, Zigbee amongst devices and wider coverage technologies with 3G, 4G, Long-term evolution, 5G as examples. In addition, hypertext transfer protocol is commonly used to connect to the web server where data storage and analysis take place.
4) The Data Management Layer plays a pivotal role in the proposed framework as it's a place where all data is stored. In this layer, due to the complex and massive amount and varieties of data collected, effective and hierarchical data storing are needed. Multiple asset management local databases as well as cloud-based databases can be used to achieve this. Also, this common data environment should be closely accessed and updated by BIM or a 3D Revit model and different data visualisation software to enable graphic user interface portal of a road digital twin to demonstrate the virtual representation of the physical road. It can act as the control panel for the road management system.
5) The Data Integration or Modelling layer aims at integrating all the data resources and different models that are available. In this modelling stage, which is the main *"brain"* of an RDT to provide various functions on the data received from multiple sources and to develop models. Key components such as data-driven modelling, physics-based modelling as well as uncertainty and probability consideration are utilised to ensure the most accurate predictions. Data-driven modelling can be composed of a standard machine learning model development pipeline including data pre-processing, data cleaning, and selection of a ML algorithm, meanwhile it can also be RL models to identify the optimal choice in a maintenance decision-making context. Physics-based modelling on the other

hand, simulates the physical behaviour of the pavement condition by considering the material of the pavement layers, the surrounding environment conditions and the interactions with other assets. Both can be categorised into inter-dependency modelling (Setola and Theocharidou, 2016) as they use data from various sources that may have causal effect in the physical layer at the logical level. Uncertainty and probability quantification is another aspect to consider as part of this layer to incorporate the errors and uncertainties experienced in data collection, as well as the modelling process.

a. After collecting data, it is necessary to go through a process of data cleaning and pre-processing, followed by filtering the most relevant features for model learning and development. Depending on the essence of the task, a wide range of data pre-processing and feature selection techniques could be applied. The end goal of this exercise is to produce a complete and high-quality dataset suitable for the ML model training and testing stage (Marcelino et al., 2021).
b. As mentioned by Carter et al. (2023), data pre-processing is an extra layer to ensure data quality within a DT environment. In summary, common data issues are 1) data duplicates; 2) missing values; 3) data outliers; 4) noise in the data; 5) illogical data according to domain knowledge. The corresponding data pre-processing techniques to address these data problems are 1) Duplicates removal; 2) Outlier detection and removal; 3) Fit and interpolation of missing values 4) Noise reduction through data smoothing.
c. Once the dataset is pre-processed, and relevant variables have been chosen, the next step would be to apply the ML methods and/or RL algorithms to develop data-driven models. Within a DT context, a virtual representation means a data-driven model and/or a physics-based computational model. Data-driven models are created based

on data with the help of computational intelligence and ML methods (Solomatine et al., 2009).

d. Physics-based models are models which are created to represent the system based on existing knowledge and known physics. In comparison to purely data-driven models, physics-based models offer a higher level of interpretability, reliability and rationality in the prediction capacity. Ritto & Rochinha (2021) provided the following summary of the characteristics of a physics-based model:

    - They are constructed based on physical principles or laws that govern the behaviour of the physical system, such as Newton's second law and constitutive models.
    - Every parameter of the model has a clear physical meaning.
    - They can be high fidelity time consuming computational models of complex engineering systems (Farhat et al., 2003).
    - There is an option to calibrate the model at a given operational condition and use it for analysis in different scenarios.

6) The Application Layer is the tope and implementation layer of the decision-making support framework that presents the actual potential applications that DT could provide for the life-cycle management of linear infrastructure such as roads. It is worth mentioning that different applications can be achieved by different types of DTs mentioned in Table 2.2, namely Descriptive, Informative, Predictive, Comprehensive, and Autonomous. For design phase, such critical tasks such as route selection, pavement design and budgeting can be solved in specific applications such as urban road network planning and geometric highway design enabled by engineering data and enabling technologies or models adopting a comprehensive DT. During construction, resource

allocation, material management, schedule coordination, and quality assurance can be achieved by functions as site real time monitoring, which can be provided by a Descriptive and Informative DT. For example, Meža et al. (2021) integrated BIM with sensor data to leverage dynamic data querying when employing secondary raw materials for road construction. For road operation stage and maintenance stage, functions or services provided as part of the Predictive, Comprehensive and Autonomous DT can help addressing tasks such as field pavement condition inspection, real-time defect monitoring and prediction, as well as optimised maintenance treatment strategy scheduling and decision-making prioritisation.

3.4 Scope of the Study

The scope of the research mainly focuses on the enabling methods for a predictive RDT, leveraging different data sources within the Data Acquisition Layer and modelling techniques in the Data Integration or Modelling Layer based on the developed decision-making support framework to achieve pavement performance modelling or predictive maintenance and the optimal sampling rate services in the Application Layer. It starts with historical data collection, while integrating data-driven ML modelling process, with the physics-based simulation as well as the consideration of uncertainty quantification. The research then also investigates the suitability of sensor data collection frequency using the modelling methodology presented in Figure 3.3 leveraging sensor data collected from an instrumented pavement in NBIF lab environment. Although both functions addressed in this research work are falling primarily within the maintenance phase of road lifecycle for a predictive DT, and no specific case studies have been conducted for assessing other DT layers (e.g., data management Layer, or data communication & transmission layer) and for other stages of the road lifecycle (e.g., design, construction and operation) as part of the decision-making

support framework, a thorough discussion is provided in each case study to estimate their potential impacts on them.

3.5 RDT Framework's Modelling Methodology

To achieve the defined Objectives 2 and 3 (in Section 1.2), a novel DT-based modelling approach was implemented. Based on the main components from the developed decision-making support theoretical framework while integrating physics with the ML model, two different approaches were used to predict road rutting and IRI for multiple road sections in the short term (1 year) and long term (2 to 13 years). The approach used two primary data sources: historical road condition data and physics-based finite element (FE) simulation data to ingest specific physical domain knowledge into the ML development process for the modelling of the pavement rutting and IRI. This study then compares the performance of different models, considering ML's inherent uncertainty expressed in the form of variance, developed based on these sources of data. A step-by-step process flow is illustrated in Figure 3.3 to describe the methods used in the modelling part within the framework. The developed framework and modelling methodology have then been applied to two case studies: one uses historical data from a public long term pavement performance database, and the other utilises real-time data produced from instrumented sensors for predicting future pavement performance and identifying optimal data collection frequency in different prediction ranges.

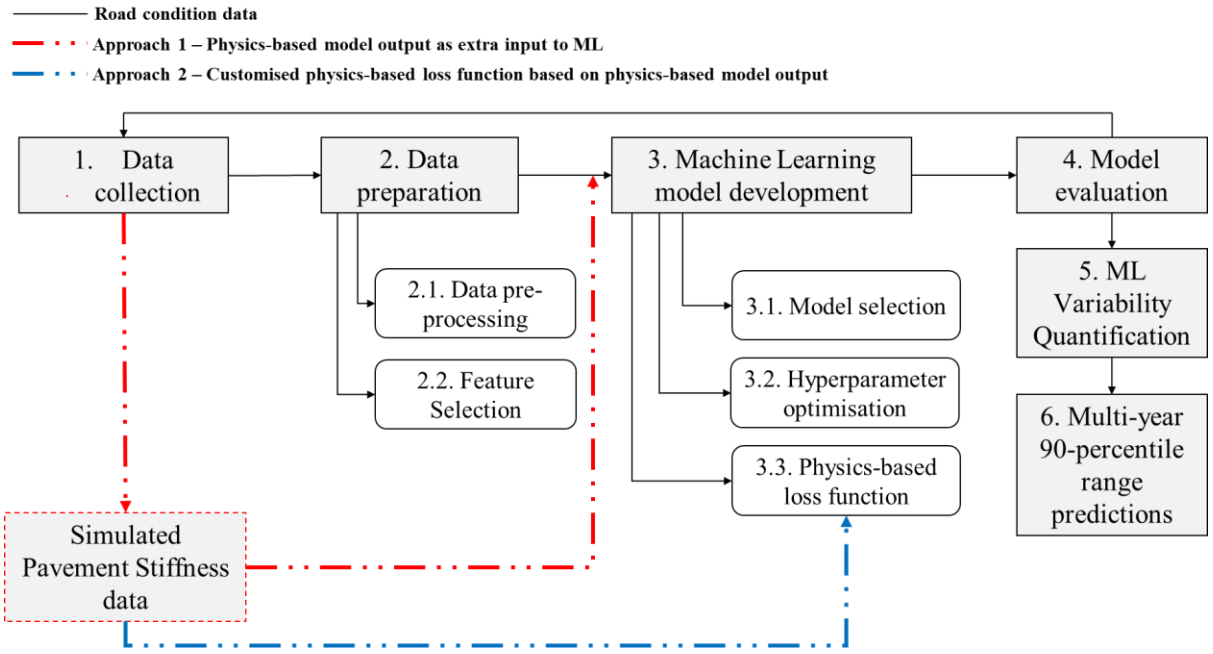


Figure 3.3. Key steps taken to conduct the research. The black lines represent the steps taken for pure data-driven ML model development. The box dotted in red is related to the physics-based model, and the coloured lines in red and blue show two different approaches to interact the physics-based model with the data-driven model

In summary, the adopted methodology consists of the following steps (Figure 3.3): 1) Data collection; 2) Data preparation (e.g., pre-process, clean the data and variable selection); 3) ML model development based on processed data considering choice of the model and hyperparameter configuration; 4) Evaluation of model performance through k-fold cross validation techniques on training data, and testing model performance on unseen data; 5) ML model variability quantification; 6) Making multi-year 90th percentile range predictions based on the variability of the ML model quantified from Step 5.

It is worth clarifying that Step 5 quantifies the ML model's predictive variances which come directly from different training data, as a reflection of the level of ML model's prediction uncertainty. Essentially Step 5 produces an interval that indicates how much fluctuations in the model's predictions for a given input after training the model multiple times. This is a commonly used approach for ML uncertainty quantification (Varley et al., 2016; Zhou et al., 2021; Eghrari et al., 2023; Blasco et al., 2024). Based on the interval, a 90th percentile range

can be produced for predictions each year to enable a probabilistic multi-year prediction approach in Step 6.

3.5.1 Introduction to common ML models in pavement performance modelling

As indicated by the literature, popular ML models that have been well adopted in modelling pavement performance research are ANN, RF, KNN and SVM (Justo-Silva et al., 2021; Marcelino et al., 2021; Jaya et al., 2023). This section provides a brief introduction to these ML models.

3.5.1.1 RF Model

RF is a supervised ML algorithm that is used widely in classification and regression problems because of its high prediction accuracy (Gong et al., 2018; Han et al., 2020; Yu et al., 2021). It is one of the decision tree algorithms where RF builds multiple decision trees and combines the results from each tree together to get a more generalised and accurate result.

RF regression was used to construct models to predict the value of rutting in the next year given the defined inputs. RF uses Bootstrap and Bagging Aggregation ML techniques (Lee et al., 2020). Its foremost advantage is the fact that it effectively deals with overfitting issues by joining multiple sub-datasets, while it requires less time for processing data when compared with other methods (Saikiran et al., 2021). The following steps were taken to utilise the RF regressor algorithm:

- Step 1) The whole dataset was used to build decision trees based on the number of defined estimators.
- Step 2) Individual decision trees were constructed.
- Step 3) Each decision tree generated an output.
- Step 4) The final output was considered based on averaging for regression.

3.5.1.2 ANN Model

An ANN is an AI deep learning method to process information based loosely upon the structure of biological nervous systems such as human brains (Haykin, 1999). ANN models are composed of three main elements: nodes representing neurons, connections between the nodes and weights (Mehrotra et al., 1997). The neurons are constructed into three or more different layers including input layer, output layer, and one or more intermediate layers which are also called hidden layers. The input layer takes the initial data fed into the network whereas the output layer produces the final regression or classification output for the given inputs. The hidden layers are where all the computational processing is performed (Agatonovic-Kustrin and Beresford, 2000). Most ANNs are of a simple form, that of a fully connected feed-forward network, and its structure is presented in Figure 3.4.

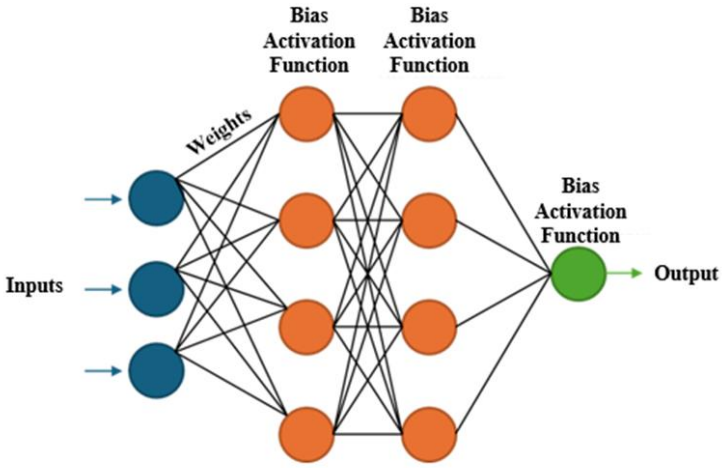


Figure 3.4. Common ANN structure with two hidden layers

Each neuron in the hidden and output layers calculates a sum based on the input values and the weights. This output is further modified using the activation function which contains bias in each neuron to introduce non-linearity (Dongare et al., 2012). The common activation functions are sigmoid, rectified linear unit (ReLU) and tanh (Szandała, 2021).The output of

the output layer is the outcome whereas the outputs from hidden layers are fed into next layer of neurons as inputs (Shanmuganathan, 2016).

The training process of an ANN updates the weights and bias in each iteration based on the loss function calculated by comparing the model's prediction output with the actual data. The goal of the whole ANN training is to minimize the error produced from the loss function. The typical function used to measure the performance is the mean sum of the squares of the residuals between the predicted and real values (Dongare et al., 2012).

3.5.1.3 KNN Model

KNN is a non-parametric model according to a simple voting decision rule where the target of a given point is predicted by averaging the targets of neighbouring samples (Nader et al., 2022). It is a supervised learning algorithm that makes predictions or classifications by finding the "neighbours" of a new data point based on the proximity to other data points. The "k" in KNN means the amount of data points considered that are nearest. Based on their Euclidean distance from the target data point, these neighbours are selected. The KNN algorithm works following below steps (Chen and Shah, 2018):

1) Select the value of k: the number of neighbours to consider when making predictions.
2) Calculate distance: For each new data point, KNN calculates the distance between this point and every other point in the dataset. Mostly Euclidean distance is used.
3) Find the k nearest neighbours: Once distances are calculated, the algorithm identifies the k closest data points to the new point.
4) Classify the new point: In regression, it predicts a value based on the average or median of the k neighbours.

3.5.1.4 SVM Model

The SVM is a supervised learning method used for classification and regression problems. The original idea of the SVM is to find a hyperplane that has the largest margin to the two categories (Jakkula, 2006). It involves following elements:

1) SVMs find a hyperplane separating classes in a multi-dimensional space
2) The hyperplane is the optimal plane that maximizes the distance between classes
3) The algorithm uses a kernel to transform data and find the boundary between outputs

3.6 Introduction to the Case Studies

Two case studies were conducted using different data sources to demonstrate the modelling methodology based on the developed framework, to investigate the enabling methods for a predictive DT for roads. One case study was performed to show DT's impact and advancement for pavement performance prediction using historical data from a public database. The other case study leveraged real-time sensor data from a controlled environment (lab experiment at NBIF) to explore how a DT could help planning for future scenarios where the proliferation of data is given and then identify the appropriate frequency of sensor data collection to ensure the predictability of pavement performance.

3.6.1 Case Study I – Pavement Performance Modelling with Historical Data

3.6.1.1 Data Collection

The condition data used in this case study was collected from the US LTPP database (www.infopave.com) as it is an open access source. Ninety-nine asphalt pavement sections across 20 US states with data available over many consecutive years were selected, providing data ranging from the year 1995 to 2007, totalling 1287 records. Data from the years after 2007 were not used because firstly some of the sections only collected data until 2007 which means no further data after this point, and secondly, there were multiple sections where data

was only reported 5 or 6 years after 2007, resulting in a huge gap in data, which reduced the feasibility for deterioration modelling. The LTPP database provides data in various categories such as traffic, pavement distress condition, pavement structure, material, as well as ambient environmental conditions.

3.6.1.2 Maintenance Assumptions and Data Pre-processing

This case study focuses on asphalt pavement as the most common surface type with survey condition data available for as many consecutive years as possible. Rutting and IRI data were collected as the parameters of interest for prediction. The age of the pavements ranges from newly constructed to 55-year-old pavements. The geographical locations of the sections are diverse across the whole US. Figure 3.5 presents details of the geographical locations of the pavement sections and the number of sections in each state.

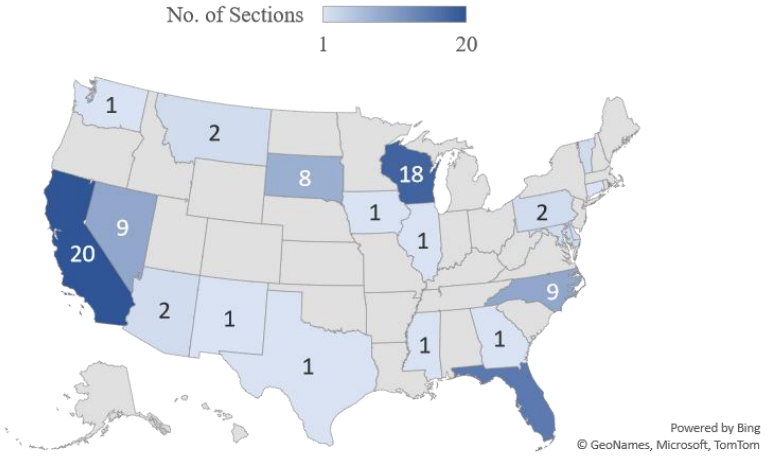


Figure 3.5. Locations and number of selected road sections from the US LTPP database

Given the LTPP database provides detailed information on the maintenance history for each pavement section, the years with relevant maintenance records were removed and it was assumed that the road condition after a maintenance treatment such as ‘Overlay’ or ‘Surface Treatment’ would be restored to that of a newly constructed pavement. After examination of the data, the assumptions made for rutting and IRI values are summarised below:

1) Rutting was set to 0 mm in the year following the relevant maintenance activity when either Overlay or Surface Treatment was applied.
2) IRI was set to 0.7 m/km (assuming it represents the construction condition) when an Overlay treatment was applied.
3) IRI was reduced 60% of the way to the perfect condition (0.7 m/km) value when a Surface Treatment was applied. The equation to calculate the IRI post maintenance is shown in Eq. 3-1 where IRI_p is IRI after the treatment whereas IRI_b is the IRI value prior to the treatment.

$$IRI_p = IRI_b - ((IRI_b - 0.7) * 0.6) \quad \text{Eq. 3-1}$$

Afterwards, to further prepare the data for the model development stage, it was reformatted to obtain the Rutting and IRI value for the next year as a separate column to serve as the output to be fed into the ML process. A total of 1,152 pavement annual data records were extracted. Once the data for selected variables was obtained, initial assessment and evaluation of the data was performed to improve its quality. Despite the completeness and comprehensiveness of the collected data, the raw data for all 99 sections from the LTPP database still suffered from several data quality issues such as missing values for certain years, duplicates, general noise, and anomalies with unreasonable data fluctuations, potentially due to measurement and human errors. For example, rutting condition sometimes slightly improves, e.g., by 1 mm, over time for some road sections without any reported relevant maintenance activities. To address these issues, multiple data pre-processing techniques, recommended by (Kargah-Ostadi et al., 2019), were performed and a Python script was written for automatic data pre-processing and cleaning (see Appendix A). The issues considered and corresponding techniques are described in more detail in Table 3.1.

Table 3.1. Data issues and applied pre-processing techniques

Data issues	Impacted variables	Pre-processing technique used
Multiple entries for the same year	Rutting, All types of cracking (Longitudinal, Transverse, Fatigue)	Mean value
Missing data for some years	Rutting, All types of cracking (Longitudinal, Transverse, Fatigue), AADT (ESALs)	Moving average Curve fitting using the least-squares method Spline interpolation method
Unreasonable data	Rutting, All types of cracking (Longitudinal, Transverse, Fatigue), AADT (ESALs)	Outlier detection based on mean and standard deviation Moving average Curve fitting using the least-squares method Spline interpolation method
General data noise	Rutting, All types of cracking (Longitudinal, Transverse, Fatigue), AADT (ESALs)	Moving average Curve fitting using the least-squares method

Various data processing techniques were used in this study with different purposes. Moving average and curve fitting, using the least-squares method, were used to reduce the noise in the data and mitigate the potential measurement errors in the collected data (Dayananda et al., 2023). In addition, the spline interpolation method was used to ensure the completeness and the smoothness of the whole dataset (Taavitsainen, 2009). Data was stored and processed in the DataFrame which is a 2-dimensional data structure provided by Pandas data analysis library (McKinney et al, 2010). A detailed data pre-processing step flowchart, with rutting variable condition data as an example, is described in Figure 3.6.

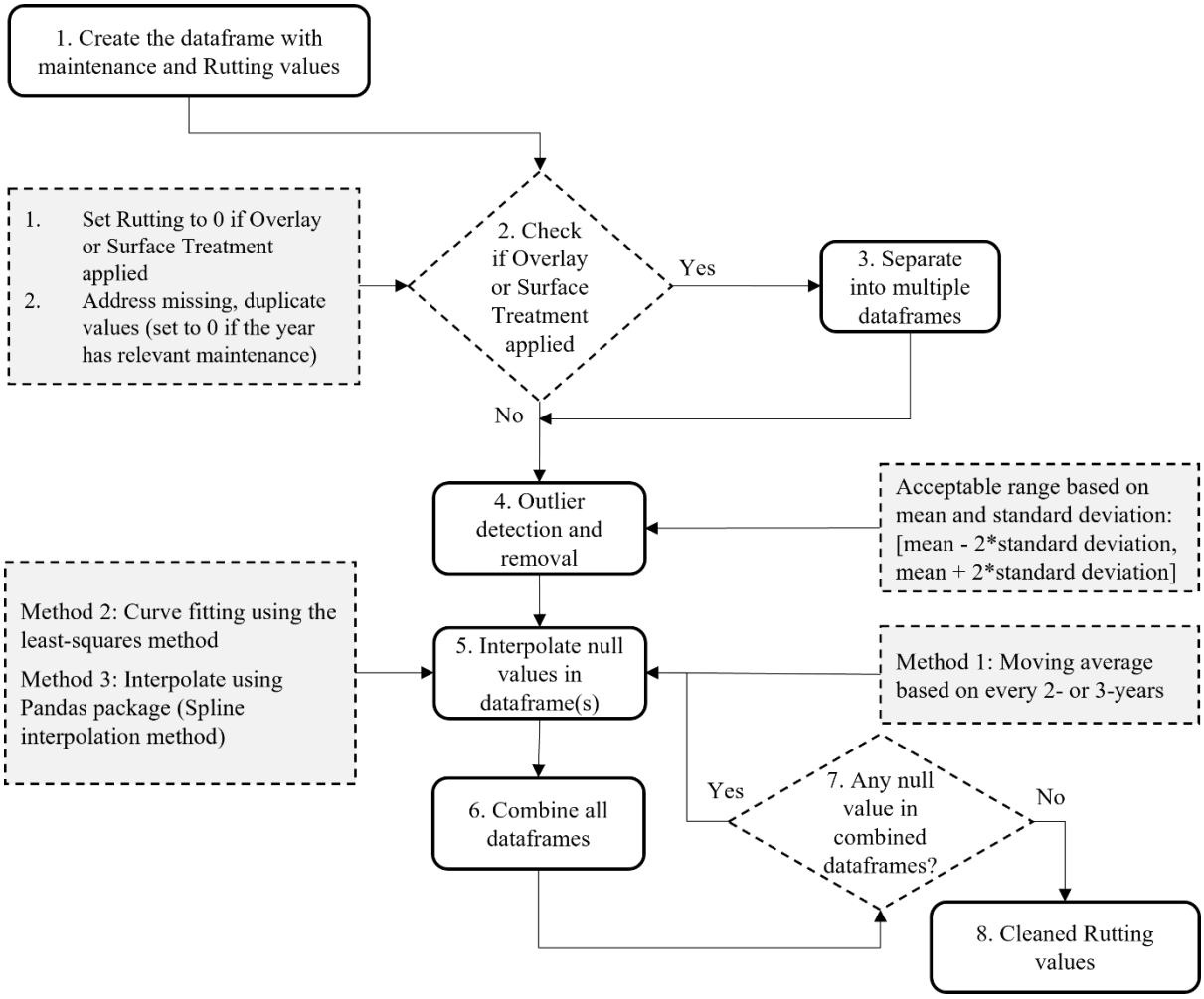


Figure 3.6. Data pre-processing procedures on rutting condition data

It is worth reiterating that, as shown in Figure 3.6, it was assumed that road condition after a relevant maintenance treatment such as “Overlay” or “Surface Treatment” would be restored to the same service level as a newly constructed pavement in its first year in service. In other words, it has been considered that a full restoration of road defects such as rutting and longitudinal cracking occurred from the year when there is a relevant maintenance treatment.

3.6.1.3 Cleaned Data Description

The data collected provides information in the following categories: road condition, road structure, ambient climate, traffic, road material, as well as relevant maintenance activities. Table 3.2 lists the details of the features in each category.

Table 3.2. Descriptions of the features collected in this case study

Category	Feature (Unit)	Details			
		Range	**Mean**	**Median**	**Standard deviation**
Condition	Years from construction	0 - 55	27.2	30	10.07
	IRI (t) (m/km) *	0.4 - 4.7	1.1	1.0	0.5
	Rutting (t) (mm) *	0 - 17	4.2	3.8	2.6
	Longitudinal cracking length (m/section length)	0 - 123.2	5.05	0	14.5
	Transverse Cracking (Count)	0 - 220	21	8	32
	Fatigue Cracking (m^2)	0 - 816.6	26.7	0	80.6
	IRI (t+1) (m/km) *	0.4 - 5.3	1.2	1	0.6
	Rutting (t+1) (m/km) *	0.5 - 17.2	4.5	4	2.6
Structure	Unbound foundation thickness (mm)	0 - 942.3	369.2	365.8	260.6
	Foundation + Asphalt thickness (mm)	185.4 - 1282.7	714.7	744.2	261.7
	Total asphalt thickness (mm)	0 - 505.5	151.2	147.3	76.3
	Dense graded asphalt thickness (mm)	0 - 502.9	131.6	119.4	73.9
	Open graded asphalt thickness (mm)	0 - 33	1.06	0	4.7
	Recycled asphalt thickness (mm)	0 - 167.6	15.8	0	38.1
	Emulsion-based sealing thickness (mm)	0 - 55.9	2.0	0	7.4
	Total foundation thickness (mm)	121.9 - 1135.4	563.4	602	256.8
	Bound foundation thickness (mm)	0 - 726.4	194.2	149.9	168.2
	Number of foundation layers	1 - 4	2	2	1
	Number of asphalt layers	1 - 11	4	4	2
Climate	Annual average ambient temperature (˚C)	4.8 - 24	13.9	12.6	5.9
	Annual average precipitation (mm)	11.5 - 2070.4	797.9	845.1	500.5

	Annual average humidity (%)	51 - 78 %	65.4	66.5	6.0
Traffic	Annual Average Daily Traffic (Equivalent Single Axle Load) - AADT (ESALs)	9 - 2405	585	405	531
Material	Contains geotextile (yes or no)	52 records in 4 sections with geotextile used			
Relevant Maintenance records	Overlay	Applied 40 times across 99 sections from 1995 to 2007			
	Surface treatment	Applied 13 times across 99 sections from 1995 to 2007			

* t refers to the year the data was collected; t+1 means the following year after the year the data was collected

3.6.1.4 Feature Selection

Feature election was necessary to identify the optimal subset of input features from the initial list of model inputs presented in Table 3.2. There are three main approaches in conducting variable selection, namely: wrapper, filter, and embedded methods (Thenmozhi and Helen, 2022). In this case study, an exhaustive variable selection method was adopted since it allows for an unambiguous understanding of the effectiveness of every single included variable and it concentrates on retrieving all possible combinations of the model inputs and gives priority to create a subset of inputs based on the performance quality of an algorithm (Deeba et al., 2018), linear regression in this case which requires the least time and computing resources.

Despite being a computationally expensive method, considering the relatively small total number of variables, and the fact that there is a much stronger emphasis and need to understand the optimal variables as model inputs, the wrapper method was selected. This is to ensure better comprehension and interpretation of the factors impacting pavement rutting development.

All combinations of 21 initial variables available from Table 3.2 were tested as part of the exhaustive variable selection method. Maintenance treatment was not selected as a variable, i.e., input, because roads after relevant maintenance treatment were treated as new sections. Mlxtend library (Raschka, 2018) was implemented to perform exhaustive feature selection that included all possible combinations of the 21 model input interactions with input number ranging from 1 to 21, to build a linear regression model and then compare the models' performances to select the one that results in the best performance (e.g., the least mean squared error). In this study, the number of input variables is 21 which results in 2097151 ($2^{21} - 1$) numbers of combinations, which means 2097151 linear regression models have been built as part of the variable selection process. Python code is provided in Appendix B.

Each of the 2097151 linear models has been evaluated through a 10-fold cross validation approach which calculated the average mean squared error of the model's performance from all iterations. Table 3.3 and 3.4 reports the best five variable combinations that yielded the least average mean squared error across the 10 cross validations when modelling Rutting (t+1) and IRI (t+1). The negated mean squared error is simply the negated value of the mean squared error, and this is due to a convention in the Scikit-learn ML software package (Pedregosa et al., 2011) where higher return values are better than lower return values.

Table 3.3. Top five exhaustive variable selection results for Rutting (t+1)

Rank	Common Variables	Extra variables specific to a particular model	Average negated mean squared error score
1	Rutting (t) (mm)**;** Unbound foundation thickness (mm); Contains geotextile (yes or no); Longitudinal cracking length (m)**;** Annual average humidity (%); Number of asphalt layers; Emulsion-based sealing thickness (mm); Number of foundation layers.	N/A	-0.56826
2		Fatigue (m^2)	-0.568703
3		Annual average ambient temperature (°C)	-0.568832
4		Annual average precipitation (mm)	-0.569025
5		Annual average precipitation (mm); Fatigue (m^2)	-0.569451

Table 3.4. Top five exhaustive variable selection results for IRI (t+1)

Rank	Common Variables	Extra variables specific to a particular model	Average negated mean squared error score
1	Annual Average Daily Traffic (Equivalent Single Axle Load) - AADT (ESALs); Annual average ambient temperature (°C); Annual average precipitation (mm); Years from construction; Transverse Cracking (Count);	N/A	-0.0168609
2		Fatigue (m^2)	-0.0168669
3		Annual average ambient	-0.0168673

	Fatigue (m2); IRI (t); Recycled asphalt thickness (mm); Emulsion-based sealing thickness (mm); Bound foundation thickness (mm)	temperature (°C)	
4		Annual average precipitation (mm)	-0.0168673
5		Annual average precipitation (mm); Fatigue (m^2)	-0.0168701

In Table 3.3, the results after exhaustive variable selection did not include any traffic volume or load which have been commonly considered as direct causes of rutting; the reason could be that the traffic is heavily related to the road design with the number of foundation and asphalt layers, thicknesses, and the quality of the materials used. Regarding environmental factors, the results may indicate that the impact of ambient temperature and precipitation could be covered by humidity.

The first results with the combination of variables that produced the highest negated mean squared error (i.e., the least mean squared error) from each table were selected to give the variables for ML model development for Rutting and IRI.

3.6.1.5 Data from FE Physics-Based Models

Extra data was obtained using a physics-based model. For each pavement section, a two-dimensional multi-layer linear elastic FE model was developed using the data provided in the LTPP database together with assumed material properties of the layers based on various sources including engineers' experience. The FE numerical model was built using the software Abaqus. It is a commonly used engineering simulation software suite based on finite element method, it possesses robust computing function and extensive simulated performance, as well as providing a huge number of multiple element models, material models and analytic processes (Kong and Yuan, 2010).

a. Finite Element (Abaqus) Pavement Model

Modelling was performed on the following modules in Abaqus: Part, Property, Assembly, Step, Interaction, Load, Mesh, Job, Visualization, Sketch. Meshing is an important process in which the Abaqus model solves the differential equations of the system models, by discretising the model to smaller nodes and elements (Smith, 2009) to avoid huge computational time and resource. Young's modulus and Poisson's ratio have been defined as the elastic properties of the materials in different layers in this study.

Ninety-nine Abaqus models, corresponding to the number of considered road sections, were created according to the specific pavement structures in the relevant test section. The cross section of each pavement test section has been modelled with a 3.66 meters width. The pavement model structure for Section 12-0566 is shown in Figure 3.7 (a) as an example. Boundary conditions and mesh techniques were consistent across all sections. Vertical displacements are the results produced from the models. Figure 3.7 (b). displays a) loading area and boundary conditions, b) meshing of layers and c) results of the vertical displacement from the finite element model for Section 12-0566. All material properties (i.e., Young's Modulus and Poisson's Ratio) have been defined in Table 3.5 based on pavement engineering expert opinion.

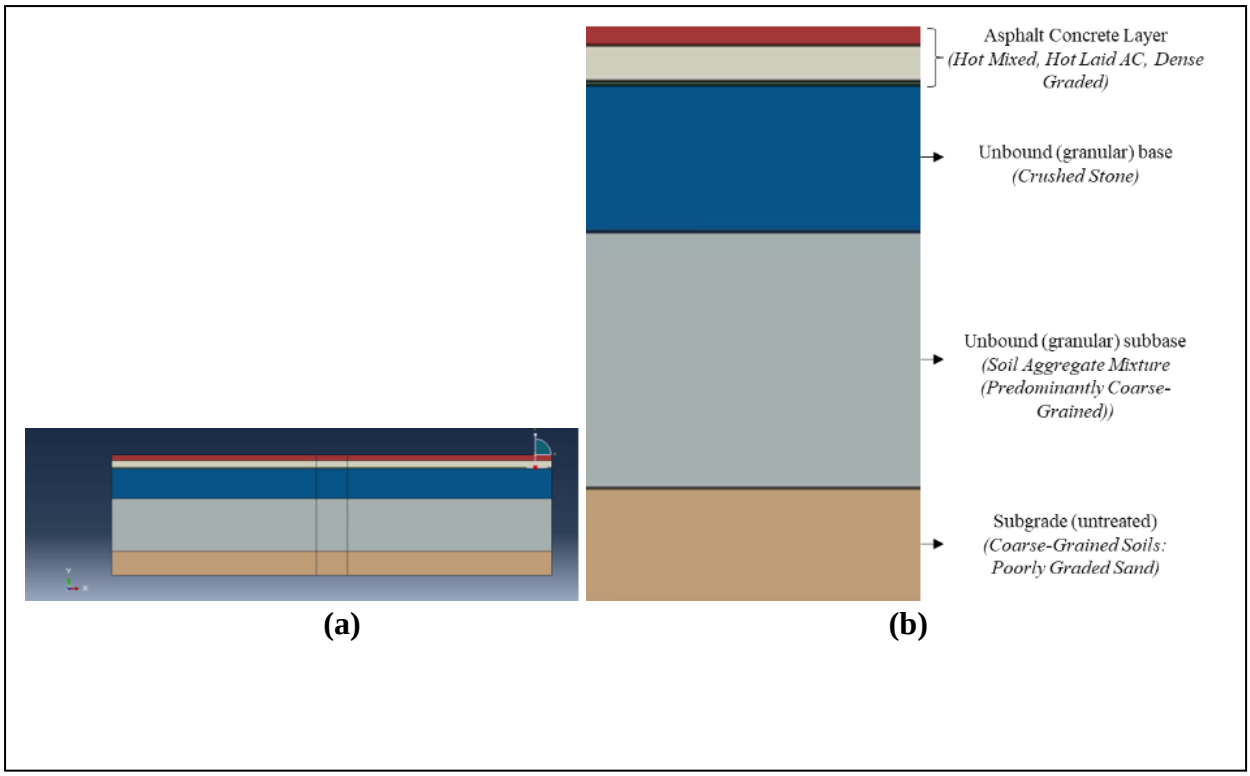


Figure 3.7. a) Pavement model and b) Its structural layers

b. Material Codes and Characteristics in US LTPP Database

Table 3.5 presents the estimated Young's modulus and Poisson's ratio values assigned to each material property types based on the description from the LTPP database.

Table 3.5. Material codes and characteristics

Material code	Material Code Description	Young's modulus (GPa)	Poisson's ratio
1	1-Hot Mixed, Hot Laid AC, Dense Graded	5	0.3
2	2-Hot Mixed, Hot Laid AC, Open Graded	3.5	0.3
4	4-Portland Cement Concrete (Jointed Plain Concrete Pavement)	41	0.2
5	5-Portland Cement Concrete (Jointed Reinforced Concrete Pavement)	41	0.2
6	6-Portland Cement Concrete (Continuously Reinforced Concrete Pavement)	41	0.2

9	9-Plant Mix (Emulsified Asphalt) Material, Cold Laid	2.5	0.3
13	13-Recycled AC, Hot Laid, Central Plant Mix	3.5	0.3
14	14-Recycled AC, Cold Laid, Central Plant Mix	3	0.3
20	20-Other	3.5	0.3
71	71-Chip Seal	Same properties as the layer below it	
72	72-Slurry Seal		
73	73-Fog Seal		
74	74-Woven Geotextile		
75	75-Nonwoven Geotextile		
77	77-Stress Absorbing Membrane Interlayer	1	0.35
81	81-Chip Seal with Modified Binder	Same properties as the layer below it	
82	82-Sand Seal		
83	83-Asphalt-Rubber Seal Coat		
102	102-Fine-Grained Soils: Lean Inorganic Clay	0.2	0.3
108	108-Fine-Grained Soils: Lean Clay with Sand	0.1	0.2
109	109-Fine-Grained Soils: Fat Clay with Sand	0.2	0.2
111	111-Fine-Grained Soils: Gravelly Lean Clay	0.13	0.2
114	114-Fine-Grained Soils: Sandy Lean Clay	0.2	0.3
117	117-Fine-Grained Soils: Gravelly Lean Clay with Sand	0.11	0.3
131	131-Fine-Grained Soils: Silty Clay	0.1	0.3
135	135-Fine-Grained Soils: Sandy Silty Clay	0.2	0.3
141	141-Fine-Grained Soils: Silt	0.2	0.3
142	142-Fine-Grained Soils: Silt with Gravel	0.2	0.2
144	144-Fine-Grained Soils: Gravelly Silt	0.2	0.2
145	145-Fine-Grained Soils: Sandy Silt	0.2	0.2
201	201-Coarse-Grained Soils: Sand	0.13	0.3
202	202-Coarse-Grained Soils: Poorly Graded Sand	0.12	0.45
204	204-Coarse-Grained Soils: Poorly Graded Sand with Silt	0.2	0.3

211	211-Coarse-Grained Soils: Well-Graded Sand with Silt and Gravel	0.2	0.25
214	214-Coarse-Grained Soil: Silty Sand	0.15	0.42
215	215-Coarse-Grained Soil: Silty Sand with Gravel	0.2	0.4
216	216-Coarse-Grained Soil: Clayey Sand	0.2	0.3
217	217-Coarse-Grained Soil: Clayey Sand with Gravel	0.2	0.15
265	265-Coarse-Grained Soil: Silty Gravel with Sand	0.2	0.4
266	266-Coarse-Grained Soil: Clayey Gravel	0.2	0.3
267	267-Coarse-Grained Soil: Clayey Gravel with Sand	0.2	0.35
302	302-Gravel (Uncrushed)	0.25	0.35
303	303-Crushed Stone	0.25	0.35
304	304-Crushed Gravel	0.25	0.35
306	306-Sand	0.08	0.2
307	307-Soil-Aggregate Mixture (Predominantly Fine-Grained)	0.2	0.35
308	308-Soil-Aggregate Mixture (Predominantly Coarse-Grained)	0.2	0.35
309	309-Fine-Grained Soils	0.2	0.2
310	310-Other (Specify, if Possible)	0.2	0.3
319	319-HMAC	5	0.3
320	320-Sand Asphalt	2	0.3
321	321-Asphalt Treated Mixture	2	0.3
325	325-Open Graded, Hot Laid, Central Plant Mix	3.5	0.3
331	331-Cement Aggregate Mixture	20	0.1
338	338-Lime-Treated Soil	0.2	0.13
339	339-Soil Cement	1	0.13

The simulation across each section generated an elastic surface deflection under 40 kN load per meter traffic loading as additional data. The elastic deflection result represents the physical stiffness of the pavement – a property which would be expected to influence the development of Rutting and IRI. Deflection has been considered as one of the variables that

would affect rut depth and therefore used as part of rutting prediction in multiple existing highway management tools and pavement research. For example, in HDM-4 manual (Kerali and Odoki, 2006), Benkelman beam deflection is one of the input variables for rutting calculation due to initial densification. In addition, rutting models developed based on data from AASHO Road Test also include deflection as one of the inputs (Suh and Cho, 2014). The equations for both are presented Eq. 3-2 and Eq. 3-3.

$$\text{RDO} = K_{rid}\left[a_0(YE4 * 10^6)^{(a_1+a_2 DEF)} SNP^{a_3} COMP^{a_4}\right] \quad \text{Eq. 3-2}$$

Where RDO is the rutting due to initial densification, YE4 is the annual number of traffic, DEF is the average annual Benkelman beam deflection, SNP is the average annual adjusted structural number of the pavement, COMP is the relative compaction and K_{rid} is the calibration factor for initial densification.

$$logRP = -5.617 + 4.343 logd - 0.167 \log(N18) - 1.118 log\sigma_c \quad \text{Eq. 3-3}$$

Where RP is the rutting rate per axle repetition, d is the surface deflection under a load of 40-kN, σ_c is the vertical compressive stress at the asphalt-based interface, and $N18$ is the number of 80-kN single axle repetitions.

Furthermore, other studies which used accelerated load test, low-volume roads, as well as in-service flexible pavements have also developed rutting prediction models, highlighting the strong correlation between deflection and rutting (Bae et al., 2000; Wiman, 2008; Alaswadko and Hassan, 2018).

At the same time, according to two generalised roughness progression model in flexible pavements developed by HDM-III (Paterson and Attoh-Okine, 1992) where the standard deviation of rut depths has been used as a function of IRI calculation. In addition, multiple studies have demonstrated that IRI has significant relationship between the pavement

distresses and especially rutting (Mubaraki, 2016; Kırbaş, 2018; Joni et al., 2020) . Hence, the simulated deflection has been used as an extra data source for the prediction model development for rutting and IRI in following sections.

In this study, the numerical models were created using identical procedures in each case, including the loading area, and the mesh generation mechanism. Table 3.6 describes the assumptions made in the creation of these physics-based models.

Table 3.6. Assumptions made in the creation of Abaqus models

No.	Assumptions
1	The load applied is 40kN per meter in the third dimension
2	The thickness of the subgrade layer is 200mm
3	Material proprieties stay constant over the years

3.6.1.6 Model Selection and Preparation

In this case study, apart from the standard purely data-driven ML modelling, two different approaches were used to combine the physics-based FE simulation output with ML development process.

For rutting prediction, the output from the physics-based model was used as an extra input in the ML process whereas in IRI prediction, not only was the FE simulation output added as an additional input to ML, but it was also used to define a physics-informed loss function to guide the training process of the ML algorithm. More details are provided in following sub-sections.

To select a suitable ML algorithm for rutting prediction, an initial assessment on the performance of the ML models introduced in Section 3.5.1 was performed on the collected data. The whole dataset was split as (70%, 15%, 15%) for train, validation and test sets,

respectively (Li and Chan, 2017; Salehi et al., 2020; Liu et al., 2021) based on the R^2 and RMSE shown in Eq. 3-4, and Eq. 3-5 respectively.

$$R^2 = 1 - \frac{\sum_{i=1}(\hat{y}_i - y_i)^2}{\sum_{i=1}(\hat{y}_i - \bar{y}_i)^2} \quad \text{Eq. 3-4}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \quad \text{Eq. 3-5}$$

where $\hat{y}_i$ is the predicted value from the ML model, y_i is the actual observed value, n is the total number of observations, and $\bar{y}_i$ is the average of the measured values.

The ML algorithms attempted were RF, ANN, KNN and SVM with suggested hyperparameters as recommended by (Bashar and Torres-Machi, 2021; Marcelino et al., 2021; Cano-Ortiz et al., 2022; Xu and Zhang, 2022). Table 3.7 shows the results for each ML technique on various data sets.

Table 3.7. Results for multiple ML algorithms

ML Algorithms	Training set (70%)	Validation set (15%)	Testing set (15%)
ANN	R^2 = 97.3% RMSE = 0.440	R^2 = 93.8% RMSE = 0.707	R^2 = 86.4% RMSE = 0.824
SVM	R^2 = 81.6% RMSE = 1.147	R^2 = 75.6% RMSE = 1.403	R^2 = 45.7% RMSE = 1.647
KNN	R^2 = 82.7% RMSE = 1.115	R^2 = 81% RMSE = 1.236	R^2 = 68.2% RMSE = 1.261
RF	R^2 = 99.1% RMSE = 0.255	R^2 = 93.9% RMSE = 0.704	R^2 = 89.7% RMSE = 0.717

Therefore, RF was selected because of its superior in prediction accuracy to predict rutting. For the case of IRI, ANN was preferred due to not only its capacity where one can tune the network architecture to fit the corresponding data (Haber and Ruthotto, 2017) but also the customised loss function can be defined to especially facilitate the integration of physics into ML training process, which also has been identified to be the most suitable ML approach for a road digital twin according to a systematic literature review reported by Chen et al. (2022). The following describes the details the model preparation combining physics with ML.

Physics-Enhanced Neural Network (PENN)

There are two steps in the proposed PENN framework: 1) create hybrid combinations of physics-based models and NNs where the physics-based model output is used as an extra input feeding into the NN. This has also been termed a hybrid physics data (HPD) model (Daw et al., 2022); (2) integrate existing scientific knowledge to customise and constrain the loss function in the NN during the learning process (Raymond and Camarillo, 2021; Gallup et al., 2023).

1. Building HPD Models

After the feature selection process, a NN model could be trained using the list of selected features. This approach is purely data driven despite the target feature being physically related to the input features. At the same time, physics-based simulation models could be used to obtain a simulated value of the target feature given the input information.

In this study, physics-based models were developed to simulate the deflection under a constant load for all sections to understand the stiffness of the pavement sections. The correlation between a pavement's IRI and its deflection has been made evident from empirical studies, for example a correlation has been considered in the HDM-4 model (Kerali and Odoki, 2006). However, the calculation of a physics-based model tends to be time-consuming, and it may not reflect the complete physics of the problem given the limitation of simulating the effect of all the input parameters. Therefore, in this step, two models (the NN model and the physics-based model) were combined to complement each other to leverage the information from both data and physics. In particular, the deflection output from the physics-based FE simulation model was added into the training process of the NN as an extra input. This enables the modelling process to extract information from existing input features and to overcome any systematic bias experienced in the simulation models.

2. Using Physics-Informed Loss Functions

As previously stated, the standard training process of an ANN involves the calculation of a loss function which is the error between the model's predicted output and the actual observed value. The aim of training during the backpropagation process is to reduce this error (the closer to zero the better). When it comes to a physics-informed loss function, if there are existing theories or theoretical governing equations between the inputs and the output available, then in addition to the data-based loss function, an extra loss function which describes this physical relationship could be added to constrain the prediction so that it obeys certain constitutive equations or partial differential equations (Cuomo et al., 2022). However, given the complexities of pavement structure, material types and climate, it is challenging to establish any explicit relationship between the input and output features for a road section, especially when considering the lack of accepted theories as well as the number of factors that could have effect at different levels. Therefore, instead of using partial differential equations to constrain the range of prediction output given a set of inputs, a different approach, similar to Deng et al. (2024) was adopted to integrate physical information based on relationships between input and output features. The process includes the follow steps:

- Step 1: Traditional ANN training on the original data collected
- Step 2: Generation of up-sampled data that expressing an explicit monotonic relationship between one input and the output
- Step 3: Further training the ANN model on the up-sampled data with a customised loss function based on physics to constrain the model's prediction

In Step 2, given that the data was collected from multiple geographical locations with different climates and materials, it was not realistic to identify a clear input that would have a monotonic

influence on the output. However, once the simulated deflection for each pavement section has been obtained, a generic physical understanding of the level of stiffness for each section is achieved. This could be used to form a monotonic relationship. As illustrated in Table 3.8 the key difference is that the up-sampled data input vector with simulated deflection value has been created by sorting the original values in an ascending order while the other columns were kept constant at their mean values, (except the columns representing features used in the simulation, e.g., number of pavement layers and layer thickness). The up-sampled data input vector allows for the examination of the isolated effect of the monotonic increase of the level of stiffness on IRI across multiple sections.

Table 3.8. Up-sampled data generation process, blue shade represents the original data, and the grey shade shows the generated additional up-sampled data

Original data	IRI (t+1)	Input 1	Input 2	Input 3	…	Input N	Simulated deflection
	…	…	…	…	…	…	…
	…	…	…	…	…	…	…
	…	…	…	…	…	…	…
Up-sampled data	IRI (t+1)	Input 1	Input 2	Input 3	…	Input N	Simulated deflection (sorted)
	low	Averaged	Averaged	Averaged	Averaged	Averaged*	low
	high						high

* Average not applied if this column is a parameter used in the physics-based simulation

In Step 3 after the preparation of the up-sampled data, the trained model from Step 1 is re-trained on the generated up-sampled data with the usage of a customised physics-informed loss function, which was defined using a rectified linear unit activation function in this study to ensure a physically consistent prediction outputs that also meet the condition of being in an ascending order, as shown in Eq. 3-6. Thus, the overall loss function including data loss and physics-informed loss was updated as in Eq. 3-7.

$$\mathcal{L}_{P,SD} = \sum_{i=1}^{K}(\max(0, -\Delta\tilde{y}_i))^2 \quad \text{Eq. 3-6}$$

$$\mathcal{L}_{Total} = \mathcal{L}_D + \lambda(\mathcal{L}_{P,SD}) \quad \text{Eq. 3-7}$$

$$\mathcal{L}_D = \frac{1}{K}\sum_1^K |(y_i - \tilde{y}_i)| \quad \text{Eq. 3-8}$$

where $\tilde{y}_i$ is the predicted output vector and y_i is the actual/measured output; K is the number of samples; $\mathcal{L}_D$ expressed in Eq. 8 is the data-based loss function, MAE that is mean absolute error, measuring the paired error between actual and predicted values; $\mathcal{L}_{P,SD}$ is the physics-informed loss leveraging the simulated deflection output; λ is a tuning parameter to decide the relative weight of importance between data loss and physics loss. $\mathcal{L}_{Total}$ is simply the addition of data-based loss and physics-informed loss.

As shown in Table 3.8, given the input simulated deflection in the up-sampled data has been sorted from low to high, it is expected that the predicted output vector - IRI (t+1) should be also in the same ascending order when the model is re-trained in Step 3. Any portion in the prediction that does not obey this constraint is handled as an error and penalised during the model training process.

To clearly compare the model's performances, different scenarios leveraging different methods to combine physics-based model with ML have been summarised in Table 3.9 with the corresponding ML algorithm and model outputs.

Table 3.9. Scenarios considered in this study for predictions

Scenario	Description	RF Prediction for Rutting	ANN Prediction for IRI
1	ML based on input data from feature selection result	√	√
2	Simulated deflection integrated as an extra input for ML	√	√
3	Simulated deflection integrated as an extra input for ML + the use of up sampled data with a physics-informed loss function		√

These three scenarios were then defined for model development and evaluation.

3.6.1.7 Model Development and Evaluation on Short-term Predictions

The Scikit-learn ML package (Pedregosa et al., 2011)was used for developing RF regression models to predict Rutting. The Grid Search method (Liu et al., 2020) was used for tuning the hyperparameters (see Appendix C) used for the RF regression model for both scenarios considering its exhaustive searching method that could find the optimal hyper-parameter values by checking all parameter combinations. The RF hyperparameter space was defined and the final configurations are summarised in Table 3.10.

Table 3.10. RF hyperparameter space and tuning results

Hyperparameters	Space	Default	Tuning results (Scenario 1)	Tuning results (Scenario 2)
n_estimators	[50, 100, 150, …, 450]	100	100	50
max_depth	[3, 4, 5, …, 49]	None	5	38
min_samples_split	[2, 4, 6, 8, …, 28]	2	12	18
min_samples_leaf	[1, 2, 5, 10, 50, 100]	1	5	2

To assess the one-year prediction performance of each model, metrics such as R^2 and RMSE were used.

To ensure the model generalisation capacity, out of the 99 road sections 15 sections (approx..15% of the total) were randomly selected and used for the final test of model performance based on suggestions made by Ardila et al. (2019) and Sun et al. (2020). The remaining 84 sections were used to train and validate the model. To avoid overfitting, which is a common issue for ML algorithms (Badillo et al., 2020), a cross-validation technique was applied on the 84 sections with a 10-fold configuration, which is the common configuration practice for cross-validation (Kohavi, 1995). Additionally, during the random selection of training and testing sections, basic statistical analysis, such as mean and standard deviation, on each model variable was performed to ensure the validity of the model by checking the distribution similarity between training and testing data. The model evaluation outcome is the

median value out of the results from 10 cross-validations. Once the model was evaluated, all data from 84 sections were trained to build a model which was tested against the 15 sections to understand the performance of the model. To further reduce the bias and randomness during the selection of the sections used for model training and testing, the whole process was repeated 30 times for both scenarios, each with a different set of 15 test sections, and this was used to provide an indication of the model's generalisation performance capacity.

The number of times the model was run was chosen to be 30 following the recommendation made in other studies with the similar approach (Menzies et al., 2005; Su et al., 2015; El-Gawady et al., 2022) in order to gain enough statistical information on the variance to understand the model uncertainties. Figure 3.8 shows the whole process of the RF model training, cross-validation evaluation, and test.

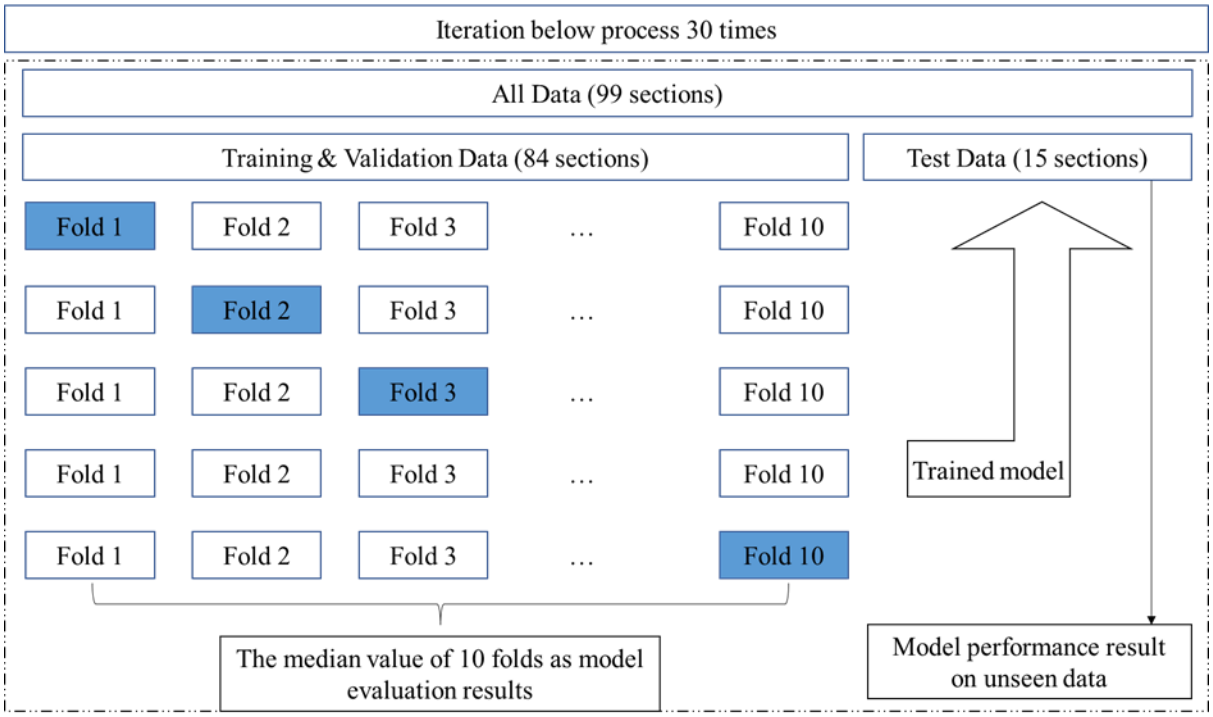


Figure 3.8. RF model training, evaluation, and test iteration process

For ANN modelling for IRI prediction, a similar approach was followed despite the differences in the usage of different ML and hyperparameter optimisation packages.

TensorFlow (Abadi et al., 2016), and Keras ML libraries (Chollet, 2018) were used in Jupyter notebooks for implementing the ANN regression models (see Appendix D). The KerasTuner hyperparameter tuning framework was used with a Bayesian optimization algorithm (O'Malley et al., 2019) for searching the optimal hyperparameters for the ANN model. The ANN hyperparameter space is defined and summarised in Table 3.11.

Table 3.11. ANN hyperparameter space and tuning results

Hyperparameters	Space
Number of layers	[1-10]
Number of neurons each layer	[32-512]
Activation function	ReLU, Linear
Learning rate	[0.01, 0.001, 0.0001]
Loss function	MAE
Optimiser	Adam

To assess the one-year prediction performance of each model, standard statistical measures such as R^2 and RMSE were used. The model development flow is described in Figure 3.9, noting that the process is repeated 50 times. This number was chosen as it has been recommended in other studies with a similar approach (Menzies et al., 2005; Su et al., 2015; El-Gawady et al., 2022). The 50 times iteration was selected to ensure sufficient statistical information to quantify the model uncertainty expressed in the form of variance, for the evaluation of the model's generalisation capacity. It is also worth pointing out that during each cycle of the training and testing data split, it was ensured that the testing data range stays within the range of the data used to train the model to mitigate overfitting issues (Alzabeebee et al., 2018, 2021). The selection of 15 test sections for the development of both ML algorithms, was made based following the procedure described in Figure 3.10 for each iteration to ensure the optimum ML learning experience.

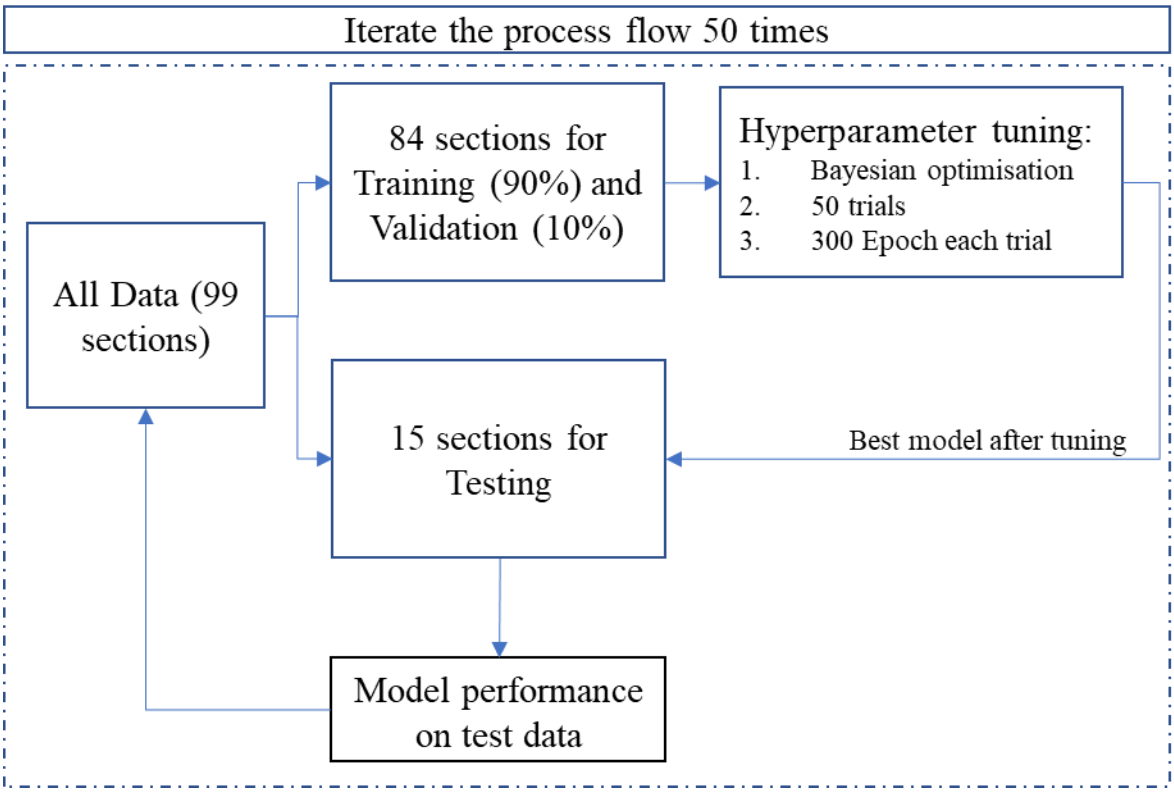


Figure 3.9. Process flow for ANN model training and evaluation

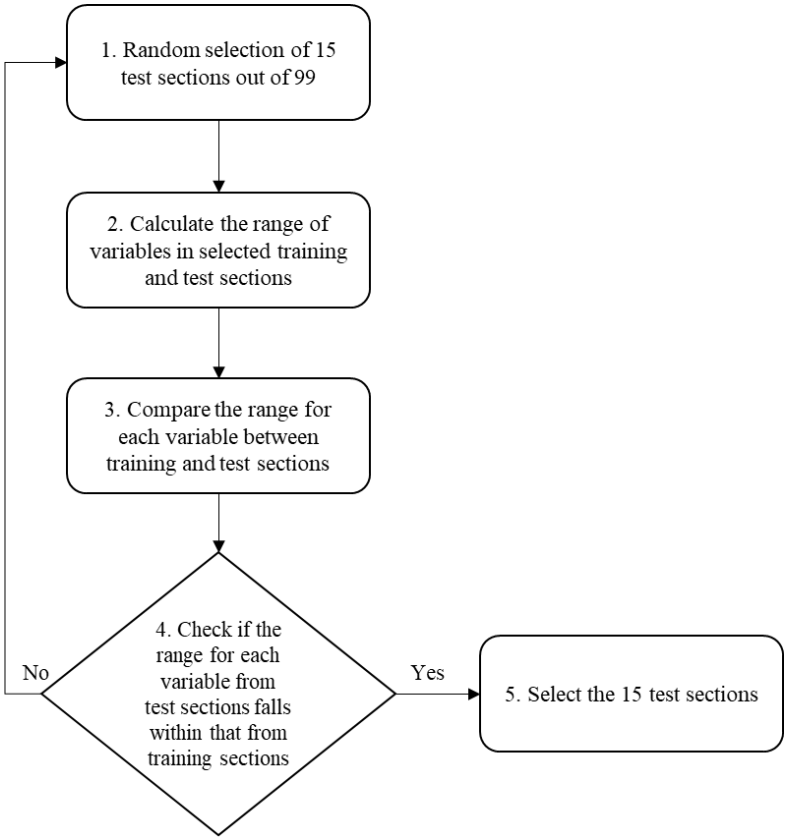


Figure 3.10. Selection process of the 15 test sections

3.6.1.8 Model Uncertainty Quantification

Prediction intervals offer a method to quantify and communicate the probability for a single prediction point which is an estimation or approximation with some level of uncertainty. Prediction intervals can reveal the level of accuracy and confidence of model prediction. The uncertainties and potential errors in general come from two main sources: 1) ineffectiveness and error of the model itself, and 2) the noise contained in the initial measurement data before or even after the data cleaning process. In this study, the uncertainty is expressed through the variance of the ML model due to different sampling of the training and testing data. Common model evaluation methods such as R^2 and RMSE fail to address the prediction confidence for individual instances (Tavazza et al., 2021). In such cases the point estimation is insufficient for the forecasting of the prediction uncertainties. Therefore, prediction intervals were generated in this study to understand the precision and accuracy of the model prediction.

Prediction intervals were calculated to understand the model performance accuracy based on ML model results after 30 repetitions for rutting prediction, and 50 repetitions for IRI. Following the process described in Figures 3.8 and 3.9, a list of predictions from both RF and ANN model results were obtained and compared against the actual results to understand prediction distributions by generating prediction intervals. The generated prediction intervals quantified the uncertainty for one prediction point value by identifying a range of prediction with a certain likelihood. This was done through calculating a linear regression fit of the predicted values and the actual values and the standard deviations of the residuals between linear fit line and the model's predicted values (Kirkwood and Sterne, 2010). A Python script was written for this purpose, see Appendix E.

3.6.1.9 Model Evaluation on Long Term (Multi-year) Predictions

The multi-year predictions were generated by leveraging bootstrapped residuals based on the collection of errors already experienced in the model results for rutting and IRI. This method assumes the future predicted errors would be similar to the existing ones (Tibshirani, 1996; Khosravi et al., 2011). By taking a uniformly distributed sample from the collection of existing sorted residuals following a cumulative distribution function and appending to the model's predicted results, different prediction outcomes were produced. Through doing this 2000 times, a collection of slightly different one-year predictions was generated. By building model inputs based on the predicted output from the previous year, multi-year predictions were then created according to the same process. A Python script was run for 2000 times to generate the ranges with lower and upper bound for multi-year predictions (see Appendix F). The 90% probability prediction intervals for the 2000 results for any single year were then computed to understand the 90^{th} percentile range predictions for that particular year. The results for the 12^{th} year 90% probability prediction intervals, as an example, based on this approach are presented in Chapter 4.

3.6.2 Case Study II – Experiment at UK National Buried Infrastructure Facility (NBIF) Using Sensor Data

During the PhD study, an opportunity arose to collect real-time sensor data through an instrumented experiment conducted at NBIF to simulate an environment where a large amount of data is available and subsequently explore the optimisation of sensor data collection frequency according to the prediction accuracy of the model. This section describes the case study that forms this analysis. The experiment details and produced data are described in the following sections.

3.6.2.1 Set up, Equipment and Sensors

The experiment consisted of a three-layer pavement structure with an actuator on top, to simulate cyclic loading. The simulated pavement consists of a subgrade layer with sand, an unbound subbase layer with UK Type 1 crushed rock as well as a cold-mix asphalt surface. At the middle of the pavement subgrade layer, a 96-cm long pipe was laid to simulate a buried pipe condition. Figures 3.11 - 3.13 show the experiment set up, data collection configuration and the test running in progress, respectively.

Figure 3.11. NBIF experiment set up

Figure 3.12. NBIF data collection

Figure 3.13. NBIF experiment test in progress

In terms of sensors, Linear Variable Differential Transformers (LVDTs), Strain Gauges (SGs), Pressure Cells (PCs) and Temperature sensor probes (Ts) were instrumented inside and outside the pavement. A sketch of the experiment including the actuator and the pavement structure, materials as well as the sensor locations are illustrated in Figure 3.14.

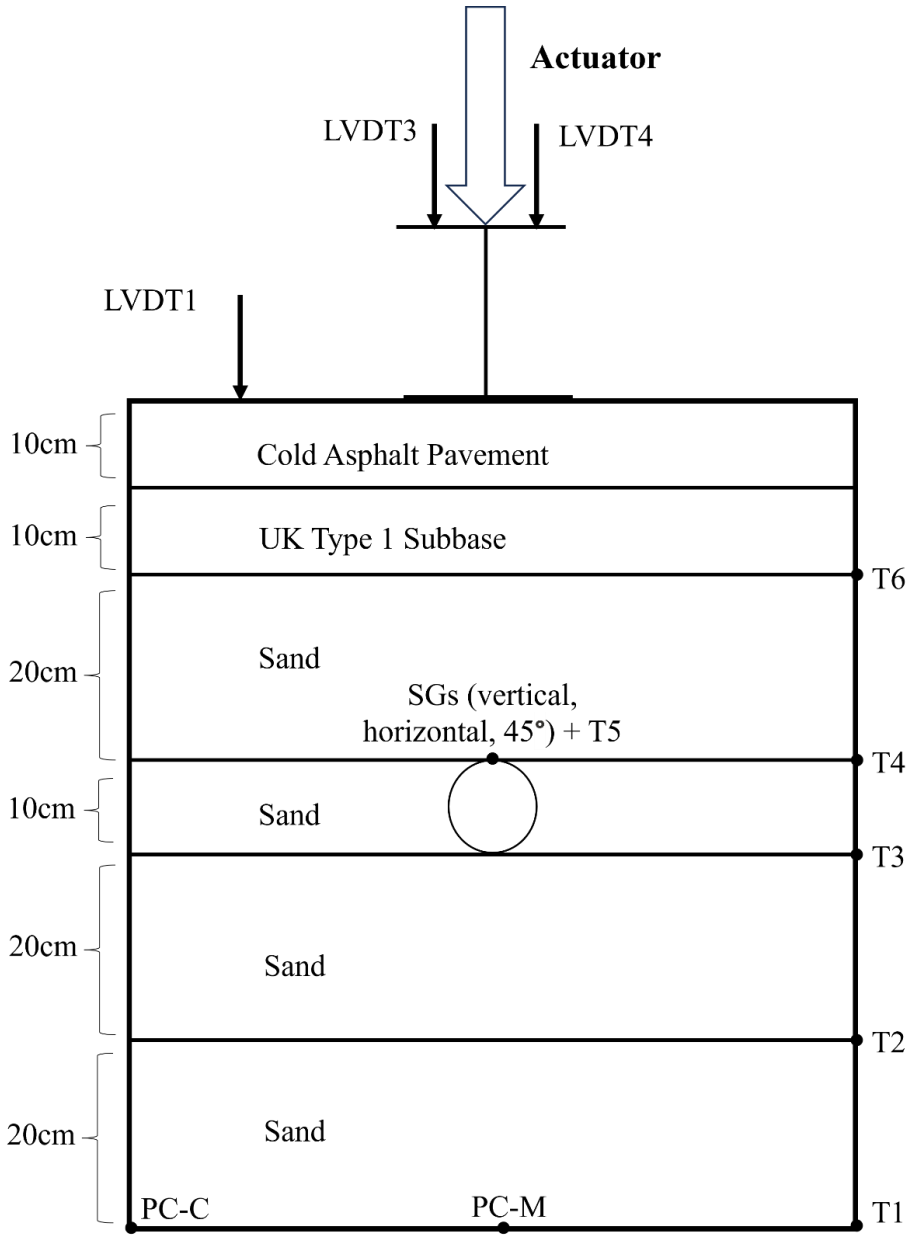


Figure 3.14. A sketch of the actuator, pavement and sensors, including 2 LVDTs on the actuator (LVDT3 & 4) as well as 1 LVDT on the pavement surface (LVDT1) and a PC in the corner (PC-C) and one in the middle (PC-M), six temperature probes (T1-T6), and three SGs on top of the buried pipe

LVDT2 and T7 are not included in the sketch as LVDT2 was located outside the container against the container wall, and T7 was located outside the container to measure laboratory ambient temperature. More details about the sensors the associated measured data, and their frequencies are described in Table 3.12.

Table 3.12. Sensor details and frequencies

Sensor	Equipment/sensor name	Details and Frequencies
Actuator	Cyclic loading actuator	• Applies 20 kN loading with a frequency of 5Hz
LVDT1-4 & SGs	Linear Variable Differential Transformer & Strain Gauges	• LVDT measures the displacement and can be used to obtain the surface permanent settlement due to the cyclic loading. • SG measures the strain on the pipe due to cyclic loading. 3 SGs were placed in different directions on the pipe (vertical, horizontal, and 45 degree) • Both LVDTs and SGs collect data at a frequency of 1613Hz
PC-M & PC-C	Pressure Cells	• Two PCs measure the stress distribution caused by the cyclic loading. Both PCs also embed a temperature sensor that measures the soil temperature • PC-M was located at the central bottom of the container • PC-C was located at the corner bottom of the container • PCs collect data at a frequency of 20Hz
T1-7	Temperature sensors (Thermometers)	• Thermometers measure the soil temperature within the pavement • Temperature sensors 1 – 6 are depicted in the sketch. Temperature sensor 7 was physically located in the laboratory to measure ambient temperature • Temperature sensors collect data with 1 record per minute

3.6.2.2 Experiment and Data Preparation Details

In this case study, the same experiment with the above set up was conducted twice to ensure data generalisability. At least 1 million cycles of loading were applied for each experiment. The details of these two experiments are presented in Table 3.13.

Table 3.13. Experiment details

Experiment Number	Test Date	Duration
Experiment 1	5th Feb 2024	4 hours
	6th Feb 2024	6.5 hours
	7th Feb 2024	8 hours

	8th Feb 2024	7.5 hours
	9th Feb 2024	7 hours
	12th Feb 2024	7 hours
	13th Feb 2024	7 hours
	14th Feb 2024	7.5 hours
	Total	54.5 hours
Experiment 2	11th March 2024	2 hours
	12th March 2024	7.5 hours
	13th March 2024	6 hours
	14th March 2024	7.5 hours
	15th March 2024	5 hours
	18th March 2024	6 hours
	Total	34 hours

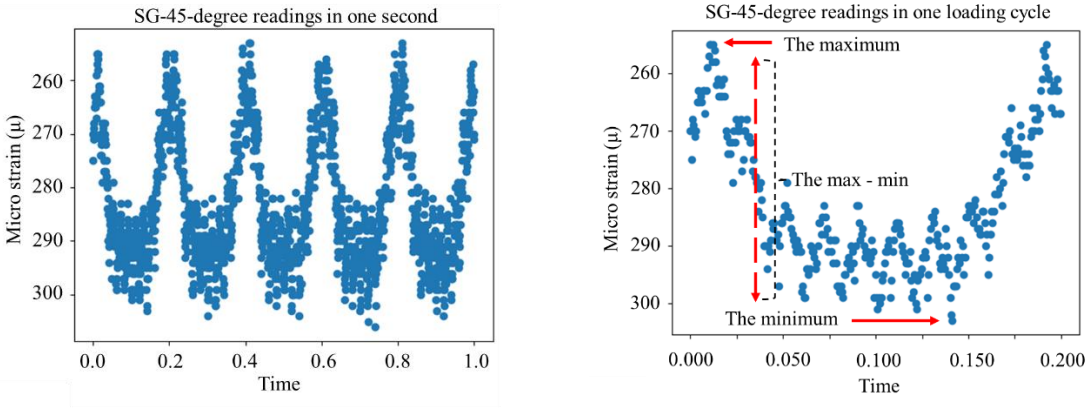
Both experiments provided data on loading, elastic and permanent surface displacement, structural stress and strain, pressure at the bottom of the container, pressure at the middle of the container, temperature within the pavement structure and container, as well as the ambient temperature, based on the sensor configurations. Due to technical issues encountered during both experiments, the strain gauge (vertical direction) became faulty and therefore was not able to produce reliable results. Nevertheless, all the other sensors described in Table 3.12 functioned properly, and data was generated and stored in a shared server on a cloud computing platform – BlueBear facility at the University of Birmingham. As a result, a huge volume of data, around 70 giga byte was produced from different sources and stored for analysis. Python scripts for NBIF data analysis can be found in Appendix G.

Sensor Data Fusion and Explanation

a. In order to fuse the data generated from different sensors to have the same number of readings, given the temperature probe has the least collection frequency amongst all sensors (1 record per minute), a sampling rate aligned with its frequency (every 1 minute) was applied to obtain a subset of data from other sensors' readings. In other words, only the data collected from LVDTs, SGs and PCs within one second period after each minute

was collected for further processing, meaning the data produced within each minute was overlooked as part of this sampling process.

b. Given the high frequency of LVDT and SG samplings, as well as the pressure cells, 1613 and 20 records were generated within one second respectively, with the actuator running with a frequency of 5 Hz, applying a cyclic load. Figure 3.15 (a) visualises how the sensor readings (strain gauge at 45 degrees) under the load appear within a one-second period where 5 cyclic loading was done as shown. Figure 3.15 (b) then presents the sensor readings only within one loading cycle, with arrows showing the specific points of interest. The relevance and purpose of these values are explained in following steps.



(a) (b)

Figure 3.15. SG readings within a) one second and b) one loading cycle

c. Following step b, three distinct values were further extracted from different sensor readings under one loading cycle: 1) the maximum; 2) the minimum; and 3) the maximum - minimum to align and fuse with the data from temperature probe. These were used as parameters for model development. These values for LVDTs, SGs and PCs within one second were also used to determine the elastic and plastic displacement and strain on the pipe, respectively, while the overall aim was to measure the permanent deformation on the pavement surface. A detailed interpretation for each sensor reading after the data fusion process is described in Table 3.14.

Table 3.14. Sensor data and meanings

Sensor readings	Meaning
Minimum LVDTs	Minimum elastic linear displacement/deformation which provides the value for permanent deformation at the end of the experiment
Maximum LVDTs	Maximum elastic linear displacement/deformation
Maximum LVDTs - Minimum LVDTs	Transient displacement or Elastic recoverable deformation
Minimum SGs	Minimum elastic strain which is the value for permanent strain at the end of the experiment
Maximum SGs	Maximum elastic strain
Maximum SGs - Minimum SGs	Transient strain or Elastic recoverable strain
Minimum PCs	Minimum total pressure
Maximum PCs	Maximum total pressure
Maximum PCs - Minimum PCs	Transient total stress or Elastic recoverable total pressure
PCs Temperature	Temperature in the soil
T1-T6	Temperature in the soil
T7	Ambient temperature

To ensure consistency with the first case study (Case study I) in which rutting was predicted, the minimum LVDT3, indicating the pavement permanent deformation was selected as the parameter of interest to be prepared as the output for ML model training. The other data sources were prepared as inputs for model development. Given the similarity in data produced from LVDT3 and LVDT4, the LVDT4 related features were not used as part of the model inputs. In addition, *maximum LVDT3* and '*maximum LVDT3 – minimum LVDT3'* were also not included in the analysis to avoid (due to their potentially) similar data trends.

3.6.2.3 Optimisation of Data Collection Frequency

With the availability of a large amount of data generated with different but high frequencies from multiple sensors in NBIF experiment, this case study aimed to identify a relatively suitable and efficient sampling rate across the sensors that would produce a model with an acceptable prediction accuracy given all the data generated. From a road management perspective, the importance of the prediction accuracy is for the associated cost reduction

considering normally limited available budgets (Hosseini and Smadi, 2021). Within this context, the optimisation of the data collection frequency was also to balance the amount of data needed (data collection interval) and the potential reductions in model's prediction accuracy, considering a range of factors such as the cost of data acquisition systems, and data storage, data processing as well as the power and electricity required for big data cloud-based systems. Essentially, the following research questions were investigated:

1. If we are generating thousands of data records per second, how much data do we need in such a big data environment?
2. To what extent can the data be reduced while producing a sufficiently accurate predictive model?

This case study implemented a ML-based method for optimising data collection frequency. To achieve this, machine learning models are repeatedly trained and evaluated based on the amount of sensor readings. A process flowchart is proposed to demonstrate how to identify the optimal sampling rate within a predictive road digital twin context as shown in Figure 3.16. A detailed discussion for each step is provided afterwards.

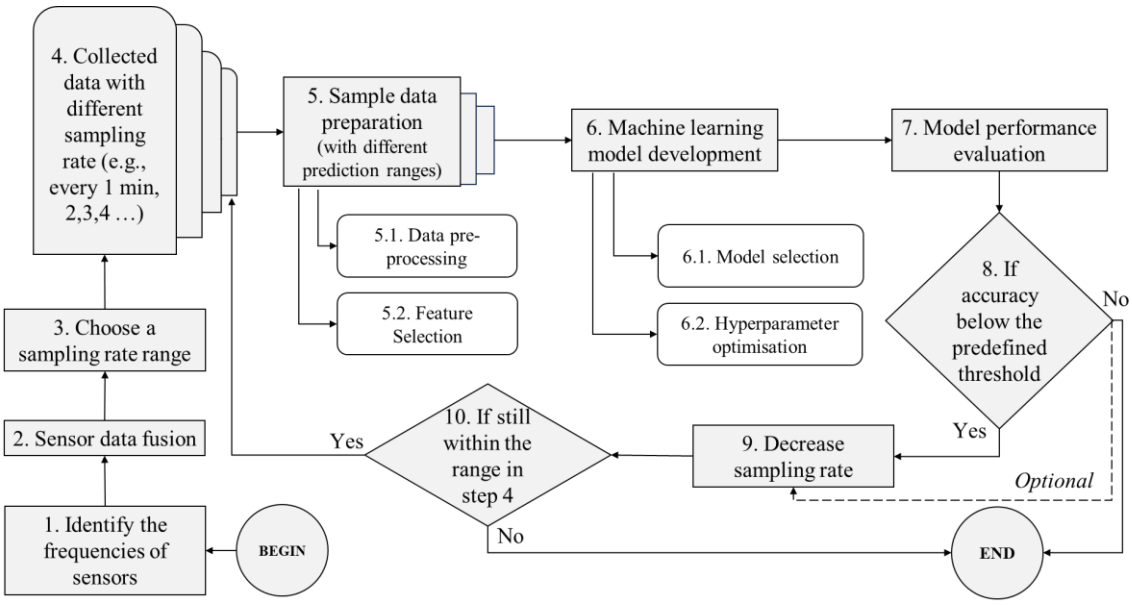


Figure 3.16. Process flowchart for selecting sensor data collection frequency

Step 1: To start with, a list of frequencies of different sensors is identified and so that it is ready for sensor data fusion. Table 3.12 details the frequency of each sensor used in NBIF experiment.

Step 2: According to the data collection frequency of sensors identified from the previous step, data from sensors needs to be fused. In NBIF case study, it was decided to sample the data at a frequency that was the lowest among the sensors to ensure the same number of readings across all the sensors for data integration. As a result, a sampling rate of one record every 1 minute was used as a minimum starting point, which is aligned with the temperature probe data sampling frequency.

Step 3: A sampling rate range is defined in this step, which means the maximum frequency can be arranged here. This frequency range varies depending on the use case and different scenarios. A larger sampling range would result in a lower number of sensor data. In the NBIF case study, every 20 minute was selected as the highest frequency after a thoughtful

consideration given the time taken to conduct each experiment, with an aim to ensure that enough data can be available to capture the trends and variations in the data.

Step 4: From this step onwards, an iterative process was formulated to identify the minimal amount of data or the least data sampling rate required to achieve the same or even better prediction accuracy that is obtained using more data points with a higher frequency. This process in this case study shares common parts of the ML modelling methodology presented in Section 3.5 but adds the element of iteration with different sampling sizes.

Step 5: In this step, the datasets based on Step 4 were pre-processed for ML modelling purposes. In addition, the inputs and the output were prepared accordingly for model developments and evaluation to achieve this. Different prediction ranges are also defined in this step in relevance to the defined data collection frequency.

Step 6: In this step, a machine learning model option is selected and built.

Step 7-8: From Step 7 onwards, the ML model performance is evaluated to decide whether to continue the iterative process of selecting the sampling rate. As shown in Step 8, the model performances such as R^2 and RMSE are checked against an existing threshold. This threshold can be based on expert knowledge or other sources. In this project, considering the overall road digital twin context, the threshold was set based on the prediction accuracy achieved in the short-term rutting predictions using historical data demonstrated in Case Study I (Section 4.1.1), assuming the prediction capacity using real-time sensor data should be more accurate than using historical data. Specifically, the expected model prediction accuracy thresholds are defined as ($R^2 \geq 90\%$ or $RMSE \leq 0.79$). Afterwards, if the model performance meets the threshold, then it proceeds to Step 9, if not, the user would have an option to choose either to

stop the iterative process and use the latest sampling rate or continue the trial-and-error procedure to check the frequencies in the sampling range, indicated by the dotted line.

Step 9-10: Step 9 reduces the sampling rate (e.g., from every minute to every other minute) and Step 10 checks if it exceeds the sampling rate range defined at Step 3. The iterative process continues if it is still within the range and stops otherwise.

Key steps (from step 4 onwards) of the iterative modelling process are presented in pseudo code (Box 3.1) for a better understanding.

Box 3.1. Pseudo Code for Iterative Modelling Process

PSEUDO CODE: ITERATIVE PROCESS FOR DATA COLLECTION FREQUENCY OPTIMISATION
While sampling rate (*s*) is less than every 20 minutes

1. Sample the data according to defined *s (start with 1 minute)*
2. Prepare data as model inputs and output for multiple prediction ranges: predict simultaneously, as well as next 20, 40, 60 minutes in future
3. Perform feature selection to select most relevant features based on the dataset
4. ML model development based on standard 80/20 split with training and test data
5. Model performance evaluation using R^2, RMSE and MAE
6. Save the results
7. Check model performance against existing prediction accuracy threshold
8. Increase *s* by 1 minute

A semi-automated iterative process was conducted to compare ML model performances under several selected sampling rates instead of all different sampling rates. A customised list of sampling rate: every 1, 2, 4, 5, 10, 20 minute and a list of prediction ranges (simultaneously, next 20 minutes, next 40 minutes and next 60 minutes) were thoughtfully observed. The frequency of data collection (F) and the range of predictions (R) can be denoted as F = {1, 2, 4, 5, 10, 20} and R = {0, 20, 40, 60}, respectively. These intervals were selected considering the average duration of the experiment each day as well as the feasibility of the data pre-processing and preparation.

In summary, the modelling process described previously was repeated 24 times to produce different scenarios with different sampling rates (F). The prediction accuracies were obtained with multiple different prediction horizons (R) to understand the sensitivity of the predictive model in identifying a relatively optimal data capturing interval that could achieve satisfying prediction accuracies to demonstrate the proposed approach to identify an optimised data collection frequency. The following sub-sections *a to c*, illustrate NBIF case study data and modelling details for from Step 5-7, such as description of prepared data, feature selection and model development and evaluation according to the proposed methodology with a baseline data collection frequency that was defined, 1 record per minute.

a. Data Description

After the data preparation stage, a statistical description of the data including inputs and output was conducted to understand the statistics of the sensor data produced. Table 3.15 presents the details of each feature based on sensor readings.

Table 3.15. Descriptions of the sensor data collected in this case study

Sensor readings		**Details**			
		Range (min - max)	Mean	Median	Standard deviation
Min	LVDT1 (mm)	(-0.0017 – 5.0031)	3.0547	2.7268	1.1617
	LVDT2 (mm)	(-0.0017 – 2.5240)	0.0290	-0.0016	0.2697
	LVDT3 (mm)	(-0.0048 – 9.1421)	6.0024	5.7250	1.9142
	SG-45 (μ)	(-183.97 – 901.60)	557.66	544.10	217.05
	SG-h (μ)	(-299.83 – 907.00)	652.76	720.00	194.97
	PC-M (kN/m^2)	(-0.4051 – 5.8684)	2.0371	2.0941	1.7535
	PC-C (kN/m^2)	(-2.0804 – 2.5363)	-0.0385	0.4741	1.1712
	PC-M-T (°C)	(12.563 – 16.262)	14.523	14.620	0.8865

	PC-C-T (°C)	(12.452 – 16.239)	14.482	14.574	0.9103
Max	LVDT1 (mm)	(-0.0014 – 5.0030)	3.3003	3.1776	0.9953
	LVDT2 (mm)	(-0.0015 – 2.5253)	0.0291	-0.0014	0.2699
	SG-45 (μ)	(7.0806 – 1201.6)	783.33	774.60	223.86
	SG-h (μ)	(19.000 – 975.00)	688.07	751.00	189.75
	PC-M (kN/m^2)	(-0.2554 – 8.5554)	4.6880	4.7961	2.3089
	PC-C (kN/m^2)	(-1.0846 – 3.7852)	1.3535	1.7889	1.3537
	PC-M-T (°C)	(12.563 – 16.262)	14.523	14.620	0.8865
	PC-C-T (°C)	(12.452 – 16.239)	14.482	14.574	0.9103
Max - Min	LVDT1 (mm)	(0.0000 – 0.5642)	0.2456	0.4056	0.2086
	LVDT2 (mm)	(0.0000 – 0.0020)	0.0002	0.0002	0.0001
	SG-45 (μ)	(3.0000 – 1116.0)	225.67	235.00	67.994
	SG-h (μ)	(4.0000 – 1059.0)	35.309	30.000	60.106
	PC-M (kN/m^2)	(0.0002 – 4.0397)	2.6509	2.6604	0.6477
	PC-C (kN/m^2)	(0.0006 – 3.7750)	1.3921	1.3696	0.2921
T1 (°C)		(13.040 – 16.160)	14.498	14.520	0.7932
T2 (°C)		(12.490 – 16.300)	14.517	14.610	0.9371
T3 (°C)		(12.900 – 16.290)	14.612	14.620	0.8541
T4 (°C)		(13.580 – 16.900)	15.050	15.080	0.9391
T5 (°C)		(13.610 – 16.510)	15.050	15.010	0.7279
T6 (°C)		(13.670 – 16.630)	15.115	15.110	0.7881
T7 (°C)		(7.5320 – 16.740)	13.034	13.680	2.0454

b. Feature Selection

This section identifies the most relevant features to serve as model inputs considering various input options. The importance and types of feature selection methods have been mentioned in Section 3.6.1.4. In short, irrelevant features, if not filtered out, can confuse the model and reduce the quality and accuracy. For simplicity purposes, this case study directly utilised an existing library provided by scikit-learn, namely SelectKBest (Pedregosa et al., 2011). This function returns the best k number of features based on p-value, where the lower p-value represents the more statistically significant correlation. Table 3.16 shows the Pearson correlation coefficient and the corresponding p-value results for all the features.

Table 3.16. Input and output features and results for their correlation coefficient and p-value, with selected features being highlighted in Green

	Features (Use the name from Table 3.14)	**Pearson correlation coefficient (sorted)**	**p-value**
Output	Min LVDT3	1.0000	n/a
Inputs	Max SG-45	0.8456	Close to 0
	Max PC-M	0.8386	Close to 0
	Min PC-M	0.8363	Close to 0
	Min SG-45	0.8190	Close to 0
	Max-Min PC-M	0.7256	Close to 0
	Max-Min LVDT1	0.3761	$6.6007 * e^{-179}$
	Max SG-h	0.2639	$1.0123 * e^{-85}$
	Min SG-h	0.2462	$1.6218 * e^{-74}$
	Max-Min SG-45	0.1697	$8.9266 * e^{-36}$
	Max-Min LVDT2	0.0980	$7.1967 * e^{-13}$
	Max LVDT1	0.0959	$2.2547 * e^{-12}$
	Max LVDT2	0.0836	$9.5385 * e^{-10}$
	Min LVDT2	0.0836	$9.5780 * e^{-10}$
	Max-Min SG-h	0.0344	0.0120
	Min LVDT1	0.0146	0.2859
	T7	-0.3025	$2.3872 e^{-113}$
	Max-Min PC-C	-0.4380	$4.3382 * e^{-249}$
	Max PC-C-T	-0.5219	Close to 0
	Min PC-C-T	-0.5219	Close to 0
	Max PC-M-T	-0.5268	Close to 0
	Min PC-M-T	-0.5268	Close to 0

	T2	-0.5507	Close to 0
	T1	-0.5620	Close to 0
	T5	-0.5642	Close to 0
	T3	-0.5669	Close to 0
	T6	-0.5879	Close to 0
	T4	-0.6109	Close to 0
	Max PC-C	-0.7980	Close to 0
	Min PC-C	-0.8130	Close to 0

In this case study, the parameter k was set to 10 (Al Alawi et al., 2024; Vengadeswaran et al., 2024). The feature selection process produced the 10 most suitable features according to their p-values. The results of the features are presented in Table 3.17. They are also highlighted in green colour in Table 3.16.

Table 3.17. Features after the feature selection process

	Selected Features
Output	Min LVDT3
Inputs	Max SG-45
	Max PC-M
	Min PC-M
	Min SG-45
	Max-Min PC-M
	T3
	T6
	T4
	Max PC-C
	Min PC-C

c. ML Model Development and Evaluation

In this case study, RF is selected to be used because of its high performance in Case study I on historical data, and the fact that it takes a relatively low level of computing power and time. In addition, it gives reasonably stable and good performance even without hyperparameter tuning based on the observation experienced in Case study I. With the data prepared after the feature selection in the previous section, the data was split into two parts -

training data (80%) on which the model was trained, and testing data (20%) that was used to evaluate the model performance. Default hyperparameters were used for RF model development. The model performance metrics used are R^2, RMSE and MAE. The model evaluation results for all scenarios are provided and discussed in Chapter 4.

3.7 Summary of the Methodology

In order to achieve the aim and objectives set out in this study, first of all a DT-based decision-making support theoretical framework for road life cycle (Figure 3.2) was developed. The model was based on the comprehensive literature review on road management as well as the concept, characteristics, components and applications of DTs. This contributes to the whole body of knowledge on road DTs as there is limited research in this area. Then the study looked specifically into the modelling part in the Data Integration or Modelling Layer of the theoretical framework by presenting a DT-based ML modelling methodology (Figure 3.3). Two case studies leveraging the usage of historical and real-time data have been conducted to demonstrate the applications of the proposed theoretical framework in the Application Layer and the modelling methodology.

Case study I built a DT-based pavement performance model using historical data. The novelty of the DT-based model is the combination of ML with physics-based simulation models using two different approaches. In addition, ML uncertainty has been taken into consideration in the form of ML model variability. This case study addressed the research gap revealed in the literature review about pavement performance modelling where integrating existing domain knowledge into the ML modelling process is still in its early stage.

Case study II demonstrated the use of real-time sensor data as part of the theoretical framework, especially the application of optimising data collection frequency with acceptable

accuracy. Utilising the sensor data available as part of an NBIF experiment, by following the iterative modelling process, the amount of data that can be reduced to still enable a satisfactory modelling prediction accuracy were quantified. This analysis is useful for future data collection considerations and practices as part of the application layer within an RDT.

3.8 Utilised Software and Packages

The end-to-end research steps for both case studies during the modelling process have been enabled by making use of multiple open-source software, computing platforms, data science libraries and packages, as well as available hardware, as presented in Table 3.18.

Table 3.18. Utilised software and packages

Software and Packages	Description	Reference
Programming language	Python 3.9.2	(Van Rossum and Drake, 1995)
Computing platform	Jupyter notebook 6.1.4	(Kluyver et al., 2016)
Software library	Pandas 1.1.0 (Data analysis library) NumPy 1.19.5 (Library used for working with arrays) SciPy 1.5.4 (Library used for scientific computing) Matplotlib 3.3.2 and Seaborn 0.11.0 (Statistical data visualisation library) Scikit-learn 0.23.1 (Machine Learning library) Mlxtend 0.18.0 (Machine leaning extension library for data science tasks)	(Hunter, 2007; McKinney et al., 2010; Pedregosa et al., 2011; Waskom et al., 2017; Raschka, 2018; Harris et al., 2020; Virtanen et al., 2020)
Software suite	Abaqus Finite Element analysis	(Smith, 2009)

4 RESULTS AND DISCUSSIONS

In this chapter, the results of the two case studies implementing the decision-making support theoretical framework as well as the modelling methodology described in Chapter 3 are presented. Individual discussions are provided for each case study, an overall discussion is also given in the context of the RDT.

The two case studies respectively achieved the objectives proposed in Chapter 1. This chapter is organised as follows to present the results for rutting predictions (Section 4.1), IRI predictions (Section 4.2), and the suitable data collection frequency using NBIF sensor data (Section 4.3).

4.1 Predictions on Rutting

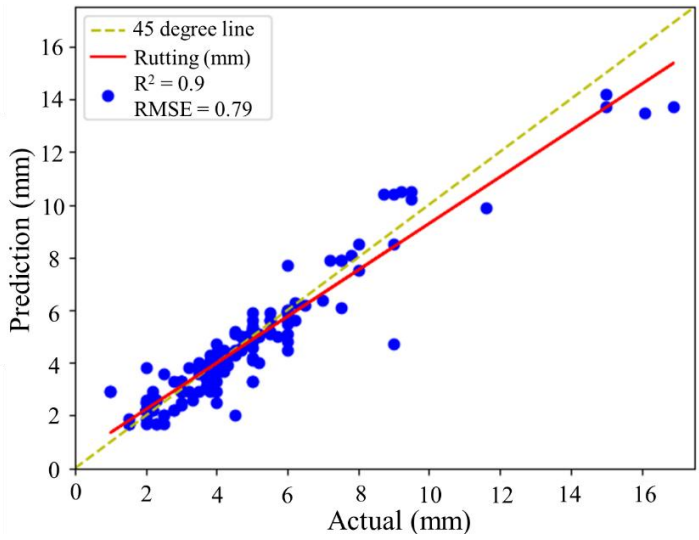
This section presents the rutting prediction results for the two scenarios defined in Table 3.9. The results entail short-term predictions (1-year) and prediction intervals quantifying ML prediction uncertainty, as well as long-term 90^{th} percentile range predictions (2^{nd} -12^{th} year) for both scenarios. Afterwards, an analysis and discussion of the performance of the predictions are presented.

4.1.1 Short-Term Predictions

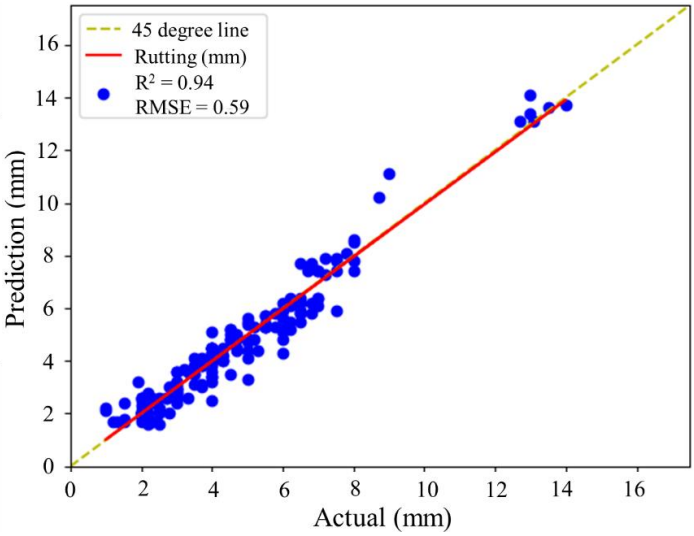
For each model generation, applying iteration according to the process defined in Figure 3.8, results of model evaluation using 10-fold cross-validation on 84 randomly selected sections and the result of the model performance on 15 randomly selected unseen test sections were acquired. Without considering the uncertainty in the ML model caused by different training datasets, the model's performance results are summarised in Table 4.1, and model's one-year prediction performances on the testing datasets are shown in Figure 4.1.

Table 4.1. Model performance for rutting one-year prediction

One-Year prediction	Results from 10-fold cross-validation on Training and Validation datasets (R^2)	Results from Testing dataset (R^2)
Scenario 1 (ML with LTPP data)	Median: 93.1% Max: 95.5% Min: 88.3%	90.3%
Scenario 2 (ML with LTPP + physics-based simulated deflection data)	Median: 93.6% Max: 94.6% Min: 83.1%	94.2%



(a)



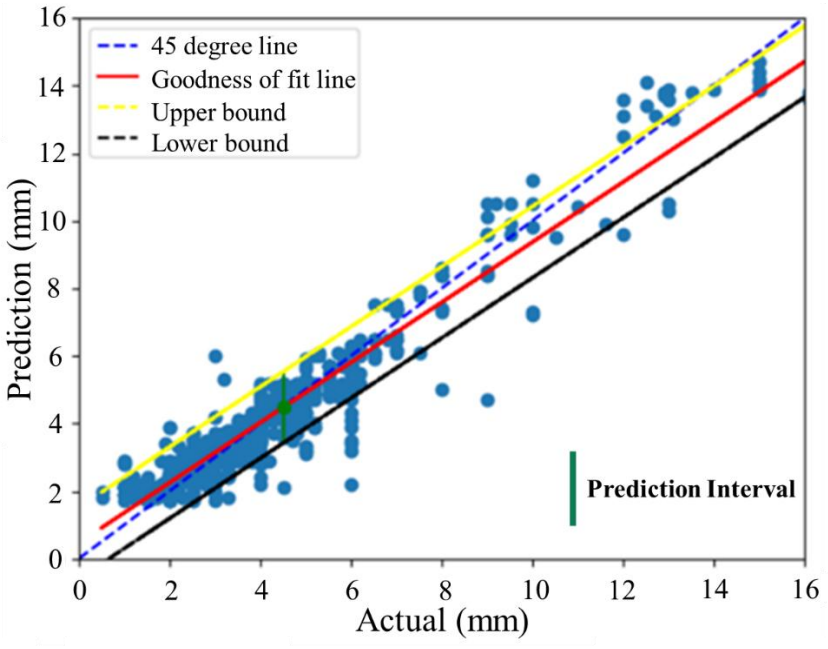
(b)

Figure 4.1. One-year model deterministic prediction results of test sections using a) LTPP data only and b) LTPP + physics-based simulated deflection data

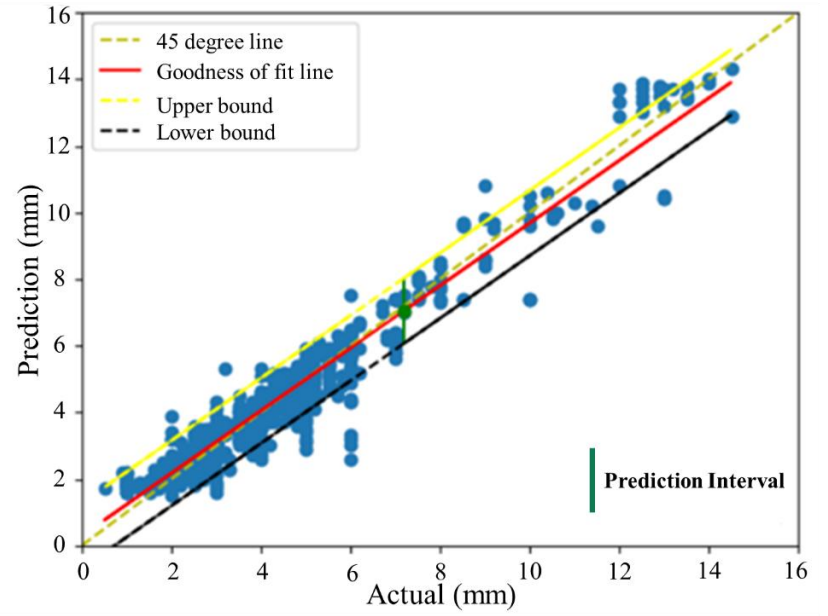
For one-year prediction, Figure 4.1 shows that Scenario 2 improves the performance compared to Scenario 1 from R^2 90.3% to 94.2%. Meanwhile, the RMSE decreased by 25.3%, from 0.79 to 0.59. This implies a model accuracy enhancement for one-year projection with data from physics-based FE simulations. This finding is in consistency with other studies in which similar approach was adopted in the prediction of steel connection stiffness (Duran et al., 2022; Cabrera et al., 2023) and lake temperature (Daw et al., 2022).

4.1.2 Prediction Intervals

In order to quantify the uncertainty of the ML model, prediction intervals with a 90% confidence level were obtained. For visualisation purposes, the model development process was repeated five times to clearly show distinct model performances as a result of multiple runs with different training data each time. Figure 4.2 shows the prediction intervals for both scenarios, and Figure 4.3 provides the visualisation of model performance on testing sets after five runs. The results illustrate that the one-year prediction interval decreased for Scenario 2 in comparison to Scenario 1 from a range of the fitted regression results of ± 1.056 mm to ± 0.980 mm. This means the predictions produced by the ML model increased their precision. This indicates that Scenario 2, with additional deflection data generated by physics-based FE simulations, has contributed to a decrease of 7.2% in the model's one-year prediction interval range, thus reducing the uncertainty.

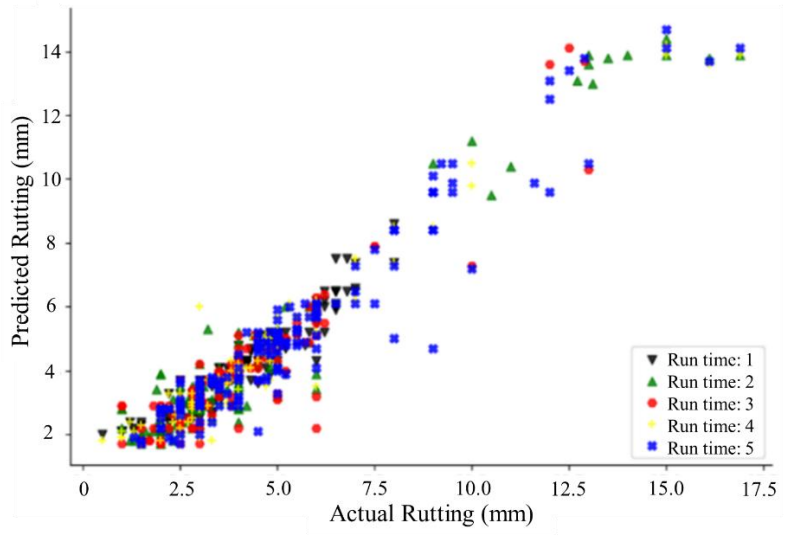


(a)

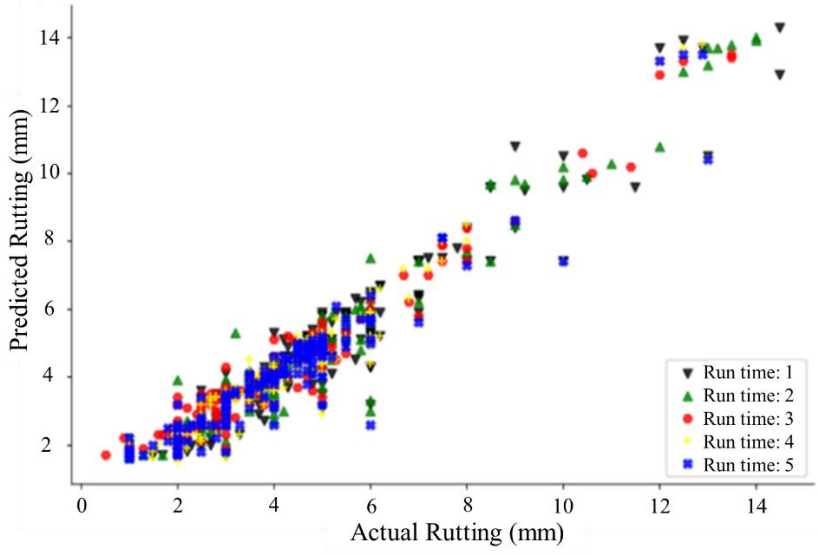


(b)

Figure 4.2. Prediction intervals with 90% confidence level on test sections for a) Scenario 1 and b) Scenario 2



(a)



(b)

Figure 4.3. Visualisation of prediction intervals with 90% confidence level on test sections for a) Scenario 1 and b) Scenario 2 after five runs of the model development process with different training data each time. Each colour represents the result of one run.

The visualisation shows that the ML model generalisation predictive capacity could yield stable results for both scenarios under varied training data. This ensures the robustness of the model performance with data from pavement sections with completely different internal and external characteristics. The reduction in the prediction interval with 90% confidence level also provides reassurance that having additional FE data improves the ML model's prediction

accuracy. This inherent ML model uncertainty, after being quantified, can be used to enable the model to make year by year 90th percentile range predictions, which are presented in the next section.

4.1.3 Multi-Year 90th Percentile Range Predictions

To achieve multi-year predictions while taking account of the ML's underlying uncertainties, an alternative method based on residual bootstrap was used to calculate the prediction intervals, thus generating 90% percentile simulation ranges for each year. A standard exemplary use case, Road Section 12-0566 from the LTPP database, was selected for demonstration purpose following this approach primarily because the rutting condition data observed for this pavement test section started from 0mm following a major maintenance, and it also followed a pattern similar to the typical pavement deterioration curve over the years (Zimmerman and Peshkin, 2003; Amarh, 2017). Two thousand predictions were generated for each year. This number was determined based on the standard error results calculated using the equation expressed in Eq. 4-1 from these sample predictions. Between 2000 and 5000 runs, a low and stabilised standard error was observed, as shown in Figure 4.4, indicating that this number of sample predictions is enough to represent the whole population of the predictions (Sisodia and Sisodia, 2022).

$$SE = \frac{\boldsymbol{\sigma}}{\sqrt{n}} \qquad \text{Eq. 4-1}$$

where SE is the standard error of the sample predictions; σ is the sample predictions standard deviation; and n is the number of sample predictions.

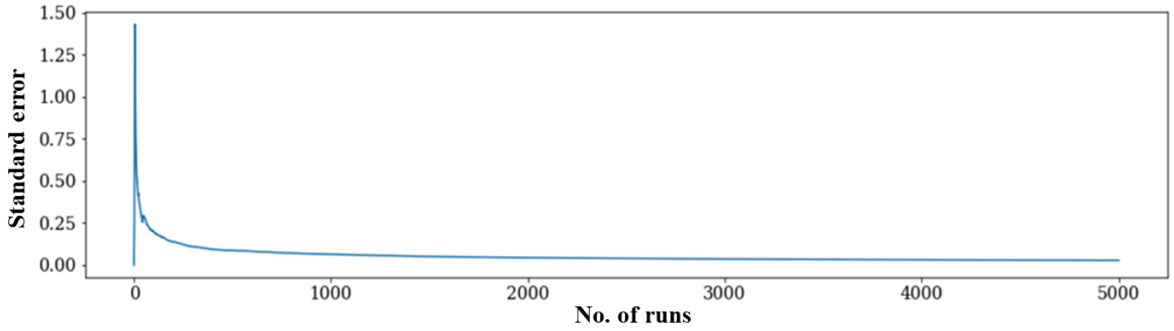
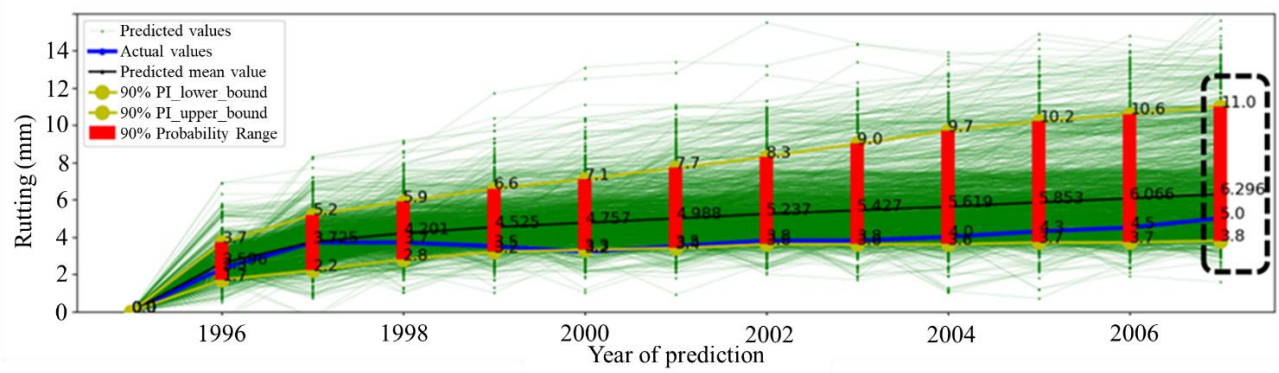


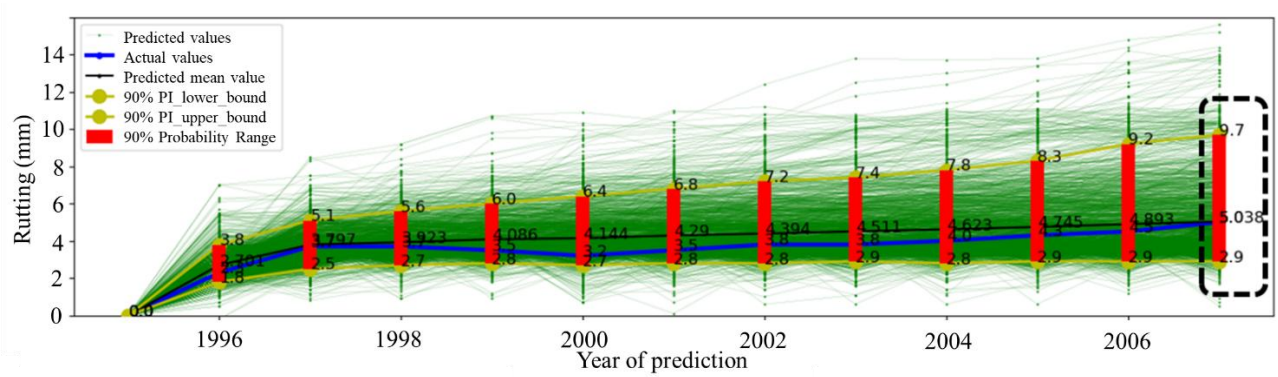
Figure 4.4. Standard error vs. number of runs

Figure 4.5 presents the results for both scenarios with the multi-year prediction ranges and the actual values throughout the years using one of the 30 developed models for this section as an example using the bootstrapped residuals method.

Considering the collection of uncertainties made from 30 different models trained on different training data, the multi-year predictions were also run 30 times to ensure 30 different models with different training sections were used to compare between both scenarios for evaluating the validation of the model's general predictability. Table 4.2 displays all the results based on all 30 models. It shows that for the majority of the 30 different models, Scenario 2 (the inclusion of extra simulated deflection data according to physical principles) improved the long-term (the 12^{th} year) prediction precision by between 1.52% and 16% regarding the 90^{th} percentile range of the predicted values. It should be acknowledged that two model results showed the potential risk of decreasing the model performance and four model results showed no improvement. But on average, Table 4.2 indicates the 90^{th} percentile range predictions have been narrowed down by 6.76% from Scenario 1 to Scenario 2, showing an improvement in the prediction certainty for the 12^{th} year.



(a)



(b)

Figure 4.5. Multi-year predictions of one model for Section 12-0566: a) Scenario 1 and b) Scenario 2 considering ML uncertainties

The example presented in Figure 4.5 shows the 90th percentile range of the model's predictions increases over the years for both scenarios, and looking into the predictions for the 12th year, Scenario 2 improved the whole prediction compared to Scenario 1 by reducing the rutting prediction ranges from 7.2 mm to 6.8 mm.

It can also be observed that the actual values in Figure 4.5 are quite distant from the centre of the confidence intervals in the predictions. This can be explained in that the actual values (the blue line) are from one particular section and the predictions (green lines) were generated based on the ML model and its associated uncertainties in the modelling process using data from different training sections. The uncertainty includes multiple sources such as measurement errors in the data collection process, as well as the ML modelling uncertainties due to the selection of training data during repetitions and the variety of road sections from

which the data has been collected. Hence, Figure 4.5 demonstrates the successful reduction of uncertainties in the multiple-year prediction ranges through the supplement of physics-based numerical modelling deflection data into the ML model development procedures.

Table 4.2. The 12th year rutting prediction details with 30 models for both scenarios

Scenario 1 (ML)	Scenario 2 (Hybrid)	% of Range Reduction from Scenario 1 to 2	Scenario 1 (ML)	Scenario 2 (Hybrid)	% of Range Reduction from Scenario 1 to 2
1 – 15 Model Results (90th Percentile Range)			16 – 30 Model Results (90th Percentile Range)		
7.2	6.8	5.56%	7.1	6.3	11.27%
7.1	6.2	12.68%	7	6.1	12.86%
6.7	6.7	0.00%	7.5	6.8	9.33%
7.5	6.6	12.00%	7	6.3	10.00%
6.8	6.8	0.00%	6.9	6.6	4.35%
7.2	6.6	8.33%	7.1	6.8	4.23%
6.9	6.9	0.00%	7.5	6.3	16.00%
7.3	6.5	10.96%	7.5	7.3	2.67%
7.1	6.2	12.68%	7.5	6.8	9.33%
7.2	6.6	8.33%	7.5	6.5	13.33%
7	6.8	2.86%	7.2	6.7	6.94%
6.6	6.5	1.52%	7	6.5	7.14%
7.3	6.4	12.33%	7.6	7.8	-2.63%
7.2	6.7	6.94%	7.4	6.8	8.11%
7.4	7.4	0.00%	6.8	7.1	-4.41%
Average % of Range Reduction from Scenario 1 to 2	6.76%				

Both results presented in Table 4.2 demonstrate that Scenario 2 in which ML is supplemented with physics-based FE simulation data in general has made a notable improvement in the accuracy and reliability of the model for multi-year predictions. This slightly limited advancement could have been because of the simplistic physics-based FE models which may not fully reflect the complete realistic physics being modelled and plus various assumptions that had to be made in the FE simulation modelling process. The assumptions include material properties, which vary with different construction methods, the age of material and subgrade properties. More advanced simulation could potentially improve the outcome of this framework, and this is included as one of the future works.

4.1.4 Key Findings of Rutting Predictions

The results shown for rutting prediction suggest that this combined approach of enriching the dataset from the public LTPP database with physics-based FE simulations based on the information available could produce 1) an increase in the ML model's short-term prediction accuracy; as well as 2) a decrease in the uncertainty in the ML model's prediction ranges (both short-term and long-term).

Especially for multi-year predictions, the addition of physics-based FE simulation data has helped the ML model to decrease the prediction ranges almost every single year especially when the range becomes larger. From an asset owner's perspective, this increase in prediction confidence could result in a more efficient long-term maintenance strategy plan with reduced maintenance spend, traffic delay and congestion and more accurate financial forecasting for multi-year investment periods without necessarily requiring additional data collection in between.

While various ML algorithms have shown a decent pavement performance prediction accuracy in previous research, the lack of data and data quality has prevented the pavement

community from fully unlocking ML capacity. These results are encouraging, partially overcoming this issue by generating synthetic data using physics-based FE simulations. The same approach could be used to model other meaningful pavement performance indexes or defects, giving the potential to enhance ML model prediction accuracy.

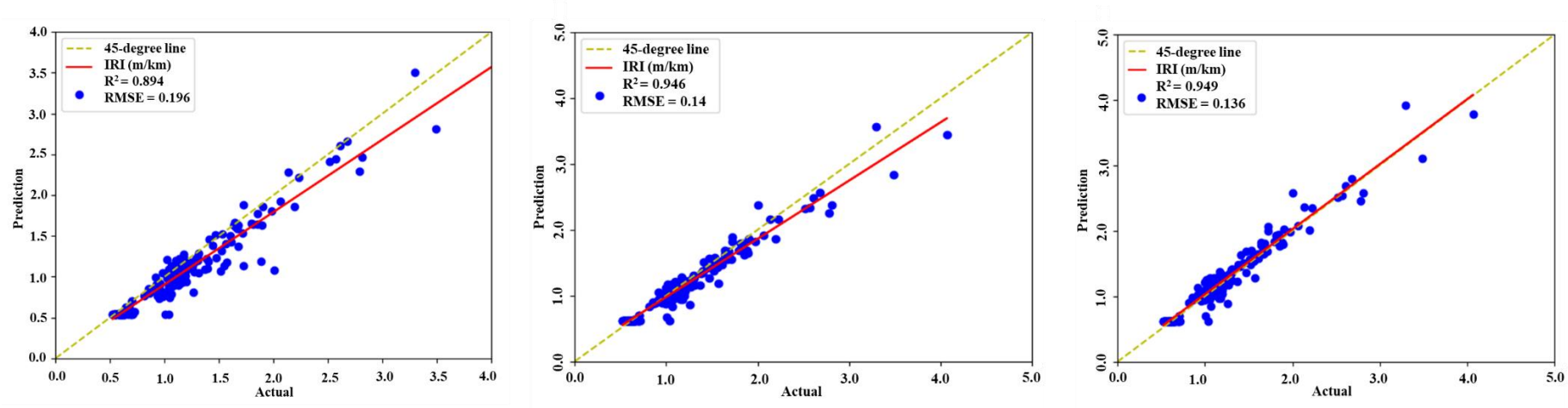
The study has also opened a new research direction where ML models are integrated with simulations for pavement performance prediction. While this integration was only at the data level in this research, future studies on rutting modelling can explore advanced combinations such as PEML to enhance a model's performance further as well as its interpretability.

4.2 Predictions on IRI

This section presents the prediction results on IRI for all three scenarios defined in Table 3.9. The results include short-term predictions (1-year) and prediction intervals quantifying ML prediction uncertainty, as well as long-term 90^{th} percentile range predictions (2^{nd} - 12^{th} years) for all scenarios. Afterwards, an insightful comparative analysis and discussion of the model performances are presented.

4.2.1 Short-Term Predictions

For IRI prediction, the training and testing sections were fixed to clearly understand and compare the one-year prediction of model performances between the different scenarios, as shown in Figure 4.6.



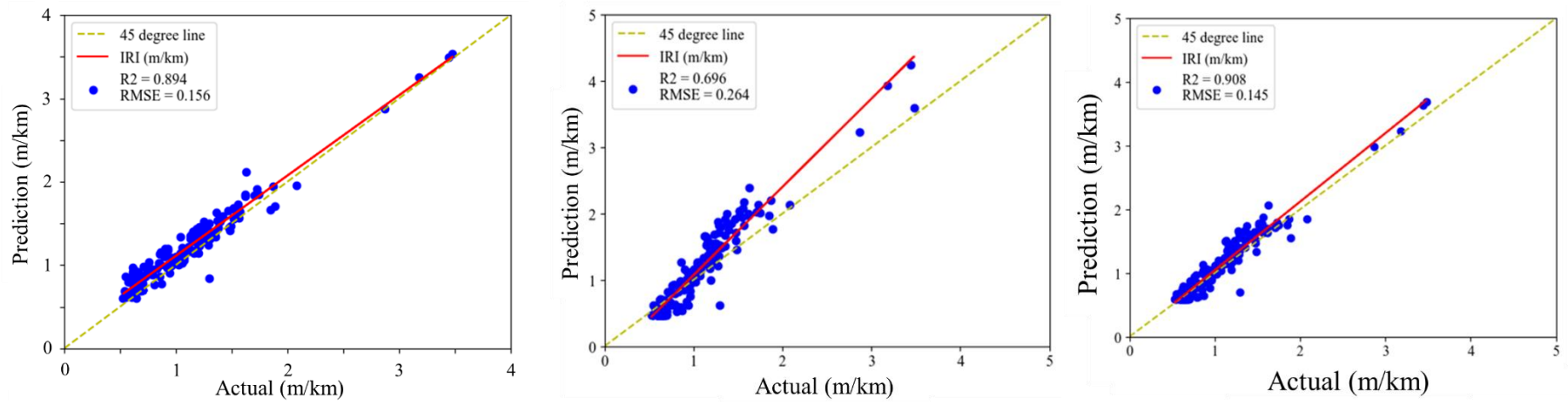
(a) **(b)** **(c)**

Figure 4.6. One-year model deterministic prediction results on test sections for three scenarios a) Scenario 1 b) Scenario 2 c) Scenario 3

The results in Figure 4.6 show that, in this case, the integration of the physics-based model into ML (Scenario 2) has improved the model performance by 5.8% in terms of R^2 from 89.4% to 94.6% and reduced the RMSE by 28.6% from 0.196 m/km to 0.140 m/km. In Scenario 3, the R^2 increased further from 94.6% to 94.9% and RMSE decreased more by 2.9% to 0.136 when incorporating a physics-informed loss function. This finding is in consistency with the rutting predictions presented in Section 4.1. The further improvement in Scenario 3 compared to Scenario 2 demonstrated the effect of using physics-informed loss function with the extra up sampled data. However, as ML builds upon data so the increased accuracy could be attributed to the additional data itself rather than the updated loss function. The performance of the model's prediction was further evaluated by training a new set of data using a traditional loss function and a physics-informed loss function within Scenario 3. The next sub-section presents the results.

4.2.1.1 Performance comparison between traditional and physics-informed loss function

In order to evaluate further the impact of physics-informed loss function in Scenario 3 and show the improvement more distinctively, another run was conducted where the training and testing sections were the same. Figure 4.7 provides the detailed results.



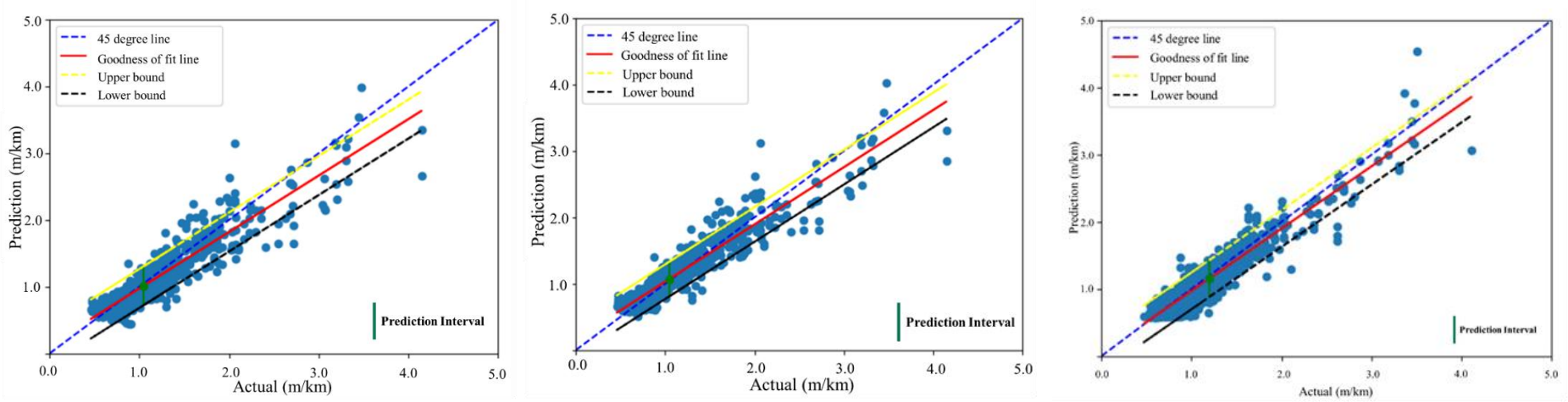
(a) **(b)** **(c)**

Figure 4.7. One-year model deterministic prediction results on different test sections for three scenarios a) Scenario 1 b) Scenario 3 (traditional loss function) c) Scenario 3 (physics-informed loss function)

4.2.2 Prediction Intervals

Similar to the rutting predictions, this section also presents the 90% prediction interval based on the methodology explained in Chapter 3 for all three scenarios.

Visualisation as a result of the 5 times repetitive model development process was produced to show the distinct model performances with different training data across all scenarios. Figure 4.8 presents the prediction interval results.



(a) **(b)** **(c)**

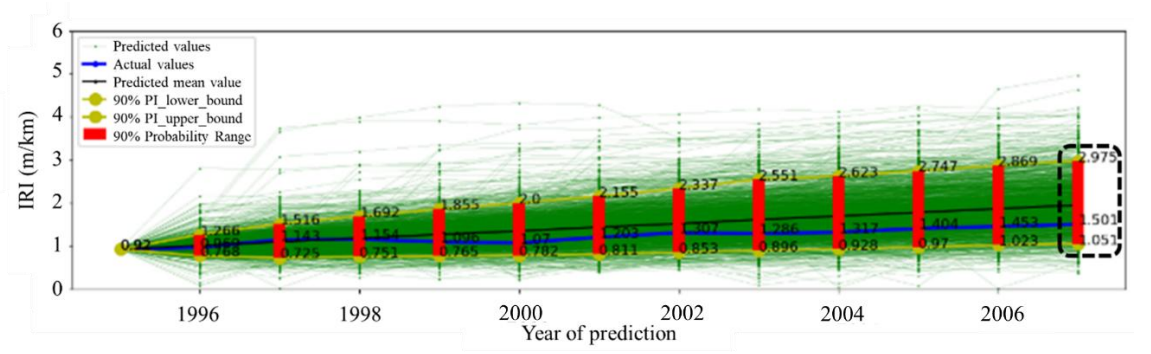
Figure 4.8. 90% prediction intervals on sections for three scenarios a) Scenario 1 b) Scenario 2 c) Scenario 3

Despite the similarity in the results among the three scenarios, the prediction interval became relatively smaller from ± 0.274 to ± 0.264 (3.7% improvement) in Scenario 2, and it was further reduced by 1.5% from ± 0.264 to ±0.260 in Scenario 3. This indicates a modest enhancement of prediction accuracy with additional physics-based elements.

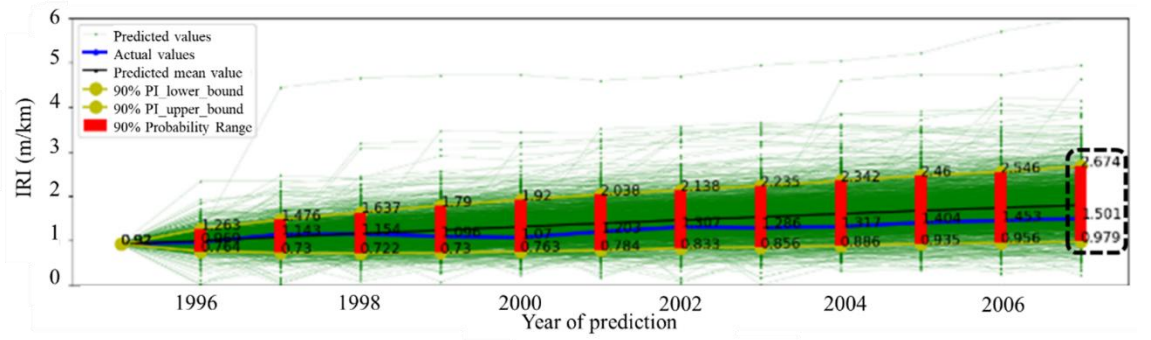
The results show the ML model's good general prediction capability with different training data, proving the robustness of the model taking data from multiple pavement sections with various internal and external characteristics.

4.2.3 Multi-Year 90th Percentile Range Predictions

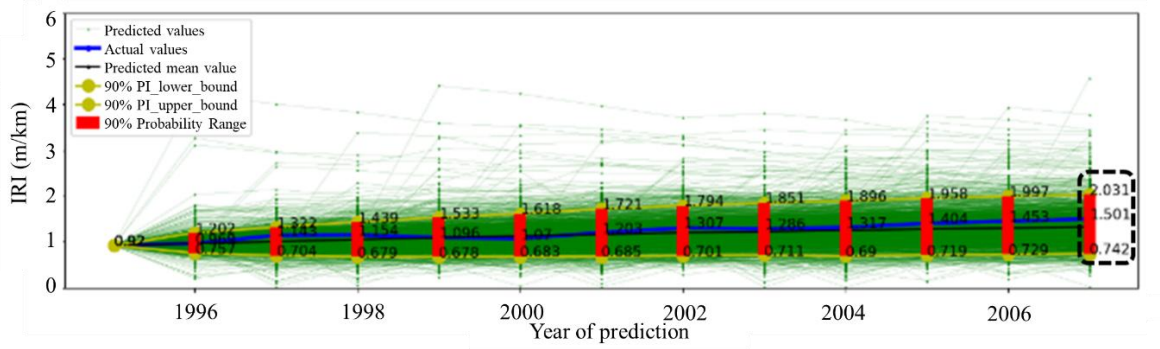
As described in Chapter 3, the residual bootstrapping approach was used to measure the uncertainty in the model's yearly predictions. The approach was then repeated 2000 times to acquire an estimated 90th percentile range predictions for each year. This section displays the results for a selected road section. Based on the 2000 predictions each year, the mean value of the predictions from the 2nd year onwards and in the last year (the 12th year), as well as the 90th percentile range predictions for the last year have been calculated to measure and quantify the uncertainty for long-term predictions for all scenarios. Road Section 46-0608, from the US LTPP database, was chosen as a test section as this section shows a steady increase in the IRI value over the years, expected due to the physical deterioration behaviour of a road under normal conditions (Zeiada et al., 2019). Figure 4.9 provides an example of the multiple year 90th percentile range predictions for this section over the years and how it differs among the three scenarios.



(a)



(b)

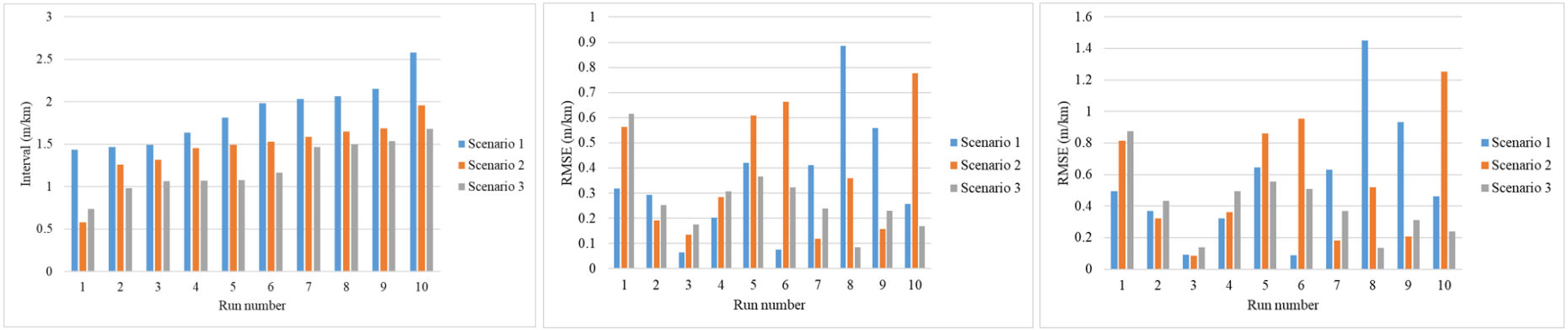


(c)

Figure 4.9. Multi-year 90th percentile range predictions of one model for Section 46-0608 for all scenarios a) Scenario 1 b) Scenario 2 c) Scenario 3

The multi-year predictions were run 10 times (Vabalas et al., 2019; Wan et al., 2021) to evaluate performances on 10 models with different training data to ensure the model's generalisation ability. For all scenarios, a comparison was performed on the average 90th percentile range predictions for the last year, and the corresponding average RMSE for the predictive values from the 2nd year onwards, as well as the corresponding average RMSE for the prediction for the last year. The results are presented in Figure 4.10.

The 10 runs are sorted in an ascending order based on the results. A detailed performance summary for all scenarios is provided in Table 4.3.



(a) **(b)** **(c)**

Figure 4.10. Performance comparisons for 10 times of run across all scenarios a) 90th percentile range prediction for the 12th year b) RMSE from the 2nd year onwards c) RMSE for the 12th year

Table 4.3. Performance evaluation of different scenarios

Scenarios	One-Year Prediction			Multi-Year Predictions		
	R^2	RMSE	90% Prediction Interval after multiple time of runs	Average 90^{th} percentile range prediction for the 12^{th} year	Average total prediction RMSE from the 2^{nd} year onwards	Average prediction RMSE for the 12^{th} year
Scenario 1	89.4%	0.196	0.274	1.865	0.349	0.549
Scenario 2	94.6%	0.140	0.264	1.452	0.386	0.556
Scenario 3	94.9%	0.136	0.260	1.228	0.276	0.406

For IRI prediction results, a prediction improvement can be observed for one-year prediction especially in the RMSE showing a decrease of 30.6% comparing Scenario 3 against Scenario 1. A more significant improvement in model stability can be seen for multi-year predictions where the RMSEs for the 12^{th} year's prediction as well as over the years dropped by 26.0%, and 20.9%, for the two scenarios respectively. The 90^{th} percentile range prediction for the 12^{th} year reduced by 34.2% from Scenario 1 to Scenario 3. Scenario 2 produced a similar improvement to Scenario 3 for one-year prediction, but a slight increase in RMSE for multi-year predictions by 9.6% over the years and 1.3% for the last year. Nevertheless, it should be noted that Scenario 2 yielded more stable predictions as indicated by the narrower 90^{th} percentile range prediction for the last year, a noticeable 22.1% decrease compared with Scenario 1.

The IRI prediction results in this case study demonstrated an improved accuracy as well as an enhanced reliability and generalisation capacity of this hybrid approach. The outcomes are consistent with findings in rutting prediction in this study. In addition, it showed the potential in solving mechanics problems (Raymond and Camarillo, 2021) by showing a significant improvement when incorporating simple physics models into the loss function during the ML model training process. In addition, the findings in Scenario 2 also align with existing works

utilising physics enhanced ML in the application of pavement management where the inclusion of physics in the ML modelling process has succeeding in improving the model's stability but at the cost of prediction accuracy reduction. For example, the R^2 values decreased by 3% in the study conducted by Deng et al. (2024) and similar results have been reported by Kargah-Ostadi et al. (2024) where the physics enhanced NN produced an increased MAE and standard error while achieving greater generalisation capacity. However, the results from this case study also provide new perspectives, especially on the results from Scenario 3, where the model demonstrated an improved stability by narrowing down the average 90^{th} percentile range predictions for the 12^{th} year without weakening the prediction capability but rather enhancing it even further by reducing the prediction errors accumulated over the years.

4.2.4 Key Findings of IRI Predictions

The results show a balanced improvement in the accuracy of prediction as well as generalisability when integrating physics into the ML modelling for pavement performance prediction in both short term and long term.

For one-year predictions for IRI, the two approaches of combining the output from the physics-based FE model with ML both further improved the performance of the ML model, which on its own could already reach almost 90% accuracy in terms of the R^2, thereby demonstrating the potential of hybrid models.

The uncertainties taken into consideration include various sources of potential error, such as measurement errors during the process of data collection, and inherent uncertainty as part of the ML modelling process because of the selection of different training data as well as the diversity of pavement sections where the data was collected. The results show that the quantified uncertainty, expressed either as a 90% prediction interval for one-year prediction

or a 90^{th} percentile range of multi-year predictions, reduces when extra data generated based on domain physical principles is added.

In addition to the advancement in prediction accuracy, the approach of assimilating domain knowledge while building an ML model has improved the model's reliability and rationality to some extent, indicated by the model's stable predictions across multiple runs with different training data.

Considering simplicity and practicality from a transportation administrative perspective, the findings of the study would increase the confidence in the model's predictions, especially when applied at a network level with predictions needing to be performed across different sections. This could result in a better optimised maintenance strategy, more informed decision support especially on prioritisation, as well as more accurate financial forecasting for multi-year investment periods.

4.3 Optimisation of Data Collection Frequency Using NBIF Sensor Data

This section discusses the results of data collection frequency optimisation from the NBIF experiment according to the methodology described in Section 3.6.2. Selected values were predefined in the list of data collection frequencies (F) and prediction ranges (P) in Section 3.6.2.3. In total, twenty-four models were built accordingly with their corresponding predictive results on the test data following the standard 80-20 data training and validation split. According to the pseudo code described in Box 3.1, twenty-four prediction results were obtained for different combinations of F and P. Figure 4.11 shows different sizes of datasets when sampled with every 1, 2, 4, 5, 10, and 20 minute for readings from LVDT3 sensor in Experiment 5 as an example. As it can be observed from the figure that downsampling would result in a reduction of total number of readings in one experiment that is used for ML model training and evaluation.

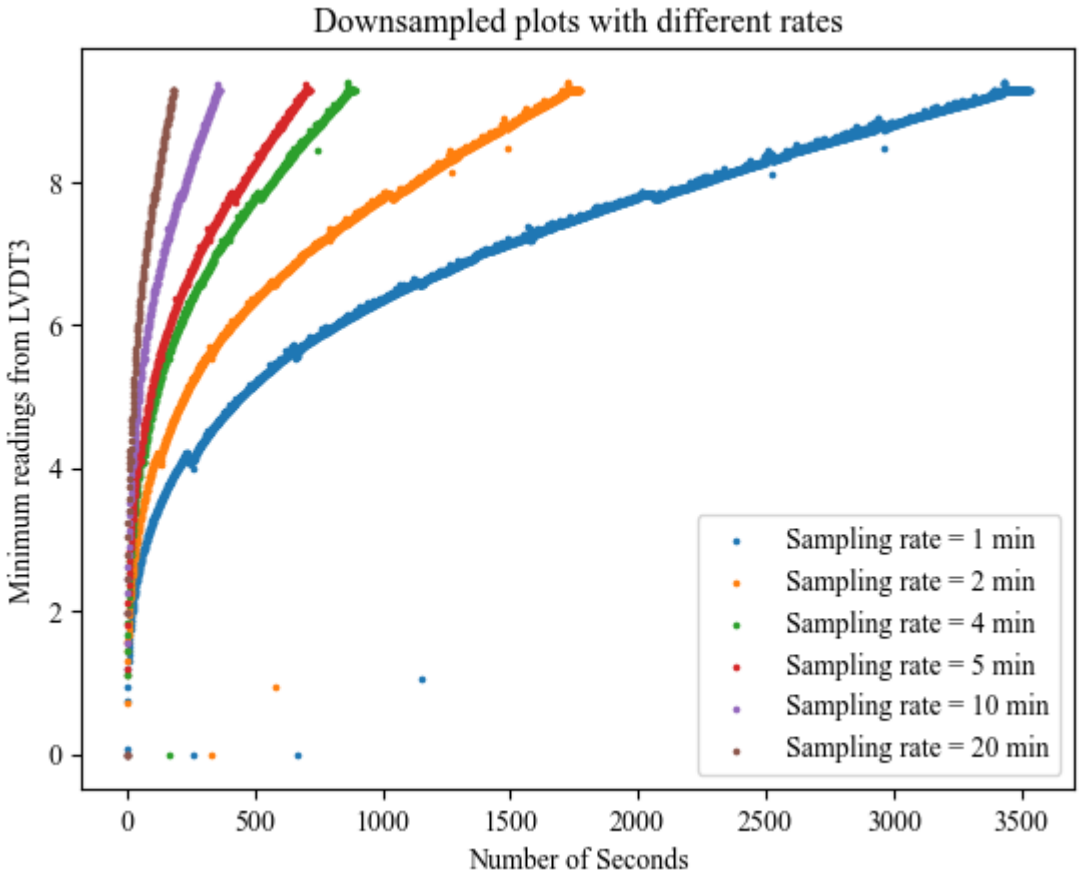


Figure 4.11. Downsampled plots with different sampling rates

Table 4.4 demonstrates the performance metrics of the built models using data sampled with different rates, tested across all selected prediction ranges. The results show that despite the variety in data collection frequencies and different future forecast horizons, the RF model's prediction accuracy remained well above the accuracy threshold defined in Section 3.6.2.3 across most of the combinations. The only one time when the prediction accuracy, i.e., R^2, dropped slightly below the 90% threshold was for the 20 minutes frequency to predict the next 20 minutes (Figure 4.12). Having said that, RMSE in the same scenario (0.728mm) was still better than the 0.790mm threshold value, meaning a satisfactory prediction. An outlier can be observed in Figure 4.12, negatively impacting the accuracy. This means the drop in prediction accuracy is largely attributed to possible outliers in the data, rather than the

decrease in the frequency itself by which the data is collected. Given the noises experienced in sensor data, the outliners are also expected due to the data pre-processing leaving a few incorrect datapoints which ML could not learn during the training, hence resulting in data errors. Overall, the rest of the predictions for this data collection interval are above the thresholds.

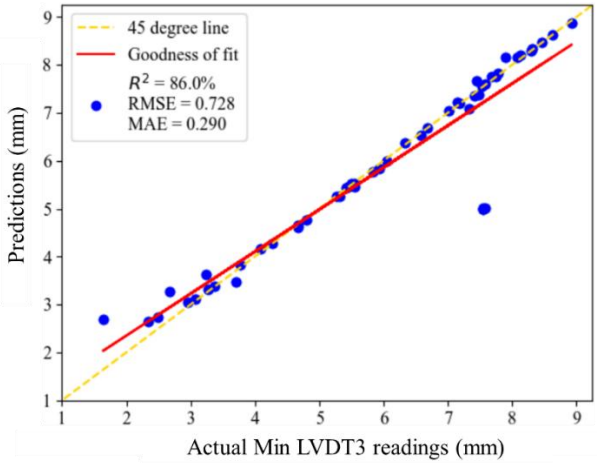
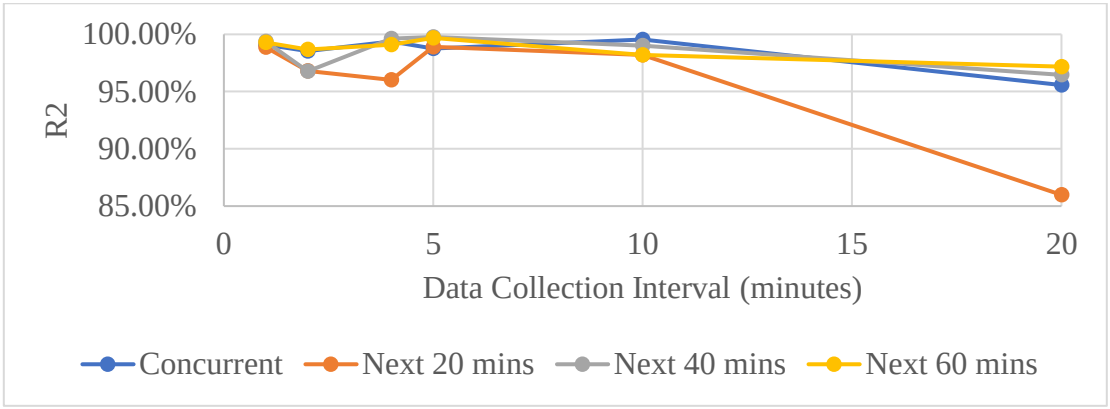


Figure 4.12. Performance metrics on test data when sampling every 20 minutes to predict the next 20 minutes

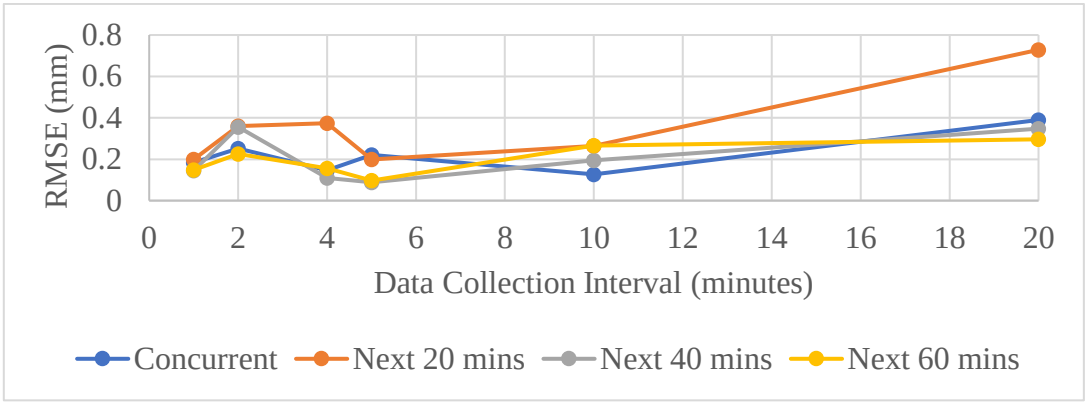
Table 4.4. Performance metrics with different sampling rates across multiple prediction ranges

Sampling rates	Prediction Ranges											
	Concurrent			Next 20 mins			Next 40 mins			Next 60 mins		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
Every 1 min	99.09%	0.1843	0.0265	98.87%	0.1988	0.0295	99.37%	0.1455	0.0245	99.30%	0.1486	0.0271
Every 2 min	98.54%	0.2514	0.0395	96.80%	0.3608	0.0608	96.77%	0.3558	0.0631	98.67%	0.2260	0.0454
Every 4 min	99.37%	0.1460	0.0360	96.03%	0.3745	0.0576	99.63%	0.1098	0.0360	99.10%	0.1555	0.0526
Every 5 min	98.76%	0.2205	0.0707	98.91%	0.1987	0.0667	99.77%	0.0886	0.0407	99.69%	0.0970	0.0416
Every 10 min	99.55%	0.1268	0.0612	98.19%	0.2637	0.0989	99.02%	0.1951	0.0722	98.19%	0.2659	0.1136
Every 20 min	95.57%	0.3896	0.2006	85.98%	0.7284	0.2896	96.46%	0.3470	0.1533	97.18%	0.2962	0.1518

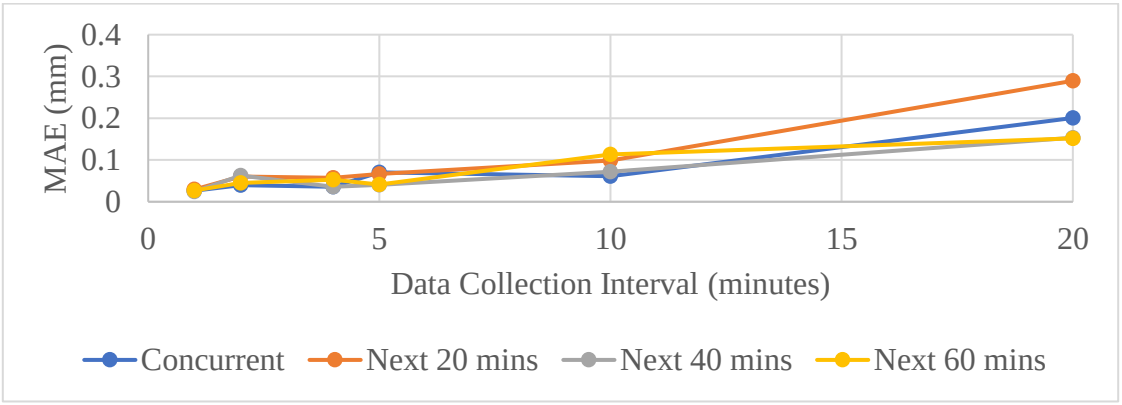
In addition, the results from Table 4.4 demonstrate that the model's prediction accuracy decreases slightly for most scenarios as the data collection interval increases. Figure 4.13 illustrates the performances measured by three different metrics.



(a)



(b)



(c)

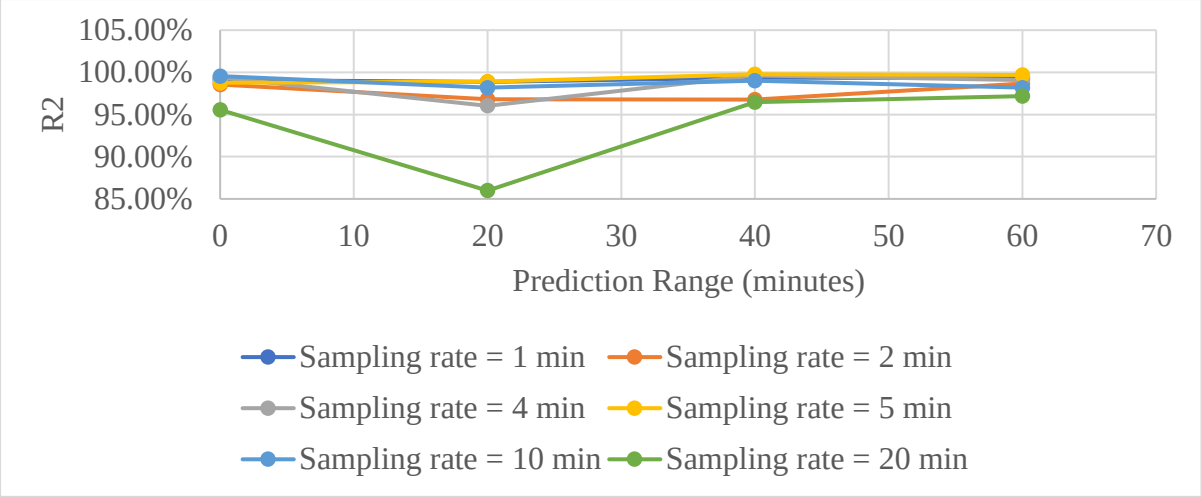
Figure 4.13. Prediction performance with different ranges in different sampling rates measured by a) R^2 b) RMSE c) MAE

It is expected that the prediction accuracy drops gradually as the data collection frequency becomes lower. However, as the results showed, the accuracies did not change significantly between data collected every 1 minute and every 5 or even 10 minutes. After 10 minutes the accuracies decreased comparatively more significantly for most of the scenarios. It is worth noting that the accuracies fluctuated over the course of interval increments from 1 to 10 minutes rather than being a linear decrease. The results show that a lower data collection frequency improved the prediction accuracy. This is expected and could have resulted from the data collection frequencies being close and the following sources of uncertainties:

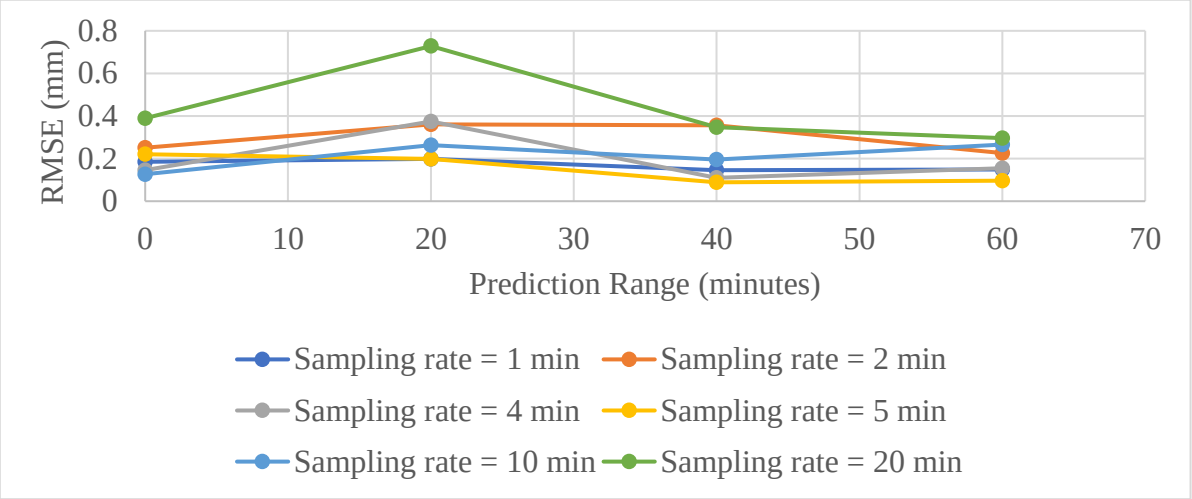
- Fluctuations in the sensor data,
- ML model's uncertainty due to the random sampled training data, as well as
- The instability and uncertainty in the data transmission system during the test despite being in a controlled environment.

The variations in the predictions again demonstrate the need to quantify the uncertainties in the RDT environment to incorporate various sources of error by following a probabilistic approach.

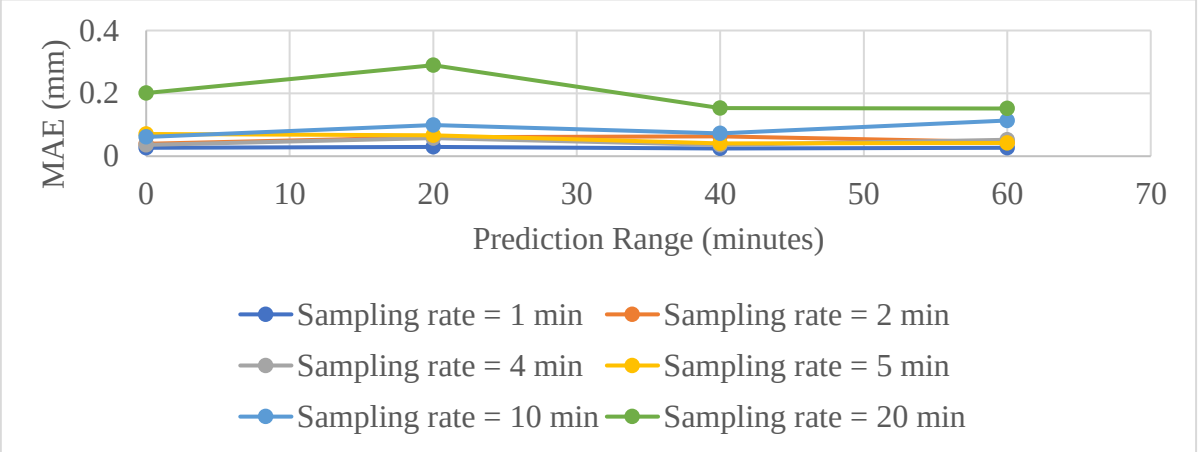
Figure 4.14 presents the prediction accuracies for each sampling rate to predict different future ranges. Similarly, the results do not vary significantly as the prediction horizons grows. Most scenarios also present a decrease in prediction performance when predicting the value for the next 20 minutes, then followed by an increase in prediction accuracy when predicting the next 40 minutes, which could also be due to uncertainty in data.



(a)



(b)



(c)

Figure 4.14. Performance of predictions using different sampling rates across different ranges, measured by a) R^2 b) RMSE c) MAE

As presented in Figures 4.13 and 4.14, the prediction accuracy showed a decreasing trend as the data collection interval increases from 1 minute to 20 minutes. However, the prediction accuracies remained acceptable given the pre-defined thresholds (Section 3.6.2.3). Taking the RMSE as the criterion, it increased from a minimum error of 0.146mm for a frequency of every 1 minute, to a maximum error of 0.728mm for a frequency of every 20 minutes across the prediction ranges, still within the 0.790mm threshold.

To balance the accuracy and data collection frequency, it can be said that for this case study, collecting sensor data at a rate of every 20 minutes would produce predictive models with sufficient accuracy instead of every minute. However, this sampling rate is not a recommendation but a demonstration of the proposed ML-based trial-and-error approach to better inform data collection practice and strategy as part of a predictive RDT with the consideration of reduced need for data storage, processing power, and less energy consumption overall.

This 95% reduction in data collection interval, would result in less quantity of data required to be generated by sensors or to be stored somewhere. In this NBIF experiment, just for the strain gauge sensor itself which produces 1613 records per second, a frequency of collecting data every 20 minutes compared to every minute would save 19 minutes' data with the continuous monitoring, that is in total 19*60*1613 = 1838843 data records in one cycle of data collection, equivalent to 229 megabytes in storage. Considering the total hours of two experiments, it can be quantified that approximately 60 gigabytes of data were saved with the identified frequency.

In addition, to measure the strain, the only meaningful value to obtain is the maximum and minimum strain during one loading cycle (see Figure 3.15(b)), this indicates a constant measurement might not be required, and even more proportion of data could be saved. This

would significantly decrease the volume of redundant data, and all the associated costs. For example, the costs of collecting, processing, storing and handling data, and their integration with different software and platforms could be reduced, as well as the on-going operational costs on the sensor network. As a result, a more efficient data acquisition system considering different technology management options can be configured and set up. In addition, it would help reducing the required power and thereby potentially reducing carbon dioxide emissions to ensure a sustainable digital development.

As mentioned previously, a sampling rate of every 4 minutes achieved better overall accuracy across different metrics compared to that of every minute. And sampling the data every 5 minutes has also obtained a comparable prediction accuracy to every minute. In fact, the highest R^2 and the lowest RMSE were both obtained when sampling the data every 5 minutes and predicting the reading for the next 40 minutes. Hence, the data collection frequency could be adjusted to every 5 minutes if the highest prediction accuracy was required. Therefore, the optimum choice of the sampling rate could be dynamic and based on the users' needs and expectations. In a real-world scenario, the thresholds could be levelled up if the prediction accuracy was the priority of the DT application.

The findings of this case study align with existing studies which attempted to optimise sampling rate based on the time-series sensor data in terms of efficient management of data storage, processing, energy cost (Van Wyk et al., 2017; Haidar et al., 2019; Čulić Gambiroža et al., 2022). Especially considering the on-going pavement digitalisation and future road DT environment where roads are embedded with sensors and technologies that collect and transmit real-time monitoring data, this study provides a useful reference and demonstration on the initial and long-term economic benefits of leveraging sampling optimisation strategy on the time-series sensor data for the efficient and sustainable data management.

As mentioned by Waqar Haider et al. (2010), the data collection frequency has a direct impact on performance prediction with historical data. The findings in this case study using sensor data could also provide insights and analysis when translating to a real-world road scenario considering data collection. In this case study, a data collection frequency of 20 minutes was found acceptable in achieving the required prediction accuracy mentioned in Section 3.6.2.3. By considering the total amount of time taken in one experiment (e.g., around 54.5 hours in Experiment 1), 164 data records were produced with the chosen frequency. Given a road is estimated to last for around 40 years (Nunn and Ferne, 2001), the result is equivalent to having a reading around four or five times each year for the same features. With this analysis, sensors could be installed and configured to capture readings at such a relatively low frequency instead of generating readings all the time, which would result in reduced energy consumption, and thereby decreasing cost in sensor management, data transmission and storage. In addition, from the pavement whole lifecycle management perspective, this could serve as a replacement for current road surveys practices with manual inspections or automated vehicles which are labour intensive and expensive while improving the accuracy of monitoring and predictions (Coenen and Golroo, 2017). However, the differences between a network of sensors, and the associated point-based condition data, and the data collected over the length of a road network using the current road surveys should be noted.

It is also worth acknowledging that there are multiple differences between the NBIF experiment and real-world roads. For instance, the simulated traffic load using an actuator does not have the same effect as the vehicles on the roads. Secondly, the ambient temperature was approximately constant as the test was conducted in a controlled lab environment

whereas on the real road, the temperature during the day and night would be quite different. These factors limit the generalisation of the NBIF case study.

4.4 Overall Discussions

This thesis has described the key components of an RDT framework that can be used as a reference for building RDTs and their interactions. The focus of the RDT framework in this thesis has been centred around data and modelling across different layers. This section provides a general discussion on the RDT for pavement management, and its development based on the results obtained in this study. In addition, a summary of the research and its value is provided.

4.4.1 RDT for Future PMS

An RDT could be used as a PMS to manage the condition monitoring and maintenance management of roads in the future. In the first case study of this research, a hybrid modelling approach was developed to probabilistically predict long term future road condition applying the RDT framework. This has demonstrated that an RDT has the capacity to produce pavement performance models that can make highly reliable and accurate predictions given the data being fed to it. Uncertainty quantification as part of the RDT modelling has taken consideration of errors and low quality in data, and in manually or automatically collected sensor data. Its performance is better than purely data-driven ML algorithms as revealed in this case study. Therefore, it is logical to expect that an RDT would significantly improve the predictive capability of a future PMS, considering that multiple studies (DT Thube, 2012; Sanabria et al., 2017; Abdelaziz et al., 2018) have reported the superior performance that a ML model has over traditional methods such as predictions using HDM-4 distress models. An advantage of RDT-based models is that they can be directly built using the data that is available from the road, rather than going through sophisticated local calibrations required for

traditional models. An RDT-based road condition prediction architecture has already been proposed in a review study conducted by Chen et al. (2022), while highlighting the potential adoption of RDTs for future pavement management across the whole pavement lifecycle at the same time.

As briefly mentioned in Chapter 2 where eight key elements of pavement management systems have been described, different functionalities apart from pavement performance prediction, could be potentially integrated into an RDT to supply the corresponding functions that would achieve the same goal. For instance, to perform economic analysis, priority evaluation, and optimization related tasks, an RDT can leverage deep learning or RL algorithms to generate and analyse 'what-if' scenarios on maintenance planning, based on the available data from maintenance activities, the history of construction and asset owner policy, from finance, and user cost. As a result, this will enhance asset maintenance prioritisation at both network and project level, by achieving maintenance optimisation through considering the associated social and economic factors (Yao et al., 2020).

Maintenance optimisation could be achieved by determining the appropriate schedule to perform the most cost-effective maintenance treatment at the right location(s) of the road network and sections with the highest priorities. In terms of the data and information it produces, while traditional PMS uses static data, and therefore limited information, an RDT will incorporate data with the Big Data properties described as 3V's (high volume, high variety and high velocity) from multiple sources and create insights. For example, an RDT could have a function of pavement health monitoring, making use of data from a variety of sensors such as pressure cells, deflectometers, strain gauges, thermocouples, moisture sensors, fibre-optic sensors, cameras, non-destructive testing surveys or other IoT devices with varying frequencies, to enable a constant monitoring of the pavement. This in turn

enables detection and classification of the formation of road distresses such as reflective cracks (Di Graziano et al., 2020; Majidifard et al., 2020; Alzraiee et al., 2021) depending on the algorithms embedded in the RDT.

An RDT is a 3D visualisation of the physical road. The data collected, the monitoring and the outputs based on the internal AI will also be represented visually for management purposes. Given there are already multiple efforts in developing GIS-based and BIM-based PMS, whereas at the same time, GIS and BIM are seen as the basis for building a DT (Meschini et al., 2022; Shi et al., 2023). Hence, an RDT is the natural progression of a platform that incorporates all different elements to enable a visual-aid advanced PMS. Moreover, in recent years new data and technologies have been used in the civil engineering domain, e.g., GPS data (Shahandashti et al., 2011; Im et al., 2013) and laser scanned LiDAR point cloud data (Laefer, 2020), therefore, a 3D model of road conditions can be developed. The model can help achieve real-time monitoring and display of the road condition and its surrounding environment. An RDT integrates the 3D visual information of all the roads, with other functions such as condition data collection, processing, and ML modelling embedded into the same platform, thereby producing a holistic 3D decision-making support tool for road authorities and transportation agencies (Dong et al., 2021).

As data is at the core of any DT (Zhang et al., 2022), the success of an RDT would also rely on having good data. The sharing of multiple data across different departments and even organisations would be the main challenge due to data privacy and data security issues (Marai et al., 2020). The technical integration of data from different sources also poses a challenge for an RDT (Botín-Sanabria et al., 2022). In addition, a heavy reliance on the data might incur unnecessary costs such as data storage, network bandwidth and processing power, and complexity as well as risks (Hu et al., 2014).

4.4.2 Development of RDT

Although this study did not directly address the development of an RDT, the RDT framework proposed in Chapter 3 provides a blueprint for an RDT creation. As applications of the RDT framework, the results and outputs produced in the two case studies would help to contribute to knowledge and understanding when considering building an RDT. As DTs can be created for different purposes at multiple stages of a physical asset, this section focuses on discussing how to develop an RDT for an improved management practice in the maintenance phase. There are many factors to consider when trying to create an RDT such as 1) existing available data 2) the purpose of the RDT 3) targeted user of the RDT 4) connectivity between data and RDT 5) additional data capturing and storage 6) data processing and computing 7) technologies involved such as AI, ML algorithms, as well as platform and applications 8) visualisation.

From the data perspective, to develop an RDT, the process should start with data. For the road sector, which is not traditionally data rich, one can start with limited historical data when real-time data is not yet available. Adopting this RDT framework with existing available data, the purpose and functionality of the RDT can be defined, as well as its target users. Taking the first case study as an example, if historical pavement condition data were available, a simple DT-based pavement performance model could be produced. Even when historical data was not sufficient, the publicly available data such as LTPP used in this study could be used to make initial predictions given the similar attributes of the roads and their surrounding environments. The model could be used by road asset owners, transportation agencies and/or local authorities that oversee managing the roads. Such models are essentially AI-based and supported by the most suitable ML algorithm. The selection of the algorithm can be done automatically depending on the input data, and there are multiple

open-source packages that provide this capability such as Auto-Sklearn (Feurer et al., 2022), and PyCaret (https://pycaret.org). The chosen model can ideally be then integrated at an appropriate level (depending on the algorithm) with the road's corresponding physics-based model where existing pavement expert knowledge could be leveraged. The developed ML model accuracy achieved could be used as a baseline to inform the design and planning of sensor data collection frequency when it comes to real-time sensor data considering costs. Apart from two methods introduced in this study to combine ML with physics, other physics-informed ML techniques could also be used to fully realise the potential of the hybrid model utilising both the data and the existing knowledge (Willard et al., 2020). Moreover, according to the RDT framework, existing pavement knowledge could be integrated with the ML modelling process in the form of numerical models. Various traditional simulation software could be used such as Abaqus and KENPAVE, which are common tools used for pavement analysis, making it easy to implement. Embedding such a model in an RDT platform would produce not only a higher accuracy in prediction but also an increased interpretability and reliability because of the pavement physical elements involved (Wu et al., 2024). With this novel approach applied in road management, it would help the pavement community improve the existing understanding of the pavement structure deterioration mechanism and potentially identify the cause of defects as the data size increases (Rizvi and Abbas, 2023).

Real-time data can be separately added to the RDT when it is available. The prediction model can be trained first on the historical data and then refined using real-time data. When it comes to real-time data, the associated costs for data capturing, storage and processing, and computational power are often concerns when developing a DT, while considering limited budgets. The prevailing industry view on RDT currently is that its associated costs outweigh its benefits (Davletshina and Brilakis, 2024). However, this research, especially the insights

from Case study II helps to understand and potentially decrease the amount of data needed for building an RDT to avoid unnecessary cost while still taking the advantages brought by the DT technology. Case study II indicates that a low frequency sensor data or data with a relatively low sampling rate for road assets, in a normal scenario, would be sufficient to train a ML model with acceptable prediction accuracy. In addition, considering the modelling capacity demonstrated in Case study I, this accuracy could be improved by the integration with physics within an RDT. Therefore, such data collection strategy for realising RDT can resolve the issue of data redundancy and the direct cost for the data storage, processing and computing. In addition, this would potentially reduce the indirect cost of energy consumption, CO_2 emission and fossil fuel consumption produced by sensor network installation, configuration as well as data clusters.

More capital investment can be made on selecting the optimal software and platforms to generate the 3D model and integrate different data formats and technologies into one RDT for scenario analysis and visualisation. This sensor data collection frequency may increase if the road has a high criticality, and it is deteriorating faster than expected. This would mean a more accurate predictive model is needed. In this case, higher frequency with higher cost would be inevitable.

4.4.3 Summary of the Research

The research carried out in this project addressed the defined aim and objectives in Section 1.2 and they can be summarised as follows:

1. **The development and application of a DT-based decision-making support theoretical framework for road lifecycle**

Based on the literature review in Chapter 2 on road management and DTs, a decision-making support theoretical framework was developed to investigate the enabling methods for a

predictive DTs on roads, especially with a focus on road condition modelling as well as the frequency of data collection. This framework was then applied to two case studies.

2. Assessment on the performance of the modelling capacity of RDT compared to ML

Based on the literature review in Chapter 2 on different techniques for modelling pavement performance, a research gap was identified to overcome the limitations of current ML models and approaches in predicting the road condition, and to incorporate domain knowledge into the ML process. According to the theoretical framework, an RDT modelling methodology (Section 3.5) with two approaches to combine ML with physics-based simulations as well as uncertainty quantification was proposed to investigate the potential improvement over the pure ML method. This was tested using the publicly available US LTPP database.

3. Optimised data collection frequency using experiment sensor data in NBIF

As the literature review (Section 2.2.6) revealed that most data used for RDT research has been image data, or 3D point cloud data or text-based log data, while very limited study has investigated the numerical condition data. There was a need to optimise the data collection frequency in the context of RDT development. With the data produced by the sensors instrumented in an experiment in NBIF, a large amount of sensor data was generated and used for analysing the balance between prediction accuracy and the data sampling rate. An iterative modelling approach (Section 3.6.2.3) was developed and identified that a data collection frequency of every 20 minutes would be sufficient to achieve an acceptable predictive capacity, resulting in a 95% reduction data that is required. For strain gauge alone in this experiment, approximately 60 gigabytes out of 64.8 gigabytes of data were saved.

4.4.4 Value of the Research

When it comes to road pavement management, there is a continuous need to improve the pavement performance modelling and accuracy in order to make optimal decisions for road maintenance strategy in terms of maintenance type and timing. Therefore, the DT-based modelling approach as demonstrated by Case study I shows a great potential to provide a road condition prediction tool that is of the highest accuracy, reliability and sustainability. In addition, this piece of research also investigates how DT concepts could be used to design data collection frequency in the context of sensor data. Given the increasing popularity and benefits shown in sensing and instrumented pavements, a large amount and variety of data from pavements are expected. As presented in Case study II, a DT capability would help addressing common questions such as 1) how much data is needed to be collected, 2) how often sensors should collect data, by making use of the data and identifying a suitable frequency for collecting data from sensors considering their availabilities, wireless network connectivity and power resources. The NBIF experiment demonstrated that with the selected data collection interval, it saved roughly 60 gigabytes of data just for one strain gauge sensor. This can lead to a remarkable decrease in the amount of data required when considering the whole road network with sensors installed in multiple places, thereby reducing their costs. This study would encourage more studies to understand how DT could be used in innovating and improving current practices across different stages as part of road lifecycle management.

In terms of contribution to knowledge, the research advances the general science on the usage of DTs in pavement engineering. Especially, with the latest developments in pavement performance modelling being purely ML based approaches, this study, by investigating the enabling methods for predictive DT for roads, progresses this field to integrate physics into the ML predictive modelling process to enable more precise, realistic predictions. More

specifically, the innovation in ML is that most ML models have the limitation of being a black box, which makes it difficult to interpret how a ML algorithm such as a neural network produces the output. Hence, with the help of a DT which combines the physical deterioration process with ML, it may be possible to produce outputs that are more interpretable. This would help to understand the causality between model inputs and outputs by iteratively re-configuring physics-based simulation models based on ML model prediction results. Conclusions from the research as well as recommendations for future works are presented in the following chapter.

5 CONCLUSION AND FUTURE WORKS

This thesis has investigated the supporting techniques for predictive DTs and their potential applications on different aspects of road management. As a result, a decision-making support theoretical framework was developed using the literature for road asset. This study focused on the functions of **modelling pavement performance** and **data collection** (sampling interval optimisation) using the developed framework. The details of the findings and the main conclusions are provided in this chapter, as well as recommendations for future works.

5.1 Accomplished Work and Main Findings

As discussed in the previous chapters, this research has demonstrated the objectives outlined in Chapter 1 by:

1) Developing a DT-based decision-making support theoretical framework for road lifecycle after identifying the key components, elements and characteristics of DTs based on the literature review in Chapter 2.
2) Evaluating the modelling capacity of a predictive RDT through a case study using the US LTPP database as an application of the framework in Chapters 3 and 4.
3) Investigating the optimal data collection frequency through a second case study using experimental sensor data from instrumented pavement tests in Chapters 3 and 4.
4) Discussing the perspectives and recommendations for the use and development of an RDT for pavement management.

The detailed findings and conclusions in this thesis are as follows:

a. Pavement performance modelling perspective

- Various ML algorithms were used and tested on pavement condition data and achieved promising results, suggesting a superior performance compared to traditional methods according to literature.
- To predict rutting, this study presented a novel ML-based approach integrated with domain physical knowledge considering ML inherent uncertainties. A RF algorithm was used with hyperparameter tuning.
- Integration was achieved through fusion of a comprehensive dataset collected from a public database and outputs (pavement surface deflection) from physics-based FE simulations to serve as model inputs for the RF.
- RF model performances were compared in two scenarios 1) without the extra FE simulation data based on physics 2) with the extra FE simulation data based on physics. The results from Case study I show an improvement in R^2 (90.3% → 94.2%) with the additional FE simulation data, and a reduction in the uncertainty by 6.76% considering the 90% probability prediction range for the 12^{th} year prediction.
- To predict IRI, two different approaches were used to integrate physics with ML using an ANN: 1) adding the output of the physics-based FE simulation as an extra input to the ANN, and by 2) customising the ANN loss function to constraint the ANN training process on the up sampled data based on the FE simulation results. This resulted in more accurate and reliable predictions.

b. Data collection perspective

- Data collection is an important element for building an RDT. It was found that a lower data collection frequency does not necessarily reduce the model's prediction accuracy much as evidenced by the Case study II in Chapter 4. This resulted in a 95% reduction in the amount of data that would need to be collected. The results from this case study also suggested a continuous monitoring might not be needed which could lead to a very considerable reduction in data requirements.
- A low data collection interval (e.g., 4 or 5 times a year) in real-world is recommended for an RDT for a road carrying average traffic which deteriorates at an average rate. This minimises all the relevant costs for data storage, processing and computing power while still producing sufficient predictability due to the modelling capability as part of the proposed RDT framework.

c. Digital twin perspective

- RDT research is still at an early stage with existing studies focusing on the generation of automatic geometric RDT which is descriptive and informative based on geospatial pavement and surrounding data as per the literature review.
- A predictive RDT can improve the current practice of road asset management with the advancement of data, and enabling approaches such as AI and ML.
- This developed predictive DT framework including key components and elements can be implemented for different purposes given the context. For example, it can be used for predicting pavement defects (rutting and roughness) and optimising the data collection interval in this study but also for other scenarios such as defect labelling or classification and maintenance planning and prioritisation.

- A predictive RDT provides an advanced modelling approach for pavement performance bringing pavement engineering knowledge into the ML process, as described in the RDT modelling methodology presented in Chapter 3. The performances of an RDT model and a ML model have been evaluated and compared via a case study in Chapter 4, and it was found that the RDT model enhanced not only the prediction accuracy but also the rationality and reliability of the model to predict road rutting and roughness.

5.2 Recommendations for Further Research

While the results from Chapter 4 have demonstrated the feasibility of the RDT framework via the case studies, the research presented in this thesis is an early attempt to understand how DTs can contribute to the development and improvement of pavement management. Future work related to the thesis could include the following considerations:

a. DT-based Decision-Making Support Theoretical Framework for Roads

As the science of DT itself and its enabling technologies are evolving at a fast pace, the framework could be improved and expanded to consider more components and elements suitable for road asset management. In particular, the aspect of technical interfaces that enable the links between different data sources, types, formats, software and technologies could be added. In addition, the user that interacts with the RDT should also be taken into account as part of the framework showing how the users can access, evaluate and modify the RDT around their specific needs and expectations.

b. Data Sources

From a historical data perspective, more databases could be explored in addition to the US LTPP database. It would be worthwhile to incorporate other real-world data sources such as

transportation agencies' PMS databases, road defect image datasets, as well as non-destructive surveys such as ground penetrating radar.

From the real time data perspective, this study used data from instrumented pavement experiments at NBIF. Further work could be to conduct field trials to embed sensors to obtain readings on actual road sections and develop a living lab using various types and sources of data in the real world such as traffic, temperature, and climate, with a high granularity. A cost benefit analysis can be studied in collaboration with civil infrastructure asset owners on the trade-off between prediction accuracies and different sensor data collection frequencies as well as their associated cost to demonstrate RDT's capacity and benefit in lifecycle management of road maintenance. Furthermore, existing road condition data from connected vehicles could be another good option for data collection.

c. Modelling and Simulation

Regarding the actual "*brain"* of any DT where the data is turned into insights and knowledge, multiple future works can be carried out in this domain to increase the DT intelligence level. Firstly, more novel ML techniques could be used as the algorithm to model the data. Similarly, more numerical simulation tools could be adopted in a DT environment for road structural behaviour analysis. A more sophisticated and realistic physics-based model representing full pavement physics and characteristics with variations of different layers and its interactions with other assets could likely make a more significant improvement.

Secondly, while this study proposed two ways to integrate physics-based simulation into ML, further research is required to explore the interactions between different innovative ML methods and physics-based models to fully unlock the limitations of ML, as well as leveraging the existing engineering knowledge in pavement assets and materials. This could

lead to further scientific development in combining domain knowledge with ML in general. Finally, for RDT-based applications, different interdisciplinary approaches or collaborations would lead to more robust and innovative research outcomes.

In addition, given there are existing traditional tools that have been used in highways management for many years, such as HDM-4, a study could be conducted to compare the development and performance of RDT-based models vs. HDM-4 models, demonstrating the advantages and disadvantages of each model.

d. RDT Portal - 3D modelling

From a visualisation perspective, future works could include the generation of RDTs from not only point cloud data but also the geospatial data to enable it to be a maintenance planning decision-making support tool for road authorities. For example, the creation of a holistic management simulation of the road can be visualised by including traffic, pedestrian flows, pavement deterioration, climate, maintenance and economic cost. Using the visualisation tools, "what-if" scenarios can be developed to include cost benefit analysis on understanding the effect of road closures due to certain types of repairs needed. Moreover, further research could be done on 3D modelling at the defect level such as rutting or cracking to understand the root causes of defects and to predict their progression, enabled by real-time data and numerical models.

e. Road Lifecycle Management

Future studies could investigate how to make use of the RDT predictions to make more informed maintenance planning and prioritisation decisions for maintenance during an asset lifecycle, making use of the RDT-based model presented in this thesis. Maintenance treatment selections can be done based on the 90^{th} percentile range prediction outcomes with

a specific confidence interval. In addition, while this study did not use maintenance data as part of the modelling process, the maintenance history could be used to validate and make adjusted multi-year 90^{th} percentile range predictions when used as inputs to train the model to understand the effects of the selected maintenance repair.

Apart from the maintenance, to implement the RDT framework, future research could investigate the possibilities of developing an RDT for other stages of a road lifecycle such as design, construction and operation to incorporate the scientific advancement in these areas brought by the RDT and identify the benefits and improvements in comparison to existing practices and approaches.

6 REFERENCES

Aarasse, S. and Idelhakkar, B. (2023) 'Technological tools and the impact of digitalisation on the supply chain.' *European Journal of Economic and Financial Research*. Open Access Publishing Group, 7(4).

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X. and Brain, G. (2016) '{TensorFlow}: a system for {Large-Scale} machine learning.' *In the 12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283.

Abadías Llamas, A., Bartie, N. J., Heibeck, M., Stelter, M. and Reuter, M. A. (2020) 'Simulation-Based Exergy Analysis of Large Circular Economy Systems: Zinc Production Coupled to CdTe Photovoltaic Module Life Cycle.' *Journal of Sustainable Metallurgy*. Springer Science and Business Media Deutschland GmbH, 6(1) pp. 34–67.

Abaza, K. A., Ashur, S. A., Abu-Eisheh, S. A. and Rabay'a, A. (2001) 'Macroscopic Optimum System for Management of Pavement Rehabilitation.' *Journal of Transportation Engineering*. American Society of Civil Engineers, 127(6) pp. 493–500.

Abdelaziz, N., Abd El-Hakim, R. T., El-Badawy, S. M. and Afify, H. A. (2018) 'International Roughness Index prediction model for flexible pavements.' *International Journal of Pavement Engineering*. Taylor & Francis, 21(1) pp. 88–99.

Abdulmohsen, A., Alabdulkarim, O. and Thesis, P. (2013) 'Understanding the effects of different levels of product monitoring on maintenance operations: A simulation approach.'

Agatonovic-Kustrin, S. and Beresford, R. (2000) 'Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research.' *Journal of Pharmaceutical and Biomedical Analysis*. Elsevier, 22(5) pp. 717–727.

Aheleroff, S., Xu, X., Zhong, R. Y. and Lu, Y. (2021) 'Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model.' *Advanced Engineering Informatics*. Elsevier, 47, January, p. 101225.

Aivaliotis, P., Georgoulias, K. and Chryssolouris, G. (2019) 'The use of Digital Twin for predictive maintenance in manufacturing.' *International Journal of Computer Integrated Manufacturing*. Taylor & Francis, 32(11) pp. 1067–1080.

Ajakaiye, M. and Amin, S. (2020) 'Machine Learning Algorithms for Rutting Modelling of Bituminous Pavements in West Midlands.' *In Conference 19th Annual International Conference on Highways and Airport Pavement Engineering, Asphalt Technology and Infrastructure*.

Al, R. and Sin, G. (2021) 'MOSKopt: A simulation-based data-driven digital twin optimizer with embedded uncertainty quantification.' *Computer Aided Chemical Engineering*. Elsevier, 50, January, pp. 649–654.

ALARM (2024) ‘ALARM Annual Local Authority Road Maintenance Survey Report.’

Alaswadko, N. and Hassan, R. (2018) ‘Rutting progression models for light duty pavements.’ *International Journal of Pavement Engineering*, 19(1) pp. 37–47.

Al Alawi, A. M., Al Shuaili, H. H., Al-Naamani, K., Al Naamani, Z. and Al-Busafi, S. A. (2024) ‘A Machine Learning-Based Mortality Prediction Model for Patients with Chronic Hepatitis C Infection: An Exploratory Study.’ *Journal of Clinical Medicine*. Multidisciplinary Digital Publishing Institute, 13(10) p. 2939.

Albayati, A. H. (2023) ‘A review of rutting in asphalt concrete pavement.’ *Open Engineering*. De Gruyter Open Ltd, 13(1) p. 20220463.

Allen, B. D. (2021) ‘Digital Twins and Living Models at NASA.’ *In Digital Twin Summit*.

Almér, H. (2015) ‘Machine learning and statistical analysis in fuel consumption prediction for heavy vehicles.’

Alnaqbi, A. J., Zeiada, W., Al-Khateeb, G. G., Hamad, K. and Barakat, S. (2023) ‘Creating Rutting Prediction Models through Machine Learning Techniques Utilizing the Long-Term Pavement Performance Database.’ *Sustainability*. Multidisciplinary Digital Publishing Institute, 15(18) p. 13653.

Al-Quraba, A. K. and Idrees, A. K. (2017) ‘Adaptive Data Collection protocol for Extending Lifetime of Periodic Sensor Networks.’ *Qalaai Zanist Journal*. Lebanese French University, 2(2) pp. 93–103.

Al-Sabaeei, A. M., Souliman, M. I. and Jagadeesh, A. (2024) ‘Smartphone applications for pavement condition monitoring: A review.’ *Construction and Building Materials*. Elsevier, 410, January, p. 134207.

Alzabeebee, S., Alshkane, Y. M., Al-Taie, A. J. and Rashed, K. A. (2021) ‘Soft computing of the recompression index of fine-grained soils.’ *Soft Computing*. Springer Science and Business Media Deutschland GmbH, 25(24) pp. 15297–15312.

Alzabeebee, S., Chapman, D. N. and Faramarzi, A. (2018) ‘Development of a novel model to estimate bedding factors to ensure the economic and robust design of rigid pipes under soil loads.’ *Tunnelling and Underground Space Technology*. Pergamon, 71, January, pp. 567–578.

Alzraiee, H., Ruiz, A. L. and Sprotte, R. (2021) ‘Detecting of Pavement Marking Defects Using Faster R-CNN.’ *Journal of Performance of Constructed Facilities*. American Society of Civil Engineers, 35(4) p. 04021035.

Amarh, E. A. (2017) ‘Evaluating the Mechanical Properties and Long-Term Performance of Stabilized Full-Depth Reclamation Base Materials.’ *Doctoral dissertation, Virginia Tech*. Virginia Tech, May.

American Association of State Highway and Transportation Officials (2002) ‘AASHTO Transportation Asset Management Guide.’ *AASHTO, Washington, D.C., 2002*.

Amin, S. R. and Amador-Jiménez, L. E. (2017) ‘Backpropagation Neural Network to estimate pavement performance: dealing with measurement errors.’ *Road Materials and Pavement Design*. Taylor and Francis Ltd., 18(5) pp. 1218–1238.

Amirhossein Hosseini, S. (2020) ‘Data-driven framework for modeling deterioration of pavements in the state of Iowa the state of Iowa.’ *Doctoral dissertation, Iowa State University*.

Anderson, B., Hy, T. S. and Kondor, R. (2019) ‘Cormorant: Covariant Molecular Neural Networks.’ *Advances in Neural Information Processing Systems*. Neural information processing systems foundation, 32, June.

Anyala, M. (2011) ‘Investigation of the impact of climate change on road maintenance.’ *Doctoral dissertation, University of Birmingham*.

Anyala, M., Odoki, J. B. and Baker, C. J. (2014) ‘Hierarchical asphalt pavement deterioration model for climate impact studies.’ *International Journal of Pavement Engineering*, 15(3) pp. 251–266.

Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., Naidich, D. P. and Shetty, S. (2019) ‘End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography.’ *Nature Medicine*. Nature Publishing Group, 25(6) pp. 954–961.

Arisekola, K. and Madson, K. (2023) ‘Digital twins for asset management: Social network analysis-based review.’ *Automation in Construction*. Elsevier, 150, June, p. 104833.

Arsić, M. (2020) ‘Impact of digitalisation on economic growth, productivity and employment.’ *Economic Themes*, 58(4) pp. 431–457.

Atkinson, V., Merrill, D. and Thom, N. (2006) *Pavement wear factors*.

Attaran, M. and Celik, B. G. (2023) ‘Digital Twin: Benefits, use cases, challenges, and opportunities.’ *Decision Analytics Journal*. Elsevier, 6, March, p. 100165.

Babanagar, N., Sheil, B., Ninić, J., Zhang, Q. and Hardy, S. (2025) ‘Digital twins for urban underground space.’ *Tunnelling and Underground Space Technology*. Pergamon, 155, January, p. 106140.

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B. and Zhang, J. D. (2020) ‘An Introduction to Machine Learning.’ *Clinical pharmacology & therapeutics*, 107(4) pp. 871–885.

Bado, M. F., Tonelli, D., Poli, F., Zonta, D. and Casas, J. R. (2022) ‘Digital Twin for Civil Engineering Systems: An Exploratory Review for Distributed Sensing Updating.’ *Sensors*. Multidisciplinary Digital Publishing Institute, 22(9) p. 3168.

Bae Kim, H., Buch, N. and Park, D. Y. (2000) ‘Mechanistic-Empirical Rut Prediction Model for In-Service Pavements.’ *Transportation Research Record*, 1730(1) pp. 99–109.

Barbedo, J. G. A. (2019) ‘A Review on the Use of Unmanned Aerial Vehicles and Imaging Sensors for Monitoring and Assessing Plant Stresses.’ *Drones*. Multidisciplinary Digital Publishing Institute, 3(2) p. 40.

Bashar, M. Z. and Torres-Machi, C. (2021) ‘Performance of Machine Learning Algorithms in Predicting the Pavement International Roughness Index.’ *Transportation Research Record*. SAGE Publications, 2675(5) pp. 226–237.

Berber, A. and Srećković, S. (2024) ‘When something goes wrong: Who is responsible for errors in ML decision-making?’ *AI and Society*. Springer Science and Business Media Deutschland GmbH, 39(4) pp. 1891–1903.

Berthelot, C. F. (2020) *A Proposed Framework for Optimized Road Maintenance*.

Bhandari, S., Luo, X. and Wang, F. (2023) ‘Understanding the effects of structural factors and traffic loading on flexible pavement performance.’ *International Journal of Transportation Science and Technology*. Elsevier, 12(1) pp. 258–272.

Bhatta, S. and Dang, J. (2024) ‘Use of IoT for structural health monitoring of civil engineering structures: a state-of-the-art review.’ *Urban Lifeline*. Springer, 2(1) pp. 1–19.

Biancardo, S. A., Gesualdi, M., Savastano, D., Intignano, M., Henke, I. and Pagliara, F. (2023) ‘An innovative framework for integrating Cost-Benefit Analysis (CBA) within Building Information Modeling (BIM).’ *Socio-Economic Planning Sciences*. Pergamon, 85, February, p. 101495.

Blasco, T., Sánchez, J. S. and García, V. (2024) ‘A survey on uncertainty quantification in deep learning for financial time series prediction.’ *Neurocomputing*. Elsevier, 576, April, p. 127339.

Bolton, A., Butler, L., Dabson, I., Enzer, M., Evans, M., Fenemore, T., Harradence, F., Keaney, E., Kemp, A., Luck, A., Pawsey, N., Saville, S., Schooling, J., Sharp, M., Smith, T., Tennison, J., Whyte, J., Wilson, A. and Makri, C. (2018) ‘Gemini Principles.’ CDBB.

Boström, H., Andler, S. F., Brohede, M., Johansson, R., Karlsson, A., van Laere, J., Niklasson, L., Nilsson, M., Persson, A. and Ziemke, T. (2007) ‘On the Definition of Information Fusion as a Field of Research.’ Institutionen för kommunikation och information.

Botín-Sanabria, D. M., Mihaita, S., Peimbert-García, R. E., Ramírez-Moreno, M. A., Ramírez-Mendoza, R. A. and Lozoya-Santos, J. de J. (2022) ‘Digital Twin Technology Challenges and Applications: A Comprehensive Review.’ *Remote Sensing*. Multidisciplinary Digital Publishing Institute, 14(6) p. 1335.

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., Levine, S. and Vanhoucke, V. (2018) ‘Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping.’ *2018 IEEE international conference on robotics and automation (ICRA)*. Institute of Electrical and Electronics Engineers Inc., September, pp. 4243–4250.

Bral, S., Kumar, P. P. and Chopra, T. (2024) ‘Prediction of International Roughness Index Using CatBooster and Shap Values.’ *International Journal of Pavement Research and Technology*. Springer, 17(2) pp. 518–533.

Broo, D. G. and Schooling, J. (2021) ‘Digital twins in infrastructure: definitions, current practices, challenges and strategies.’ *International Journal of Construction Management*. Taylor & Francis, 23(7) pp. 1254–1263.

Burningham, S. and Stankevich, N. (2005) ‘Why road maintenance is important and how to get it done.’ *The World Bank*, Transport Note(121) pp. 535–546.

Cabrera, M., Ninic, J. and Tizani, W. (2023) ‘Fusion of experimental and synthetic data for reliable prediction of steel connection behaviour using machine learning.’ *Engineering with Computers*. Springer Science and Business Media Deutschland GmbH, 39(6) pp. 3993–4011.

Callcut, M., Cerceau Agliozzo, J. P., Varga, L. and McMillan, L. (2021) ‘Digital Twins in Civil Infrastructure Systems.’ *Sustainability*. Multidisciplinary Digital Publishing Institute, 13(20) p. 11549.

Cano-Ortiz, S., Pascual-Muñoz, P. and Castro-Fresno, D. (2022) ‘Machine learning algorithms for monitoring pavement performance.’ *Automation in Construction*. Elsevier, 139, July, p. 104309.

Carter, A., Imtiaz, S. and Naterer, G. F. (2023) ‘Review of interpretable machine learning for process industries.’ *Process Safety and Environmental Protection*. Elsevier, 170, February, pp. 647–659.

Chakraborty, S. and Adhikari, S. (2021) ‘Machine learning based digital twin for dynamical systems with multiple time-scales.’ *Computers & Structures*. Pergamon, 243, January, p. 106410.

Chang, C. M., Cheng, D. X., Smith, R. E., Tan, S. G. and Hossain, A. (2024) ‘SMART quality control analysis of pavement condition data for pavement management applications.’ *International Journal of Transportation Science and Technology*. Elsevier, June.

Chang, C. and Zeng, T. (2023) ‘A hybrid data-driven-physics-constrained Gaussian process regression framework with deep kernel for uncertainty quantification.’ *Journal of Computational Physics*. Academic Press, 486, August, p. 112129.

Chen, D. and Mastin, N. (2016) ‘Sigmoidal Models for Predicting Pavement Performance Conditions.’ *Journal of Performance of Constructed Facilities*. American Society of Civil Engineers (ASCE), 30(4) p. 04015078.

Chen, F., Xu, S., Zhao, Y. and Zhang, H. (2020) ‘An Adaptive Genetic Algorithm of Adjusting Sensor Acquisition Frequency.’ *Sensors (Basel, Switzerland)*. MDPI AG, 20(4) p. 990.

Chen, G. H. and Shah, D. (2018) ‘Explaining the Success of Nearest Neighbor Methods in Prediction.’ *Foundations and Trends® in Machine Learning*. Now Publishers, Inc., 10(5–6) pp. 337–588.

Chen, J., Di Liu, Wang, C., Zhang, W. and Zhu, L. (2022) ‘A fractal contact model of rough surfaces considering detailed multi-scale effects.’ *Tribology International*. Elsevier, 176, December, p. 107920.

Chen, K., Torbaghan, M. E., Chu, M., Zhang, L. and Garcia, A. (2022) ‘Identifying the most suitable machine learning approach for a road digital twin.’ *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, 174(3) pp. 88–101.

Chen, K., Torbaghan, M. E., Thom, N., Garcia-Hernández, A., Faramarzi, A. and Chapman, D. (2024) ‘A Machine Learning based approach to predict road rutting considering uncertainty.’ *Case Studies in Construction Materials*. Elsevier, April, p. e03186.

Chen, W. (2023) ‘Asphalt Pavement Performance Prediction Based on K-Nearest Neighbor Algorithm.’ *In 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*. IEEE, pp. 1–4.

Chen, W. and Brilakis, I. (2023) ‘Developing Digital Twin Data Structure and Integrated Cloud Digital Twin Architecture for Roads.’ *In Computing in Civil Engineering 2023*, pp. 424–432.

Chen, Z., Xiao, F., Guo, F. and Yan, J. (2023) ‘Interpretable machine learning for building energy management: A state-of-the-art review.’ *Advances in Applied Energy*. Elsevier, 9, February, p. 100123.

Choi, S. and Do, M. (2020) ‘Development of the road pavement deterioration model based on the deep learning method.’ *Electronics (Switzerland)*. MDPI AG, 9(1) p. 3.

Chollet, F. (2018) ‘Keras: The python deep learning library.’ *Astrophysics source code library*. GitHub p. ascl1806.

Cicirello, A. (2024) ‘Physics-Enhanced Machine Learning: a position paper for dynamical systems investigations.’ *arXiv preprint arXiv:2405.05987*.

von Coburg, F., Westbeld, J., Buchmann, E. and Höfer, P. (2024) ‘Investigation of a perturbation-based model updating approach for structural health and event monitoring.’ *Journal of Vibration and Control*. SAGE Publications Inc., 10775463241229476.

Coenen, T. B. J. and Golroo, A. (2017) ‘A review on automated pavement distress detection methods.’ *Cogent Engineering*. Cogent, 4(1) p. 1374822.

Consilvio, A., Solís Hernández, J., Chen, W., Brilakis, I., Bartoccini, L., Di Gennaro, F. and Van Welie, M. (2022) ‘Towards a digital twin-based intelligent decision support for road maintenance.’ *Transportation Research Procedia*, 69 pp. 791–798.

Corker, J., Remenyte-Prescott, R., Eskandari Torbaghan, M. and Ninic, J. (2023) ‘A GIS based approach for predicting pavement deterioration on the UK road network,’ July.

Cui, B. and Wang, H. (2024) ‘Predicting Asphalt Pavement Deterioration Under Climate Change Uncertainty Using Bayesian Neural Network.’ *IEEE Transactions on Intelligent Transportation Systems*. Institute of Electrical and Electronics Engineers Inc.

Čulić Gambiroža, J., Mastelić, T., Nižetić Kosović, I. and Čagalj, M. (2022) 'Dynamic monitoring frequency for energy-efficient data collection in Internet of Things.' *Journal of Computational Science*. Elsevier, 64, October, p. 101842.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M. and Piccialli, F. (2022) 'Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next.' *Journal of Scientific Computing*. Springer, 92(3) p. 88.

Damirchilo, F, Hosseini, A., Mellat Parast, M. and Fini, E. H. (2021) 'Machine Learning Approach to Predict International Roughness Index Using Long-Term Pavement Performance Data.' *Journal of Transportation Engineering, Part B: Pavements*. American Society of Civil Engineers, 147(4) p. 04021058.

Davletshina, D. and Brilakis, I. (2024) 'A Review on Constructing and Maintaining Geometric Digital Twins of Highways.' *Transportation Research Board 103rd Annual Meeting Transportation Research Board*, (TRBAM-24-00258).

Davletshina, D., Reja, V. and Brilakis, I. (2024) 'Capturing Reality Changes from Point Clouds for Updating Road Geometric Digital Twins.' *In EC3 Conference 2024*. European Council on Computing in Construction, pp. 0–0.

Daw, A., Karpatne, A., Watkins, W. D., Read, J. S. and Kumar, V. (2022) 'Physics-Guided Neural Networks (PGNN): An Application in Lake Temperature Modeling.' *In Knowledge-Guided Machine Learning*. Chapman and Hall/CRC, pp. 353–372.

Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P. and Karpatne, A. (2019) 'Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling.' *In Proceedings of the 2020 siam international conference on data mining*. Society for Industrial and Applied Mathematics Publications, pp. 532–540.

Dawes, B., Hunt, M., Meah, N., Kudryavtsev, A. and Evans, R. (2019) 'Physics-Based Simulation in Support of a Through-Life Gas Turbine Service Business Model.' *Proceedings of the ASME Turbo Expo*. American Society of Mechanical Engineers Digital Collection, 1, November.

Dayananda, B., Owen, S., Kolobaric, A., Chapman, J. and Cozzolino, D. (2023) 'Pre-processing Applied to Instrumental Data in Analytical Chemistry: A Brief Review of the Methods and Examples.' *Critical Reviews in Analytical Chemistry*. Taylor & Francis pp. 1–9.

Deeba, F., Islam, M., Bui, F. M. and Wahid, K. A. (2018) 'Performance assessment of a bleeding detection algorithm for endoscopic video based on classifier fusion method and exhaustive feature selection.' *Biomedical Signal Processing and Control*. Elsevier, 40, February, pp. 415–424.

Deng, Y., Wang, H. and Shi, X. (2024) 'Physics-guided neural network for predicting asphalt mixture rutting with balanced accuracy, stability and rationality.' *Neural Networks*. Pergamon, 172, April, p. 106085.

Deryabin, S. A., Temkin, I. O. and Zykov, S. V. (2020) ‘About some issues of developing Digital Twins for the intelligent process control in quarries.’ *Procedia Computer Science*. Elsevier, 176, January, pp. 3210–3216.

Dihan, M. S., Akash, A. I., Tasneem, Z., Das, P., Das, S. K., Islam, M. R., Islam, M. M., Badal, F. R., Ali, M. F., Ahamed, M. H., Abhi, S. H., Sarker, S. K. and Hasan, M. M. (2024) ‘Digital twin: Data exploration, architecture, implementation and future.’ *Heliyon*. Elsevier, 10(5) p. e26503.

Ding, J. and Brilakis, I. (2023) ‘The potential for creating a geometric digital twin of road surfaces using photogrammetry and computer vision.’ *In Proceedings of the European Conference on Computing in Construction*. European Council on Computing in Construction, pp. 0–0.

Dong, J., Meng, W., Liu, Y. and Ti, J. (2021) ‘A framework of pavement management system based on IoT and big data.’ *Advanced Engineering Informatics*. Elsevier Ltd, 47, January, p. 101226.

Dongare, A. D., Kharde, R. R. and Kachare, A. D. (2012) ‘Introduction to artificial neural network.’ *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1) pp. 189–194.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. and Koltun, V. (2017) ‘CARLA: An Open Urban Driving Simulator.’ *Conference on robot learning*, November, pp. 1–16.

DT Thube (2012) ‘Artificial neural network (ANN) based pavement deterioration models for low volume roads in India.’ *International Journal of Pavement Research and Technology*, 5(2) p. 115.

Duggal, R., Gupta, N., Pandya, A., Mahajan, P., Sharma, K., kaundal, T. and Angra, P. (2022) ‘Building structural analysis based Internet of Things network assisted earthquake detection.’ *Internet of Things*. Elsevier B.V., 19 p. 100561.

Duran, M. C., Ninic, J., Tizani, W. and Wang, F. (2022) ‘Machine learning-based fusion of experimental and synthetic data for reliable prediction of steel connection stiffness.’ *UKACM 2022 Conference*.

Eça, L. and Hoekstra, M. (2014) ‘A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies.’ *Journal of Computational Physics*. Academic Press, 262, April, pp. 104–130.

Editorial (2024) ‘The increasing potential and challenges of digital twins.’ *Nature Computational Science*, 4(3) pp. 145–146.

Efe, S. and Shokouhian, M. (2020) ‘Proposal on Implementing Machine Learning with Highway Datasets.’ *International Journal of Engineering Research and Technology*, 9(5) pp. 189–194.

Eghrari, Z., Delavar, M. R., Zare, M., Mousavi, M., Nazari, B. and Ghaffarian, S. (2023) ‘Groundwater level prediction using deep recurrent neural networks and uncertainty

assessment.' *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10 pp. 493–500.

El-Gawady, A., Makhlouf, M. A., Tawfik, B. S. and Nassar, H. (2022) 'Machine Learning Framework for the Prediction of Alzheimer's Disease Using Gene Expression Data Based on Efficient Gene Selection.' *Symmetry*. Multidisciplinary Digital Publishing Institute, 14(3) p. 491.

Elhamod, M., Bu, J., Singh, C., Redell, M., Ghosh, A., Podolskiy, V., Lee, W. C. and Karpatne, A. (2022) 'CoPhy-PGNN: Learning Physics-guided Neural Networks with Competing Loss Functions for Solving Eigenvalue Problems.' *ACM Transactions on Intelligent Systems and Technology*. Association for Computing Machinery, 13(6).

Enders, M. and Hoßbach, N. (2019) 'Dimensions of Digital Twin Applications - A Literature Review.' *In Twenty-fifth Americas Conference on Information Systems, Cancun, 2019*.

Erdogdu, F., Datta, A., Vitrac, O., Marra, F., Verboven, P., Sarghini, F. and Nicolai, B. (2022) 'Mathematical modeling—Computer-aided food engineering.' *Food Engineering Innovations Across the Food Supply Chain*. Academic Press, January, pp. 277–290.

Errandonea, I., Beltrán, S. and Arrizabalaga, S. (2020) 'Digital Twin for maintenance: A literature review.' *Computers in Industry*. Elsevier B.V. p. 103316.

Faghmous, J. H. and Kumar, V. (2014) 'A Big Data Guide to Understanding Climate Change: The Case for Theory-Guided Data Science.' *Big Data*. Mary Ann Liebert Inc., 2(3) pp. 155–163.

Fähndrich, J. (2023) 'A literature review on the impact of digitalisation on management control.' *Journal of Management Control*. Springer Science and Business Media Deutschland GmbH, 34(1) pp. 9–65.

Farhat, C., Geuzaine, P. and Brown, G. (2003) 'Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter.' *Computers & Fluids*. Pergamon, 32(1) pp. 3–29.

Faroughi, S. A., Pawar, N. M., Raissi, M., Das, S., Kalantari, N. K. and Kourosh Mahjour, S. (2024) 'Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks and Operators in Scientific Computing: Fluid and Solid Mechanics.' *Journal of Computing and Information Science in Engineering*, 24(4) p. 040802.

Fathi, A., Mazari, M., Saghafi, M., Hosseini, A. and Kumar, S. (2019) 'Parametric study of pavement deterioration using machine learning algorithms.' *In Airfield and highway pavements*. American Society of Civil Engineers (ASCE), pp. 31–41.

Felsberger, A., Qaiser, F. H., Choudhary, A. and Reiner, G. (2022) 'The impact of Industry 4.0 on the reconciliation of dynamic capabilities: evidence from the European manufacturing industries.' *Production Planning & Control*. Taylor & Francis, 33(2–3) pp. 277–300.

Ferreira, A., De Picado-Santos, L., Wu, Z. and Flintsch, G. (2010) 'International Journal of Pavement Engineering Selection of pavement performance models for use in the Portuguese

PMS Selection of pavement performance models for use in the Portuguese PMS.' *International Journal of Pavement Engineering*, 12(1) pp. 87–97.

Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M. and Hutter, F. (2022) 'Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning.' *Journal of Machine Learning Research*. Microtome Publishing, 23(261) pp. 1–61.

Fitch, E. C. (2013) *Proactive maintenance for mechanical systems (No. 5)*. Elsevier.

Flintsch, G. W., Zaniewski, J. P. and Delton, J. (1996) 'Artificial Neural Network for Selecting Pavement Rehabilitation Projects.' *Transportation Research Record: Journal of the Transportation Research Board*. SAGE Publications, 1524(1) pp. 185–193.

Freitas, C. J. (2002) 'The issue of numerical uncertainty.' *Applied Mathematical Modelling*. Elsevier, 26(2) pp. 237–248.

Fuller, A., Fan, Z., Day, C. and Barlow, C. (2020) 'Digital Twin: Enabling Technologies, Challenges and Open Research.' *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 8 pp. 108952–108971.

Funari, M. F., Hajjat, A. E., Masciotta, M. G., Oliveira, D. V. and Lourenço, P. B. (2021) 'A Parametric Scan-to-FEM Framework for the Digital Twin Generation of Historic Masonry Structures.' *Sustainability*. Multidisciplinary Digital Publishing Institute, 13(19) p. 11088.

Fwa, T. F., Chan, W. T. and Hoque, K. Z. (1998) 'Network level programming for pavement management using genetic algorithms.' *In Proc. 4th International Conference on Managing Pavements*, pp. 815–829.

Gallup, E., Gallup, T. and Powell, K. (2023) 'Physics-guided neural networks with engineering domain knowledge for hybrid process modeling.' *Computers & Chemical Engineering*. Pergamon, 170, February, p. 108111.

Gardner, P., Dal Borgo, M., Ruffini, V., Hughes, A. J., Zhu, Y. and Wagg, D. J. (2020) 'Towards the Development of an Operational Digital Twin.' *Vibration*. Multidisciplinary Digital Publishing Institute, 3(3) pp. 235–265.

George, K. P., Rajagopal, A. S. and Lim, L. K. (1989) 'Models for Predicting Pavement Deterioration.' *Transportation Research Record* p. 1215.

Georgiou, P., Plati, C. and Loizos, A. (2018) 'Soft Computing Models to Predict Pavement Roughness: A Comparative Study.' *Advances in Civil Engineering*. Hindawi Limited, 2018(1).

Gogoi, R., Das, A. and Chakroborty, P. (2020) 'Rut depth measurement of an asphalt pavement from its original profile.' *Australian Journal of Civil Engineering*. Taylor & Francis, 18(2) pp. 119–125.

Gong, H., Cheng, S., Chen, Z. and Li, Q. (2022) 'Data-Enabled Physics-Informed Machine Learning for Reduced-Order Modeling Digital Twin: Application to Nuclear Reactor Physics.' *Nuclear Science and Engineering*. Taylor & Francis, 196(6) pp. 668–693.

Gong, H., Sun, Y., Mei, Z. and Huang, B. (2018) 'Improving accuracy of rutting prediction for mechanistic-empirical pavement design guide with deep neural networks.' *Construction and Building Materials*. Elsevier, 190, November, pp. 710–718.

Gong, H., Sun, Y., Shu, X. and Huang, B. (2018) 'Use of random forests regression for predicting IRI of asphalt pavements.' *Construction and Building Materials*. Elsevier Ltd, 189, November, pp. 890–897.

Gössling, S., Neger, C., Steiger, R. and Bell, R. (2023) 'Weather, climate change, and transport: a review.' *Natural Hazards*. Springer Science and Business Media B.V., 118(2) pp. 1341–1360.

Di Graziano, A., Marchetta, V. and Cafiso, S. (2020) 'Structural health monitoring of asphalt pavements using smart sensor networks: A comprehensive review.' *Journal of Traffic and Transportation Engineering (English Edition)*. Elsevier, 7(5) pp. 639–651.

Gregory, J., Noshadravan, A., Swei, O., Xu, X. and Kirchain, R. (2017) 'The importance of incorporating uncertainty into pavement life cycle cost and environmental impact analyses.' *Pavement Life-Cycle Assessment* pp. 131–142.

Grieves, M. (2014) 'Digital Twin: Manufacturing Excellence through Virtual Factory Replication.' *White paper*, 1(2014) pp. 1–7.

Gu, Y., Zhang, C. and Golub, M. V. (2022) 'Physics-informed neural networks for analysis of 2D thin-walled structures.' *Engineering Analysis with Boundary Elements*. Elsevier, 145, December, pp. 161–172.

Guo, F., Gregory, J. and Kirchain, R. (2020) 'Incorporating cost uncertainty and path dependence into treatment selection for pavement networks.' *Transportation Research Part C: Emerging Technologies*. Pergamon, 110, January, pp. 40–55.

Guo, J., Zhao, N., Sun, L. and Zhang, S. (2019) 'Modular based flexible digital twin for factory design.' *Journal of Ambient Intelligence and Humanized Computing*. Springer Verlag, 10(3) pp. 1189–1200.

Guo, R., Fu, D. and Sollazzo, G. (2022) 'An ensemble learning model for asphalt pavement performance prediction based on gradient boosting decision tree.' *International Journal of Pavement Engineering*. Informa UK Limited, 23(10) pp. 3633–3646.

Haas, R. and Hudson, W. R. (1987) 'Future Prospects for Pavement Management.' *In Proc., Second North Am. Conf. on Managing Pavements*, pp. 1–3.

Haas, R., Hudson, W. R. and Tighe, S. (2001) 'Maximizing customer benefits as the ultimate goal of pavement management.' *In 5th International Conference on Managing Pavements*, pp. 1–14.

Haas, R., Hudson, W. and Zaniewski, J. (1994) *Modern Pavement Management*. *Krieger Publishing Company*.

Haber, E. and Ruthotto, L. (2017) 'Stable architectures for deep neural networks.' *Inverse Problems*. IOP Publishing, 34(1) p. 014004.

Haddad, A. J., Chehab, G. R. and Saad, G. A. (2022) ‘The use of deep neural networks for developing generic pavement rutting predictive models.’ *International Journal of Pavement Engineering*, 23(12) pp. 4260–4276.

Haidar, N., Tamani, N., Nienaber, F., Wesseling, M. T., Bouju, A. and Ghamri-Doudane, Y. (2019) ‘Data collection period and sensor selection method for smart building occupancy prediction.’ *In 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, pp. 1–6.

Haider, S. W., Chatti, K., Baladi, G. Y. and Sivaneswaran, N. (2011) ‘Impact of pavement monitoring frequency on pavement management system decisions.’ *Transportation Research Record*. SAGE PublicationsSage CA: Los Angeles, CA pp. 43–55.

Hajek, J. J. and Phang, W. A. (1988) *Prioritization and Optimization of Pavement Preservation Treatments*. *Ontario. Ministry of Transportation. Research and Development Branch.* ONTARIO. MINISTRY OF TRANSPORTATION. RESEARCH AND DEVELOPMENT BRANCH.

Hamdi, Hadiwardoyo, S. P., Correia, A. G., Pereira, P. and Cortez, P. (2017) ‘Prediction of surface distress using neural networks.’ *In AIP Conference Proceedings*. AIP Publishing.

Han, C., Ma, T., Xu, G., Chen, S. and Huang, R. (2020) ‘Intelligent decision model of road maintenance based on improved weight random forest algorithm.’ *International Journal of Pavement Engineering*. Taylor & Francis, 23(4) pp. 985–997.

Hancock, J. T., Wang, H., Khoshgoftaar, T. M. and Liang, Q. (2024) ‘Data reduction techniques for highly imbalanced medicare Big Data.’ *Journal of Big Data*. Springer Nature, 11(1) p. 8.

Hao, C., Wang, Z., Zou, Y. and Zhao, Z. (2023) ‘Self-learning Time-varying Digital Twin System for Intelligent Monitoring of Automatic Production Line.’ *Journal of Physics: Conference Series*. IOP Publishing, 2456(1) p. 012021.

Harb, H. and Makhoul, A. (2017) ‘Energy-efficient sensor data collection approach for industrial process monitoring.’ *IEEE Transactions on Industrial Informatics*. IEEE Computer Society, 14(2) pp. 661–672.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T. E. (2020) ‘Array programming with NumPy.’ *Nature*, 585(7825) pp. 357–362.

Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M. and Hussain, A. (2024) ‘Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence.’ *Cognitive Computation*. Springer, 16(1) pp. 45–74.

Haykin, S. (1999) ‘Neural networks: a comprehensive foundation.’ *The Knowledge Engineering Review*, 13(4) pp. 409–412.

Haywood-Alexander, M., Arcieri, G., Kamariotis, A. and Chatzi, E. (2024) ‘Response Estimation and System Identification of Dynamical Systems via Physics-Informed Neural Networks.’ *arXiv preprint arXiv:2410.01340*.

Haywood-Alexander, M., Liu, W., Bacsa, K., Lai, Z. and Chatzi, E. (2024) ‘Discussing the spectrum of physics-enhanced machine learning: a survey on structural mechanics applications.’ *Data-Centric Engineering*. Cambridge University Press, 5, October, p. e30.

He, Q. Z., Barajas-Solano, D., Tartakovsky, G. and Tartakovsky, A. M. (2020) ‘Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport.’ *Advances in Water Resources*. Elsevier, 141, July, p. 103610.

Heggie, I. G. (1995) ‘Management and financing of roads: An agenda for reform.’ *In Publication of: Transportation Research Forum*.

Heidari, M. R., Heravi, G. and Esmaeeli, A. N. (2020) ‘Integrating life-cycle assessment and life-cycle cost analysis to select sustainable pavement: A probabilistic model using managerial flexibilities.’ *Journal of Cleaner Production*. Elsevier, 254, May, p. 120046.

Hicks, G. and Groeger, J. (2001) ‘Pavement Management Practices in State Highway Agencies: Newington, Connecticut Peer Exchange Results.’ *(No. FHWA-HIF-11-036). United States. Federal Highway Administration.*

Hinchy, E. P., Carcagno, C., O’Dowd, N. P. and McCarthy, C. T. (2020) ‘Using finite element analysis to develop a digital twin of a manufacturing bending operation.’ *Procedia CIRP*. Elsevier, 93, January, pp. 568–574.

Hosseini, S. A. and Smadi, O. (2021) ‘How prediction accuracy can affect the decision-making process in pavement management system.’ *Infrastructures*. MDPI AG, 6(2) p. 28.

Hou, S. and Wu, G. (2019) ‘A low-cost IoT-based wireless sensor system for bridge displacement monitoring.’ *Smart Materials and Structures*. Institute of Physics Publishing, 28(8) p. 085047.

Hu, H., Wen, Y., Chua, T. S. and Li, X. (2014) ‘Toward scalable systems for big data analytics: A technology tutorial.’ *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 2 pp. 652–687.

Hu, W., Zhang, T., Deng, X., Liu, Z. and Tan, J. (2021) ‘Digital twin: a state-of-the-art review of its enabling technologies, applications and challenges.’ *Journal of Intelligent Manufacturing and Special Equipment*. Emerald Publishing Limited, 2(1) pp. 1–34.

Huang, Y. (2004) *Pavement analysis and design*.

Hunt, P. D. and Bunker, J. M. (2002) ‘Road performance studies using roughness progression & pavement maintenance costs.’ *In Queensland Department of Main Roads Roads System and Engineering Forum*.

Hunter, J. D. (2007) ‘Matplotlib: A 2D graphics environment.’ *Computing in Science & Engineering*. IEEE COMPUTER SOC, 9(3) pp. 90–95.

Hussam, R., Supervisor, R., Emad, D., Co-Supervisor, D. and Issa, A. (2018) ‘Designof an Integrated Pavement Management Systems with Geographic Information Systems.’ *Doctoral dissertation, An-Najah National University*.

Ibraheem, A. T. and Atia, N. S. (2016) ‘Applying decision making with analytic hierarchy process (AHP) for maintenance strategy selection of flexible pavement.’ *The Global Journal of Researches in Engineering*, 16(5) pp. 25–34.

Ibrahim, A., Eltawil, A., Na, Y. and El-Tawil, S. (2019) ‘A Machine Learning Approach for Structural Health Monitoring Using Noisy Data Sets.’ *IEEE Transactions on Automation Science and Engineering*. Institute of Electrical and Electronics Engineers (IEEE), 17(2) pp. 900–908.

Im, S. B., Hurlebaus, S. and Kang, Y. J. (2013) ‘Summary Review of GPS Technology for Structural Health Monitoring.’ *Journal of Structural Engineering*. American Society of Civil Engineers (ASCE), 139(10) pp. 1653–1664.

Inzerillo, L., Di Mino, G. and Roberts, R. (2018) ‘Image-based 3D reconstruction using traditional and UAV datasets for analysis of road pavement distress.’ *Automation in Construction*. Elsevier, 96, December, pp. 457–469.

Jafari Ahangari, H. (2014) ‘Deterioration modelling of low volume roads in Australia using laser profilometer measurements.’ *Doctoral dissertation, Swinburne*. Swinburne, January.

Jakkula, V. (2006) ‘Tutorial on Support Vector Machine (SVM).’ *School of EECS, Washington State University*, 37(2.5) p. 3.

Javaid, M., Haleem, A. and Suman, R. (2023) ‘Digital Twin applications toward Industry 4.0: A Review.’ *Cognitive Robotics*. Elsevier, 3, January, pp. 71–92.

Jaya, P., Masri, A., Al-Saffar, Z. H., Sandamal, K., Shashiprabha, S., Muttil, N. and Rathnayake, U. (2023) ‘Pavement Roughness Prediction Using Explainable and Supervised Machine Learning Technique for Long-Term Performance.’ *Sustainability*. Multidisciplinary Digital Publishing Institute, 15(12) p. 9617.

Jia, X., Willard, J., Karpatne, A., Read, J. S., Zwart, J. A., Steinbach, M. and Kumar, V. (2021) ‘Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles.’ *ACM/IMS Transactions on Data Science*. ACMPUB27New York, NY, 2(3) pp. 1–26.

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M. and Kumar, V. (2019) ‘Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles.’ *In Proceedings of the 2019 SIAM international conference on data mining*. Society for Industrial and Applied Mathematics, pp. 558–566.

Jiang, F., Ma, L., Broyd, T., Chen, W. and Luo, H. (2022) ‘Building digital twins of existing highways using map data based on engineering expertise.’ *Automation in Construction*. Elsevier, 134, February, p. 104081.

Jin, Y. and Mukherjee, A. (2014) ‘Markov chain applications in modelling facility condition deterioration.’ *International Journal of Critical Infrastructures*. Inderscience Publishers, 10(2) pp. 93–112.

Jones, D., Snider, C., Nassehi, A., Yon, J. and Hicks, B. (2020) ‘Characterising the Digital Twin: A systematic literature review.’ *CIRP Journal of Manufacturing Science and Technology*. Elsevier Ltd, 29, May, pp. 36–52.

Joni, H. H., Hilal, M. M. and Abed, M. S. (2020) ‘Developing International Roughness Index (IRI) Model from visible pavement distresses.’ *In IOP Conference Series: Materials Science and Engineering*. IOP Publishing, p. 012119.

Justo-Silva, R., Ferreira, A. and Flintsch, G. (2021) ‘Review on Machine Learning Techniques for Developing Pavement Performance Prediction Models.’ *Sustainability*, 13(9) p. 5248.

Kaewunruen, S. and Lian, Q. (2019) ‘Digital twin aided sustainability-based lifecycle management for railway turnout systems.’ *Journal of Cleaner Production*. Elsevier, 228, August, pp. 1537–1551.

Kaliske, M., Behnke, R. and Wollny, I. (2021) ‘Vision on a Digital Twin of the Road-Tire-Vehicle System for Future Mobility.’ *Tire Science and Technology*. Allen Press, 49(1) pp. 2–18.

Kanaga Priya, P. and Reethika, A. (2024) ‘A Review of Digital Twin Applications in Various Sectors.’ *Transforming Industry using Digital Twin Technology*. Springer, Cham pp. 239–258.

Kapteyn, M. G., Knezevic, D. J., Huynh, D. B. P., Tran, M. and Willcox, K. E. (2022) ‘Data-driven physics-based digital twins via a library of component-based reduced-order models.’ *International Journal for Numerical Methods in Engineering*. John Wiley & Sons, Ltd, 123(13) pp. 2986–3003.

Kapteyn, M. G. and Willcox, K. E. (2020) ‘From Physics-Based Models to Predictive Digital Twins via Interpretable Machine Learning.’ *arXiv preprint arXiv:2004.11356*, April.

Kar, S., Tanaka, R., Iwata, H., Kholova, J., Durbha, S. S., Adinarayana, J. and Vadez, V. (2020) ‘Multi-Scale Time Series Analysis of Evapotranspiration for High-Throughput Phenotyping Frequency Optimization.’ *In 2020 IEEE Latin American GRSS and ISPRS Remote Sensing Conference (LAGIRS)*. Institute of Electrical and Electronics Engineers Inc., pp. 98–103.

Kara De Maeijer, P., Luyckx, G., Vuye, C., Voet, E., Van Den Bergh, W., Vanlanduit, S., Braspenninckx, J., Stevens, N. and De Wolf, J. (2019) ‘Fiber Optics Sensors in Asphalt Pavement: State-of-the-Art Review.’ *Infrastructures*. Multidisciplinary Digital Publishing Institute, 4(2) p. 36.

Karan, M. (1978) ‘Municipal Pavement Management System.’

Karanam, G. D., Goenaga, B. and Underwood, B. S. (2023) ‘Quantifying Uncertainty with Pavement Performance Models: Comparing Bayesian and Non-Parametric Methods.’ *Transportation Research Record*. SAGE Publications Ltd, 2677(7) pp. 661–679.

Kargah-Ostadi, N., Stoffels, S. M. and Tabatabaee, N. (2010) ‘Network-Level Pavement Roughness Prediction Model for Rehabilitation Recommendations.’ *Transportation Research Record*. SAGE PublicationsSage CA: Los Angeles, CA, 1(2155) pp. 124–133.

Kargah-Ostadi, N., Vasylevskyi, K., Ablets, A. and Drach, A. (2024) ‘Physics-informed neural networks to advance pavement engineering and management.’ *Road Materials and Pavement Design*. Taylor & Francis, February, pp. 1–22.

Kargah-Ostadi, N., Vasylevskyi, Kostiantyn, Ablets, A. and Drach, A. (2024) ‘Reconciling Pavement Condition Data from Connected Vehicles with the International Roughness Index from Standard Monitoring Equipment Using Physics-Integrated Machine Learning.’ *Transportation Research Record*. SAGE Publications Ltd, 2678(2) pp. 416–429.

Kargah-Ostadi, N., Zhou, Y. (Mina) and Rahman, T. (2019) ‘Developing Performance Prediction Models for Pavement Management Systems in Local Governments in Absence of Age Data.’ *Transportation Research Record*. SAGE PublicationsSage CA: Los Angeles, CA, 2673(3) pp. 334–341.

Karimi, M. N. and Mallick, R. B. (2023) ‘Flexible Pavement Instrumentation: A State-of-the-Art Review.’ *Journal of Transportation Engineering, Part B: Pavements*. American Society of Civil Engineers, 149(2) p. 03123001.

Karimzadeh, A. (2020) ‘Predictive Analytics for Roadway Maintenance: A Review of Current Models, Challenges, and Opportunities.’ *Civil Engineering Journal*, 6(3) pp. 602–625.

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N. and Kumar, V. (2017) ‘Theory-guided data science: A new paradigm for scientific discovery from data.’ *IEEE Transactions on Knowledge and Data Engineering*. IEEE Computer Society, 29(10) pp. 2318–2331.

Kaur, M. J., Mishra, V. P. and Maheshwari, P. (2020) ‘The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action.’ *In Digital twin technologies and smart cities*. Springer International Publishing, pp. 3–17.

Kerali, H. and Odoki, J. (2006) ‘HDM-4 Highway Development and Management. Volume 4: Analytical Framework and Model Descriptions.’ *The Highway Development and Management Series, 4*.

Kerali, H. R. (2001) ‘The role of HDM-4 in road management.’ *In Proceedings, First Road Transportation Technology Transfer Conference in Africa, Ministry of Works, Tanzania*, pp. 320–333.

Kheradmandi, N. and Mehranfar, V. (2022) ‘A critical review and comparative study on image segmentation-based techniques for pavement crack detection.’ *Construction and Building Materials*. Elsevier, 321, February, p. 126162.

Khoei, T. T. and Singh, A. (2024) ‘Data reduction in big data: a survey of methods, challenges and future directions.’ *International Journal of Data Science and Analytics*. Springer Science and Business Media Deutschland GmbH, July, pp. 1–40.

Khosravi, A., Nahavandi, S., Creighton, D. and Atiya, A. F. (2011) ‘Comprehensive review of neural network-based prediction intervals and new advances.’ *IEEE Transactions on Neural Networks*, 22(9) pp. 1341–1356.

Kibira, D. and Shao, G. (2023) ‘Data Requirements for A Digital Twin of A Robot Workcell.’ *Proceedings - Winter Simulation Conference*. Institute of Electrical and Electronics Engineers Inc. pp. 3272–3283.

Kim, H. C., Kim, M. H. and Choe, D. E. (2019) ‘Structural health monitoring of towers and blades for floating offshore wind turbines using operational modal analysis and modal properties with numerical-sensor.’ *Ocean Engineering*, 188(106226).

Kim, Y. M., Jung, D., Chang, Y. and Choi, D. H. (2019) ‘Intelligent micro energy grid in 5G era: Platforms, business cases, testbeds, and next generation applications.’ *Electronics*, 8(4) p. 468.

Kirkwood, B. R. and Sterne, J. A. (2010) *Essential medical statistics*. *John Wiley & Sons*.

Kırbaş, U. (2018) ‘IRI Sensitivity to the Influence of Surface Distress on Flexible Pavements.’ *Coatings*. Multidisciplinary Digital Publishing Institute, 8(8) p. 271.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S. and Willing, C. (2016) ‘Jupyter Notebooks – a publishing format for reproducible computational workflows.’ *In Positioning and power in academic publishing: Players, agents and agendas*. IOS press, pp. 87–90.

Kochunas, B. and Huan, X. (2021) ‘Digital Twin Concepts with Uncertainty for Nuclear Power Applications.’ *Energies*. Multidisciplinary Digital Publishing Institute, 14(14) p. 4235.

Kohavi, R. (1995) ‘A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.’ *In International Joint Conferences on Artificial Intelligence*, pp. 1137–1145.

Kong, J. and Yuan, J. Y. (2010) ‘Application of linear viscoelastic differential constitutive equation in ABAQUS.’ *In 2010 International Conference On Computer Design and Applications*. IEEE, pp. 5–152.

Kritzinger, W., Karner, M., Traar, G., Henjes, J. and Sihn, W. (2018) ‘Digital Twin in manufacturing: A categorical literature review and classification.’ *IFAC-PapersOnLine*. Elsevier, 51(11) pp. 1016–1022.

Kulkarni, R. B. and Miller, R. W. (2003) ‘Pavement Management Systems Past, Present, and Future.’ *Transportation Research Record*, 1853(1) pp. 65–71.

De La Garza, J. M., Akyildiz, S., Bish, D. R. and Krueger, D. A. (2011) 'Network-level optimization of pavement maintenance renewal strategies.' *Advanced Engineering Informatics*. Elsevier, 25(4) pp. 699–712.

Laborie, F., Røed, O. C., Engdahl, G. and Camp, A. (2019) 'Extracting Value from Data Using an Industrial Data Platform to Provide a Foundational Digital Twin.' *In Offshore Technology Conference*. OTC, p. D011S010R005.

Laefer, D. F. (2020) 'Harnessing Remote Sensing for Civil Engineering: Then, Now, and Tomorrow.' *Lecture Notes in Civil Engineering*. Springer, Singapore, 33 pp. 3–30.

Lee, K. and Lee, H. M. (2017) 'Numerical analysis and modeling for crack width calculation using IoT in reinforced concrete members.' *Journal of Ambient Intelligence and Humanized Computing*. Springer Verlag, 9(4) pp. 1119–1130.

Lee, T. H., Ullah, A. and Wang, R. (2020) 'Bootstrap Aggregating and Random Forest.' *Macroeconomic forecasting in the era of big data: Theory and practice*. Springer pp. 389–429.

Leung, H. and Haykin, S. (1991) 'The Complex Backpropagation Algorithm.' *IEEE Transactions on Signal Processing*, 39(9) pp. 2101–2104.

Li, N., Xie, W.-C. and Haas, R. (1996) 'Reliability-Based Processing of Markov Chains for Modeling Pavement Network Deterioration.' *Transportation Research Record*. SAGE Publications, 1524(1) pp. 203–213.

Li, Q. and Chan, M. F. (2017) 'Predictive time-series modeling using artificial neural networks for Linac beam symmetry: an empirical study.' *Annals of the New York Academy of Sciences*. John Wiley & Sons, Ltd, 1387(1) pp. 84–94.

Li, T., Rui, Y., Zhu, H., Lu, L. and Li, X. (2024) 'Comprehensive digital twin for infrastructure: A novel ontology and graph-based modelling paradigm.' *Advanced Engineering Informatics*. Elsevier, 62, October, p. 102747.

Li, W., Bazant, M. Z. and Zhu, J. (2021) 'A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches.' *Computer Methods in Applied Mechanics and Engineering*. North-Holland, 383, September, p. 113933.

Lin, J. F., Li, X. Y., Wang, J., Wang, L. X., Hu, X. X. and Liu, J. X. (2021) 'Study of Building Safety Monitoring by Using Cost-Effective MEMS Accelerometers for Rapid After-Earthquake Assessment with Missing Data.' *Sensors 2021, Vol. 21, Page 7327*. Multidisciplinary Digital Publishing Institute, 21(21) p. 7327.

Lin, L., Bao, H. and Dinh, N. (2021) 'Uncertainty quantification and software risk analysis for digital twins in the nearly autonomous management and control systems: A review.' *Annals of Nuclear Energy*. Pergamon, 160, September, p. 108362.

Ling, J., Kurzawski, A. and Templeton, J. (2016) ‘Reynolds averaged turbulence modelling using deep neural networks with embedded invariance.’ *Journal of Fluid Mechanics*. Cambridge University Press, 807, November, pp. 155–166.

Lipp, J., Rudack, M., Vroomen, U. and Bührig-Polaczek, A. (2020) ‘When to Collect What? Optimizing Data Load via Process-driven Data Collection.’ *ICEIS (1)* pp. 220–225.

Liu, B., Zhan, L. M., Xu, L., Ma, L., Yang, Y. and Wu, X. M. (2021) ‘Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering.’ *In 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 1650–1654.

Liu, D., Jiang, W., Mu, L. and Wang, S. (2020) ‘Streamflow Prediction Using Deep Learning Neural Network: Case Study of Yangtze River.’ *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 8 pp. 90069–90086.

Liu, L., Yang, D. Y. and Frangopol, D. M. (2020) ‘Probabilistic cost-benefit analysis for service life extension of ships.’ *Ocean Engineering*. Pergamon, 201, April, p. 107094.

Liu, M., Fang, S., Dong, H. and Xu, C. (2021) ‘Review of digital twin about concepts, technologies, and industrial applications.’ *Journal of Manufacturing Systems*. Elsevier, 58, January, pp. 346–361.

Liu, Z., Meyendorf, N. and Mrad, N. (2018) ‘The role of data fusion in predictive maintenance using digital twin.’ *AIP Conference Proceedings*. American Institute of Physics Inc., 1949(1) p. 33.

Lo, C. K., Chen, C. H. and Zhong, R. Y. (2021) ‘A review of digital twin in product design and development.’ *Advanced Engineering Informatics*. Elsevier, 48, April, p. 101297.

Loske, D. and Klumpp, M. (2022) ‘Verifying the effects of digitalisation in retail logistics: an efficiency-centred approach.’ *International Journal of Logistics Research and Applications*. Taylor & Francis, 25(2) pp. 203–227.

Loulizi, A., Al-Qadi, I. L., Lahouar, S. and Freeman, T. E. (2001) ‘Data collection and management of instrumented smart road flexible pavement sections.’ *Transportation Research Record*. National Research Council, 1769(1) pp. 142–151.

Lu, Q., Parlikad, A. K., Woodall, P., Don Ranasinghe, G., Xie, X., Liang, Z., Konstantinou, E., Heaton, J. and Schooling, J. (2020) ‘Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus.’ *Journal of Management in Engineering*. American Society of Civil Engineers (ASCE), 36(3) p. 05020004.

Lu, Q., Xie, X., Heaton, J., Parlikad, A. K. and Schooling, J. (2020) ‘From BIM towards digital twin: Strategy and future development for smart asset management.’ *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2019 9*. Springer Verlag, October, pp. 392–404.

Lunn, D. J., Thomas, A., Best, N. and Spiegelhalter, D. (2000) ‘WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility.’ *Statistics and Computing*. Springer Netherlands, 10(4) pp. 325–337.

Luo, P., Asce, S. M., Parn, E., Kookalani, S., Brilakis, I. and Asce, M. (2024) 'TailorAlert: Large Language Model-Based Personalized Alert Generation System for Road Infrastructure Management with Digital Twins.'

Luo, X., Gong, H., Tao, J., Wang, F., Minifie, J. and Qiu, X. (2022) 'Improving Data Quality of Automated Pavement Condition Data Collection: Summary of State of the Practices of Transportation Agencies and Views of Professionals.' *Journal of Transportation Engineering, Part B: Pavements*. American Society of Civil Engineers (ASCE), 148(3) p. 04022042.

Macchi, M., Roda, I., Negri, E. and Fumagalli, L. (2018) 'Exploring the role of Digital Twin for Asset Lifecycle Management.' *IFAC-PapersOnLine*. Elsevier B.V., 51(11) pp. 790–795.

Macías, A., Muñoz, D., Navarro, E. and González, P. (2024) 'Data fabric and digital twins: An integrated approach for data fusion design and evaluation of pervasive systems.' *Information Fusion*. Elsevier, 103, March, p. 102139.

Madni, A. M., Madni, C. C. and Lucero, S. D. (2019) 'Leveraging Digital Twin Technology in Model-Based Systems Engineering.' *Systems*. Multidisciplinary Digital Publishing Institute, 7(1) p. 7.

Majidifard, H., Adu-Gyamfi, Y. and Buttlar, W. G. (2020) 'Deep Machine Learning Approach to Develop a New Asphalt Pavement Condition Index.' *Construction and building materials*, 247 p. 118513.

Mallick, R. B. and El-Korchi, T. (2008) *Pavement engineering: principles and practice*. *Pavement Engineering: Principles and Practice: Second Edition*. CRC Press.

Marai, O. El, Taleb, T. and Song, J. S. (2020) 'Roads Infrastructure Digital Twin: A Step Toward Smarter Cities Realization.' *IEEE Network*. Institute of Electrical and Electronics Engineers Inc., 35(2) pp. 136–143.

Marcelino, P., de Lurdes Antunes, M., Fortunato, E. and Gomes, M. C. (2019) 'Transfer learning for pavement performance prediction.' *International Journal of Pavement Research and Technology*. Springer, 13(2) pp. 154–167.

Marcelino, P., de Lurdes Antunes, M., Fortunato, E. and Gomes, M. C. (2021) 'Machine learning approach for pavement performance prediction.' *International Journal of Pavement Engineering*. Taylor and Francis Ltd., 22(3) pp. 341–354.

Marie d'Avigneau, A., Potseluyko, L., Anvo, N. R., Taha, H. M., Reja, V. K., Davletshina, D., Lam, P., de Silva, L., Al-Tabbaa, A. and Brilakis, I. (2025) 'CAMHighways: The Cambridge Highways dataset.' *Advanced Engineering Informatics*. Elsevier, 64, March, p. 103036.

Martin, T. and Choummanivong, L. (2016) 'The Benefits of Long-Term Pavement Performance (LTPP) Research to Funders.' *Transportation Research Procedia*. Elsevier, 14, January, pp. 2477–2486.

Matchett, R. and Wium, J. (2022) ‘Digital Twins for Road Infrastructure.’ *In 40th Annual Southern African Transport Conference*.

McKinney, W. (2010) ‘Data structures for statistical computing in python.’ *In SciPy*, pp. 51–56.

McNeil, S. and Humplick, F. (1991) ‘Evaluation of Errors in Automated Pavement Distress Data Acquisition.’ *Journal of Transportation Engineering*. American Society of Civil Engineers, 117(2) pp. 224–241.

Mehdi, M. A., Cherradi, T., Bouyahyaoui, A., El Karkouri, S. and Qachar, A. (2022) ‘Evolution of a flexible pavement deterioration, analyzing the road inspections results.’ *Materials Today: Proceedings*. Elsevier, 58, January, pp. 1222–1228.

Mehrotra, K., Mohan, C. K. and Ranka, S. (1997) *Elements of artificial neural networks*. MIT press.

Menzies, T., Port, D., Chen, Z., Hihn, J. and Stukes, S. (2005) ‘Validation Methods for Calibrating Software Effort Models.’ *In Proceedings of the 27th international conference on Software engineering*, pp. 587–595.

Meschini, S., Pellegrini, L., Locatelli, M., Accardo, D., Tagliabue, L. C., Di Giuda, G. M. and Avena, M. (2022) ‘Toward cognitive digital twins using a BIM-GIS asset management system for a diffused university.’ *Frontiers in Built Environment*. Frontiers Media S.A., 8, December, p. 959475.

Meža, S., Mauko Pranjić, A., Vezočnik, R., Osmokrović, I. and Lenart, S. (2021) ‘Digital Twins and Road Construction Using Secondary Raw Materials.’ *Journal of Advanced Transportation*. Hindawi Limited, 2021(1) p. 8833058.

Mihai, S., Yaqoob, M., Hung, D. V., Davis, W., Towakel, P., Raza, M., Karamanoglu, M., Barn, B., Shetve, D., Prasad, R. V., Venkataraman, H., Trestian, R. and Nguyen, H. X. (2022) ‘Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects.’ *IEEE Communications Surveys and Tutorials*. Institute of Electrical and Electronics Engineers Inc., 24(4) pp. 2255–2291.

Mohammadi, N. and Taylor, J. E. (2018) ‘Smart city digital twins.’ *In 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–5.

Mohapatra, A. G., Khanna, A., Gupta, D., Mohanty, M. and de Albuquerque, V. H. C. (2022) ‘An experimental approach to evaluate machine learning models for the estimation of load distribution on suspension bridge using FBG sensors and IoT.’ *Computational Intelligence*. John Wiley & Sons, Ltd, 38(3) pp. 747–769.

Morales, F. J., Reyes, A., Cáceres, N., Romero, L. M., Benitez, F. G., Morgado, J., Duarte, E. and Martins, T. (2017) ‘Historical maintenance relevant information road-map for a self-learning maintenance prediction procedural approach.’ *In IOP Conference Series: Materials Science and Engineering*. Institute of Physics Publishing, p. 012107.

Mubaraki, M. (2016) ‘Highway subsurface assessment using pavement surface distress and roughness data.’ *International Journal of Pavement Research and Technology*. 9(5) pp. 393–402.

Muntaka, A. S., Agyei-Owusu, B., Manso, J. F. and Kankam-Boadu, E. (2023) ‘Pursuing transport digitalisation to achieve transport cost optimisation.’ *In Supply Chain Forum: An International Journal*. Taylor & Francis, pp. 1–13.

Muralidhar, N., Bu, J., Cao, Z., He, L., Ramakrishnan, N., Tafti, D. and Karpatne, A. (2020) ‘Phynet: Physics guided neural networks for particle drag force prediction in assembly.’ *In Proceedings of the 2020 SIAM international conference on data mining*. Society for Industrial and Applied Mathematics, pp. 559–567.

De Muth, J. E. (2019) ‘Dealing with Inherent Statistical Error.’ *AAPS Advances in the Pharmaceutical Sciences Series*. Springer, Cham, 40 pp. 55–71.

Mythily, M., David, B. and Vijay, J. A. (2024) ‘Digital Twin Application in Various Sectors.’ *In Transforming Industry using Digital Twin Technology*. Cham: Springer Nature Switzerland, pp. 219–237.

de N Santos, F., D’Antuono, P., Robbelein, K., Noppe, N., Weijtjens, W. and Devriendt, C. (2023) ‘Long-term fatigue estimation on offshore wind turbines interface loads through loss function physics-guided learning of neural networks.’ *Renewable Energy*. Pergamon, 205, March, pp. 461–474.

Nader, Y., Sixt, L. and Landgraf, T. (2022) ‘DNNR: Differential Nearest Neighbors Regression.’ *In International Conference on Machine Learning*. PMLR, pp. 16296–16317.

Niaz, A., Khan, S., Niaz, F., Shoukat, M. U., Niaz, I. and Yanbing, J. (2022) ‘Smart City IoT Application for Road Infrastructure Safety and Monitoring by Using Digital Twin.’ *In 2022 International Conference on IT and Industrial Technologies (ICIT)*. IEEE, pp. 1–6.

Nodrat, F. and Kang, D. (2017) ‘GIS-Based Decision-Making Model for Road Maintenance with Vb. Net for Kabul City Roads.’ *Journal of Advances in Information Technology*, 8(3) pp. 199–203.

Noshadravan, A., Wildnauer, M., Gregory, J. and Kirchain, R. (2013) ‘Comparative pavement life cycle assessment with parameter uncertainty.’ *Transportation Research Part D: Transport and Environment*, 25 pp. 131–138.

Nunn, M. and Ferne, B. W. (2001) ‘Design and assessment of long-life flexible pavements.’ *Transportation Research Circular*, 503(12) pp. 32–49.

OECD (1987) *OECD Economic Outlook, Volume 1987 Issue 1*. OECD (OECD Economic Outlook).

O’Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H. and Invernizzi, L. (2019) ‘KerasTuner.’

Oreto, C., Biancardo, S. A., Abbondati, F. and Veropalumbo, R. (2023) ‘Leveraging Infrastructure BIM for Life-Cycle-Based Sustainable Road Pavement Management.’ *Materials*. MDPI, 16(3) p. 1047.

Pan, Y., Wang, M., Lu, L., Wei, R., Cavazzi, S., Peck, M. and Brilakis, I. (2024) ‘Scan-to-graph: Automatic generation and representation of highway geometric digital twins from point cloud data.’ *Automation in Construction*. Elsevier, 166, October, p. 105654.

Parida, M., Aggarwal, S. and Jain, S. S. (2005) ‘Enhancing pavement management systems using GIS.’ *Proceedings of the Institution of Civil Engineers: Transport*. ICE Publishing, 158(2) pp. 107–113.

Parmar, R., Leiponen, A. and Thomas, L. D. W. (2020) ‘Building an organizational digital twin.’ *Business Horizons*. Elsevier, 63(6) pp. 725–736.

Pasupunuri, S. K., Thom, N. and Li, L. (2024) ‘Roughness Prediction of Jointed Plain Concrete Pavement Using Physics Informed Neural Networks.’ *Transportation Research Record*, 2678(11) pp. 1733–1746.

Paterson, W. and Attoh-Okine, B. (1992) ‘Summary Models of Paved Road Deterioration Based on HDM-III.’ *Transportation Research Record*, (1344).

Paterson, W. D. O. (1987) *Road deterioration and maintenance effects: Models for planning and management*.

Patiño-Rodriguez, C. E. and Carazas, F. J. G. (2019) ‘Maintenance and asset life cycle for reliability systems.’ *In Reliability and Maintenance-An Overview of Cases*. IntechOpen.

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Varoquaux, G., Gramfort, A., Thirion, B., Dubourg, V., Passos, A., Brucher, M., Perrot, M. and Duchesnay, É. (2011) ‘Scikit-learn: Machine learning in Python.’ *the Journal of machine Learning research*, 12 pp. 2825–2830.

Peraka, N. S. P. and Biligiri, K. P. (2020) ‘Pavement asset management systems and technologies: A review.’ *Automation in Construction*. Elsevier, 119, November, p. 103336.

Pérez-Acebo, H., Linares-Unamunzaga, A., Abejón, R. and Rojí, E. (2018) ‘Research Trends in Pavement Management during the First Years of the 21st Century: A Bibliometric Analysis during the 2000–2013 Period.’ *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 8(7) p. 1041.

Petersen, Ø. W., Øiseth, O., & Lourens, E. (2022) ‘Wind load estimation and virtual sensing in long-span suspension bridges using physics-informed Gaussian process latent force models.’ *Mechanical Systems and Signal Processing*. Elsevier, 170 p. 108742.

Philip, B. and AlJassmi, H. (2024) ‘A Bayesian decision support system for optimizing pavement management programs.’ *Heliyon*. Elsevier, 10(3) p. e25625.

Pires, F., Cachada, A., Barbosa, J., Moreira, A. P. and Leitao, P. (2019) ‘Digital twin in industry 4.0: Technologies, applications and challenges.’ *In 2019 IEEE 17th international conference on industrial informatics (INDIN)*. IEEE, pp. 721–726.

Piryonesi, S. M. (2019) ‘The Application of Data Analytics to Asset Management: Deterioration and Climate Change Adaptation in Ontario Roads.’ *Doctoral dissertation, University of Toronto (Canada)*.

Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L. and Nee, A. Y. C. (2021) ‘Enabling technologies and tools for digital twin.’ *Journal of Manufacturing Systems*. Elsevier, 58, January, pp. 3–21.

Qiuchen Lu, V., Parlikad, A. K., Woodall, P., Ranasinghe, G. D. and Heaton, J. (2019) ‘Developing a Dynamic Digital Twin at a Building Level: using Cambridge Campus as Case Study.’ *In International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*. ICE Publishing, pp. 67–75.

Raschka, S. (2018) ‘MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack.’ *Journal of open source software*. The Open Journal, 3(24) p. 638.

Raymond, S. J. and Camarillo, D. B. (2021) ‘Applying physics-based loss functions to neural networks for improved generalizability in mechanics problems.’ *arXiv preprint arXiv*, 2105.00075.

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., Karpatne, A., Hansen, G. J. A., Hanson, P. C., Watkins, W., Steinbach, M. and Kumar, V. (2019) ‘Process-Guided Deep Learning Predictions of Lake Water Temperature.’ *Water Resources Research*. John Wiley & Sons, Ltd, 55(11) pp. 9173–9190.

Redelinghuys, A., Basson, A. and Kruger, K. (2019) ‘A six-layer digital twin architecture for a manufacturing cell.’ *In International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*. Cham: Springer International Publishing, pp. 412–423.

Refsgaard, J. C., van der Sluijs, J. P., Brown, J. and van der Keur, P. (2006) ‘A framework for dealing with uncertainty due to model structure error.’ *Advances in Water Resources*. Elsevier, 29(11) pp. 1586–1597.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N. and Prabhat (2019) ‘Deep learning and process understanding for data-driven Earth system science.’ *Nature*. Nature Publishing Group, 566(7743) pp. 195–204.

Ren, Z., Wan, J. and Deng, P. (2022) ‘Machine-Learning-Driven Digital Twin for Lifecycle Management of Complex Equipment.’ *IEEE Transactions on Emerging Topics in Computing*. IEEE Computer Society, 10(1) pp. 9–22.

Ricciardi, W. (2019) ‘Assessing the impact of digital transformation of health services: Opinion by the Expert Panel on Effective Ways of Investing in Health (EXPH).’ *European Journal of Public Health, 29(Supplement_4)* pp. ckz185-769.

Ritto, T. G. and Rochinha, F. A. (2021) ‘Digital twin, physics-based model, and machine learning applied to damage detection in structures.’ *Mechanical Systems and Signal Processing*. Academic Press, 155, June, p. 107614.

Rizvi, S. H. M. and Abbas, M. (2023) ‘From data to insight, enhancing structural health monitoring using physics-informed machine learning and advanced data collection methods.’ *Engineering Research Express*. IOP Publishing, 5(3) p. 032003.

Roberts, R., Inzerillo, L. and Di Mino, G. (2021) ‘Exploiting Data Analytics and Deep Learning Systems to Support Pavement Maintenance Decisions.’ *Applied Sciences*. MDPI AG, 11(6) p. 2458.

Robinson, H., Pawar, S., Rasheed, A. and San, O. (2022) ‘Physics guided neural networks for modelling of non-linear dynamics.’ *Neural Networks*. Pergamon, 154, October, pp. 333–345.

Robinson, Richard., Snaith, M. S. and Danielson, Uno. (1998) *Road maintenance management: concepts and systems*. Basingstoke, UK: Macmillan.

Van Rossum, G. and Drake, F. L. (1995) *Python 3 Reference Manual*. Amsterdam: Centrum voor Wiskunde en Informatica.

Sacks, R., Brilakis, I., Pikas, E., Xie, H. S. and Girolami, M. (2020) ‘Construction with digital twin information systems.’ *Data-Centric Engineering*. Cambridge University Press, 1(6) p. e14.

Saghafi, B., Hassani, A., Noori, R. and Bustos, M. G. (2009) ‘Artificial neural networks and regression analysis for predicting faulting in jointed concrete pavements considering base condition.’ *International Journal of Pavement Research and Technology*, 2(1) pp. 20–25.

Sahal, R., Alsamhi, S. H., Brown, K. N., O’shea, D., McCarthy, C. and Guizani, M. (2021) ‘Blockchain-Empowered Digital Twins Collaboration: Smart Transportation Use Case.’ *Machines*. Multidisciplinary Digital Publishing Institute, 9(9) p. 193.

Saikiran, K., Lithesh, G., Srinivas, B. and Ashok, S. (2021) ‘Prediction of Air Quality Index Using Supervised Machine Learning Algorithms.’ *In 2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*. IEEE, pp. 1–4.

Salehi, B., Belgiovine, M., Sanchez, S. G., Dy, J., Ioannidis, S. and Chowdhury, K. (2020) ‘Machine Learning on Camera Images for Fast mmWave Beamforming.’ *In 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, pp. 338–346.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E. and Valaee, S. (2017) ‘Recent Advances in Recurrent Neural Networks.’ *arXiv preprint arXiv:1801.01078*, December.

Salour, F. and Erlingsson, S. (2013) ‘Investigation of a pavement structural behaviour during spring thaw using falling weight deflectometer.’ *Road Materials and Pavement Design*. Taylor & Francis, 14(1) pp. 141–158.

San, O. and Maulik, R. (2017) ‘Neural network closures for nonlinear model order reduction.’ *Advances in Computational Mathematics*. Springer New York LLC, 44(6) pp. 1717–1750.

San, O. and Maulik, R. (2018) ‘Machine learning closures for model order reduction of thermal fluids.’ *Applied Mathematical Modelling*. Elsevier, 60, August, pp. 681–710.

Sanabria, N., Valentin, V., Bogus, S., Zhang, G. and Kalhor, E. (2017) ‘Comparing Neural Networks and Ordered Probit Models for Forecasting Pavement Condition in New Mexico,’ (No. 17-01037).

Sati, A. S., Dabous, A. and Zeiada, W. (2020) ‘Pavement Deterioration Model Using Markov Chain and International Roughness Index.’ *In IOP Conference Series: Materials Science and Engineering*. IOP Publishing, p. 012012.

Sayers, M. W., Gillespie, T. D. and V Queiroz, C. A. (1986) ‘The international road roughness experiment: Establishing correlation and a calibration standard for measurements.’ *University of Michigan, Ann Arbor, Transportation Research Institute*.

Seites-Rundlett, W., Bashar, M. Z., Torres-Machi, C. and Corotis, R. B. (2022) ‘Combined evidence model to enhance pavement condition prediction from highly uncertain sensor data.’ *Reliability Engineering & System Safety*. Elsevier, 217, January, p. 108031.

Setola, R. and Theocharidou, M. (2016) ‘Modelling dependencies between critical infrastructures.’ *Studies in Systems, Decision and Control*. Springer International Publishing, 90 pp. 19–41.

Shahandashti, S. M., Razavi, S. N., Soibelman, L., Berges, M., Caldas, C. H., Brilakis, I., Teizer, J., Vela, P. A., Haas, C., Garrett, J., Akinci, B. and Zhu, Z. (2011) ‘Data-Fusion Approaches and Applications for Construction Engineering.’ *Journal of Construction Engineering and Management*. American Society of Civil Engineers (ASCE), 137(10) pp. 863–869.

Shanmuganathan, S. (2016) ‘Artificial neural network modelling: An introduction.’ Springer International Publishing, February, pp. 1–14.

Shi, J., Pan, Z., Jiang, L. and Zhai, X. (2023) ‘An ontology-based methodology to establish city information model of digital twin city by merging BIM, GIS and IoT.’ *Advanced Engineering Informatics*. Elsevier, 57, August, p. 102114.

Shim, C. S., Kang, H. R. and Dang, N. S. (2019) ‘Digital twin models for maintenance of cable-supported bridges.’ *In International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*. ICE Publishing, pp. 737–742.

Shohel, M. and Amin, R. (2015) ‘The Pavement Performance Modeling: Deterministic vs. Stochastic Approaches.’ *Numerical Methods for Reliability and Safety Assessment*. Springer, Cham pp. 179–196.

Shrestha, A. and Dang, J. (2020) ‘Deep Learning-Based Real-Time Auto Classification of Smartphone Measured Bridge Vibration Data.’ *Sensors*. Multidisciplinary Digital Publishing Institute, 20(9) p. 2710.

Shrestha, A., Dang, J., Nakajima, K. and Wang, X. (2020) ‘Image processing–based real-time displacement monitoring methods using smart devices.’ *Structural Control and Health Monitoring*. John Wiley & Sons, Ltd, 27(2) p. e2473.

Shtayat, A., Moridpour, S., Best, B. and Rumi, S. (2022) ‘An Overview of Pavement Degradation Prediction Models.’ *Journal of Advanced Transportation*. Hindawi Limited, 2022(1) p. 7783588.

Shu, X., Wang, Z. and Basheer, I. A. (2022) ‘Large-scale evaluation of pavement performance models utilizing automated pavement condition survey data.’ *International Journal of Transportation Science and Technology*. Elsevier, 11(4) pp. 678–689.

Silva, B. N., Khan, M. and Han, K. (2018) ‘Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities.’ *Sustainable Cities and Society*. Elsevier, 38, April, pp. 697–713.

Silva, C. A., Vilaça, R., Pereira, A. and Bessa, R. J. (2024) ‘A review on the decarbonization of high-performance computing centers.’ *Renewable and Sustainable Energy Reviews*. Pergamon, 189, January, p. 114019.

Simmonds, E. G., Adjei, K. P., Andersen, C. W., Hetle Aspheim, J. C., Battistin, C., Bulso, N., Christensen, H. M., Cretois, B., Cubero, R., Davidovich, I. A., Dickel, L., Dunn, B., Dunn-Sigouin, E., Dyrstad, K., Einum, S., Giglio, D., Gjerløw, H., Godefroidt, A., González-Gil, R., Gonzalo Cogno, S., Große, F., Halloran, P., Jensen, M. F., Kennedy, J. J., Langsæther, P. E., Laverick, J. H., Lederberger, D., Li, C., Mandeville, E. G., Mandeville, C., Moe, E., Navarro Schröder, T., Nunan, D., Sicacha-Parada, J., Simpson, M. R., Skarstein, E. S., Spensberger, C., Stevens, R., Subramanian, A. C., Svendsen, L., Theisen, O. M., Watret, C. and O’Hara, R. B. (2022) ‘Insights into the quantification and reporting of model-related uncertainty across different disciplines.’ *iScience*. Elsevier, 25(12) p. 105512.

Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N. and Devine, D. (2021) ‘Digital Twin: Origin to Future.’ *Applied System Innovation*. Multidisciplinary Digital Publishing Institute, 4(2) p. 36.

Singh, M., Srivastava, R., Fuenmayor, E., Kuts, V., Qiao, Y., Murray, N. and Devine, D. (2022) ‘Applications of Digital Twin across Industries: A Review.’ *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 12(11) p. 5727.

Singh, V. (2019) ‘Digitalization, BIM ecosystem, and the future of built environment: How widely are we exploring the different possibilities?’ *Engineering, Construction and Architectural Management*. Emerald Group Publishing Ltd.

Sisodia, D. and Sisodia, D. S. (2022) ‘Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset.’ *Engineering Science and Technology, an International Journal*. Elsevier, 28, April, p. 101011.

Sisson, W., Karve, P. and Mahadevan, S. (2022) ‘Digital Twin Approach for Component Health-Informed Rotorcraft Flight Parameter Optimization.’ *AIAA Journal*. AIAA International, 60(3) pp. 1923–1936.

Smith, M. (2009) *ABAQUS/Standard User's Manual, Version 6.9*. Dassault Systèmes Simulia Corp.

Sofia, H., Anas, E. and Faïz, O. (2020) 'Mobile mapping, machine learning and digital twin for road infrastructure monitoring and maintenance: Case study of mohammed VI bridge in Morocco.' *In 2020 IEEE International conference of Moroccan Geomatics (Morgeo)*. IEEE, pp. 1–6.

Solomatine, D., See, L. M. and Abrahart, R. J. (2009) 'Data-Driven Modelling: Concepts, Approaches and Experiences.' *Practical Hydroinformatics*. Springer, Berlin, Heidelberg, October, pp. 17–30.

Song, M., Behmanesh, I., Moaveni, B. and Papadimitriou, C. (2020) 'Accounting for Modeling Errors and Inherent Structural Variability through a Hierarchical Bayesian Model Updating Approach: An Overview.' *Sensors (Basel, Switzerland)*. MDPI AG, 20(14) p. 3874.

Song, Y., Wang, Y. D., Hu, X. and Liu, J. (2022) 'An Efficient and Explainable Ensemble Learning Model for Asphalt Pavement Condition Prediction Based on LTPP Dataset.' *IEEE Transactions on Intelligent Transportation Systems*. Institute of Electrical and Electronics Engineers Inc., 23(11) pp. 22084–22093.

Souza, V. M. A., Giusti, R. and Batista, A. J. L. (2018) 'Asfault: A low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning.' *Pervasive and Mobile Computing*. Elsevier, 51, December, pp. 121–137.

Srikonda, R., Rastogi, A. and Oestensen, H. (2020) 'Increasing Facility Uptime Using Machine Learning and Physics-Based Hybrid Analytics in a Dynamic Digital Twin.' *In Offshore Technology Conference*. OTC.

Steindl, G., Stagl, M., Kasper, L., Kastner, W. and Hofmann, R. (2020) 'Generic Digital Twin Architecture for Industrial Energy Systems.' *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 10(24) p. 8903.

Stevens, D., Salt, G., Henning, T. F. P. and Roux, D. C. (2009) 'Pavement Performance Prediction: A Comprehensive New Approach to Defining Structural Capacity (SNP).' *Proceedings*.

Steyn, W. J. V. D. M. (2020) 'Selected implications of a hyper-connected world on pavement engineering.' *International Journal of Pavement Research and Technology*. Springer, 13(6) pp. 673–678.

Steyn, W. J. V. D. M. and Broekman, A. (2022) 'Development of a digital twin of a local road network: a case study.' *Journal of Testing and Evaluation*. ASTM International, 50(6) pp. 2901–2915.

Su, W., Qian, J. and Liu, L. (2015) 'Communication-Efficient False Discovery Rate Control via Knockoff Aggregation.' *arXiv preprint arXiv:1506.05446*.

Suh, Y. C. and Cho, N. H. (2014) ‘Development of a Rutting Performance Model for Asphalt Concrete Pavement based on Test Road and Accelerated Pavement Test Data.’ *KSCE Journal of Civil Engineering*, 18 pp. 165–171.

Sun, C. and Shi, V. G. (2021) ‘PhysiNet: A Combination of Physics-based Model and Neural Network Model for Digital Twins.’ *International Journal of Intelligent Systems*. John Wiley and Sons Ltd, 37(8) pp. 5443–5456.

Sun, H., Qiu, C., Lu, L., Gao, X., Chen, J. and Yang, H. (2020) ‘Wind turbine power modelling and optimization using artificial neural network with wind field experimental data.’ *Applied Energy*. Elsevier, 280, December, p. 115880.

Sun, J., Niu, Z., Innanen, K. A., Li, J. and Trad, D. O. (2020) ‘A theory-guided deep-learning formulation and optimization of seismic waveform inversion.’ *Geophysics*. Society of Exploration Geophysicists, 85(2) pp. R87–R99.

Sun, Z., Liang, B., Liu, S. and Liu, Z. (2024) ‘Data and Knowledge-Driven Bridge Digital Twin Modeling for Smart Operation and Maintenance.’ *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 15(1) p. 231.

Sundin, S. and Braban-Ledoux, C. (2001) ‘Artificial intelligence–based decision support technologies in pavement management.’ *Computer-Aided Civil and Infrastructure Engineering*. Blackwell Publishing Inc., 16(2) pp. 143–157.

Szandała, T. (2021) ‘Review and comparison of commonly used activation functions for deep neural networks.’ *Bio-inspired neurocomputing*. Springer, Singapore pp. 203–224.

Taavitsainen, V. M. (2009) ‘Denoising and Signal-to-Noise Ratio Enhancement: Splines.’ *Comprehensive Chemometrics: Chemical and Biochemical Data Analysis*. Elsevier, 3, January, pp. 165–177.

Taha, M. A. and Hanna, A. S. (1995) ‘Evolutionary neural network model for the selection of pavement maintenance strategy.’ *Transportation Research Record* p. 1497.

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B. and Liang, J. (2016) ‘Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?’ *IEEE Transactions on Medical Imaging*. Institute of Electrical and Electronics Engineers Inc., 35(5) pp. 1299–1312.

Tamagusko, T. and Ferreira, A. (2023) ‘Machine Learning for Prediction of the International Roughness Index on Flexible Pavements: A Review, Challenges, and Future Directions.’ *Infrastructures*. Multidisciplinary Digital Publishing Institute, 8(12) p. 170.

Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P. and Gao, X. (2019) ‘A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends.’ *Automation in Construction*. Elsevier, 101, May, pp. 127–139.

Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Guo, Z., C-Y Lu, S. and C Nee, A. Y. (2019) 'Digital twin-driven product design framework.' *International Journal of Production Research*. Taylor and Francis Ltd., 57(12) pp. 3935–3953.

Tao, F., Zhang, M. and Nee, A. Y. C. (2019) 'Five-Dimension Digital Twin Modeling and Its Key Technologies.' *Digital Twin Driven Smart Manufacturing*. Elsevier pp. 63–81.

Tavazza, F., Decost, B. and Choudhary, K. (2021) 'Uncertainty prediction for machine learning models of material properties.' *ACS Omega*. American Chemical Society, 6(48) pp. 32431–32440.

Tchana, Y., Ducellier, G. and Remy, S. (2019) 'Designing a unique Digital Twin for linear infrastructures lifecycle management.' *Procedia CIRP*. Elsevier B.V., 84, January, pp. 545–549.

Teng, S. Y., Touš, M., Leong, W. D., How, B. S., Lam, H. L. and Máša, V. (2021) 'Recent advances on industrial data-driven energy savings: Digital twins and infrastructures.' *Renewable and Sustainable Energy Reviews*. Pergamon, 135, January, p. 110208.

Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B. D., Todd, M. D., Mahadevan, S., Hu, C. and Hu, Z. (2022) 'A comprehensive review of digital twin — part 1: modeling and twinning enabling technologies.' *Structural and Multidisciplinary Optimization*. Springer Science and Business Media Deutschland GmbH, 65(12) pp. 1–55.

Thenmozhi, T. and Helen, R. (2022) 'Feature Selection Using Extreme Gradient Boosting Bayesian Optimization to upgrade the Classification Performance of Motor Imagery signals for BCI.' *Journal of Neuroscience Methods*. Elsevier, 366, January, p. 109425.

Thom, N. (2024) *Principles of pavement engineering*. Emerald Publishing Limited.

Tibshirani, R. (1996) 'A comparison of some error estimates for neural network models.' *Neural computation*, 8(1) pp. 152–163.

Titscher, T., van Dijk, T., Kadoke, D., Robens-Radermacher, A., Herrmann, R. and Unger, J. F. (2023) 'Bayesian model calibration and damage detection for a digital twin of a bridge demonstrator.' *Engineering Reports*. John Wiley and Sons Inc, 5(11).

Titus-Glover, L. and Darter, M. I. (2001) 'Appendix PP: smoothness prediction for rigid pavements, guide for mechanistic-empirical design of new and rehabilitated pavement structures.' *Transportation Research Board, Washington, DC.*

Tran, H., Robert, D. and Setunge, S. (2024) 'Extending Service Life of Stormwater Drainage Pipes with Proactive Maintenance Tools.' *Journal of Pipeline Systems Engineering and Practice*. American Society of Civil Engineers, 15(4) p. 04024049.

Trauer, J., Schweigert-Recksiek, S., Engel, C., Spreitzer, K. and Zimmermann, M. (2020) 'What is a digital twin?–definitions and insights from an industrial case study in technical product development.' *In Proceedings of the design society: DESIGN conference*. Cambridge University Press, pp. 757–766.

TRL (1993) *A Guide to the Structural Design of Bitumen-Surfaced Roads in Tropical and Sub-tropical Countries*. Crowthorne, Berkshire, United Kingdom.

Trousdale, D. (2019) 'The Sensorisation of Infrastructure Management and Maintenance: Enabling Better Service Delivery.' *In 26th World Road Congress World Road (PIARC)*.

Tu, Z., Qiao, L., Nowak, R., Lv, H. and Lv, Z. (2022) 'Digital Twins-Based Automated Pilot for Energy-Efficiency Assessment of Intelligent Transportation Infrastructure.' *IEEE Transactions on Intelligent Transportation Systems*. Institute of Electrical and Electronics Engineers Inc., 23(11) pp. 22320–22330.

Tuhaise, V. V., Tah, J. H. M. and Abanda, F. H. (2023) 'Technologies for digital twin applications in construction.' *Automation in Construction*. Elsevier, 152, August, p. 104931.

Tuncali, C. E., Fainekos, G., Ito, H. and Kapinski, J. (2018) 'Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components.' *In 2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1555–1562.

Vabalas, A., Gowen, E., Poliakoff, E. and Casson, A. J. (2019) 'Machine learning algorithm validation with a limited sample size.' *PloS one*. Public Library of Science, 14(11) p. e0224365.

Vagdatli, T. and Petroutsatou, K. (2022) 'Modelling Approaches of Life Cycle Cost–Benefit Analysis of Road Infrastructure: A Critical Review and Future Directions.' *Buildings*. Multidisciplinary Digital Publishing Institute, 13(1) p. 94.

VanDerHorn, E. and Mahadevan, S. (2021) 'Digital Twin: Generalization, characterization and implementation.' *Decision Support Systems*. North-Holland, 145, June, p. 113524.

Varley, A., Tyler, A., Smith, L., Dale, P. and Davies, M. (2016) 'Mapping the spatial distribution and activity of 226Ra at legacy sites through Machine Learning interpretation of gamma-ray spectrometry data.' *Science of The Total Environment*. Elsevier, 545–546, March, pp. 654–661.

Vengadeswaran, Binu, D. and Rai, L. (2024) 'An Efficient Framework for Crime Prediction Using Feature Engineering and Machine Learning.' *Lecture Notes in Networks and Systems*. Springer, Singapore, 796 pp. 49–59.

Viale, L. and Zouari, D. (2020) 'Impact of digitalization on procurement: the case of robotic process automation.' *Supply Chain Forum: An International Journal*. Taylor & Francis, 21(3) pp. 185–195.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, Ilhan, Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P. and SciPy 1.0 Contributors (2020) 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.' *Nature Methods*, 17 pp. 261–272.

Visintine, B., Rada, G., Simpson, A. and Wheeler, A. (2018) *Guidelines for Informing Decision making to Affect Pavement Performance Measures*. (No. FHWA-HRT-17-090). United States. Federal Highway Administration. Office of Infrastructure Research and Development.

Wan, Z., Xu, Y. and Šavija, B. (2021) 'On the Use of Machine Learning Models for Prediction of Compressive Strength of Concrete: Influence of Dimensionality Reduction on the Model Performance.' *Materials*. Multidisciplinary Digital Publishing Institute, 14(4) p. 713.

Wan, Z. Y., Vlachas, P. R., Koumoutsakos, P. and Sapsis, T. P. (2018) 'Data-assisted reduced-order modeling of extreme events in complex dynamical systems.' *PloS one*. Public Library of Science, 13(5) p. e0197704.

Wang, D., Guo, J., Ouyang, Y., Wang, S., Yang, A., Ren, Z., Ding, Y., Chen, G., Zhou, C. and Chen, D. (2022) 'Leverage Digital Twins Technology for Network Lifecycle Management.' *In 2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*. IEEE, pp. 1–5.

Wang, G. (2019) 'An investigation of the suitability of smartphone devices for road condition assessment.' *Doctoral dissertation, University of Birmingham*, December.

Wang, H., Ou, S., Dahlhaug, O. G., Storli, P. T., Skjelbred, H. I. and Vilberg, I. (2023) 'Adaptively Learned Modeling for a Digital Twin of Hydropower Turbines with Application to a Pilot Testing System.' *Mathematics*. Multidisciplinary Digital Publishing Institute, 11(18) p. 4012.

Wang, H., Xu, Z. and Yue, L. (2020) 'Comparing of Data Collection for Network Level Pavement Management of Urban Roads and Highways.' *Journal of Advanced Transportation*. John Wiley & Sons, Ltd, 2020(1) p. 9237963.

Wang, K. C. P., Zaniewski, J. and Way, G. (1994) 'Probabilistic behavior of pavements.' *Journal of Transportation Engineering*, 120(3) pp. 358–375.

Wang, K., Zhang, L., Jia, Z., Cheng, H., Lu, H. and Cui, J. (2024) 'A framework and method for equipment digital twin dynamic evolution based on IExATCN.' *Journal of Intelligent Manufacturing*. Springer, 35(4) pp. 1571–1583.

Wang, R. and Yu, R. (2021) 'Physics-Guided Deep Learning for Dynamical Systems.' *arXiv preprint arXiv:2107.01272*.

Wang, R., Yu, R. and Walters, R. (2020) 'Incorporating Symmetry into Deep Dynamics Models for Improved Generalization.' *arXiv preprint arXiv:2002.03061*. International Conference on Learning Representations, ICLR, February.

Wang, W., Qin, Y., Li, X., Wang, D. and Chen, H. (2017) 'Comparisons of Faulting-Based Pavement Performance Prediction Models.' *Advances in Materials Science and Engineering*. Hindawi Limited, 2017(1) p. 6845215.

Wang, W., Zaheer, Q., Qiu, S., Wang, Weidong, Ai, C., Wang, J., Wang, S. and Hu, W. (2024) 'Digital Twins Technologies.' *Digital Twin Technologies in Transportation Infrastructure Management*. Springer, Singapore pp. 27–74.

Wang, Y., Wang, J., Wang, F., Wang, L. and Wei, W. (2016) 'An Optimized Data Obtaining Strategy for Large-Scale Sensor Monitoring Networks.' *International Journal of Distributed Sensor Networks*. SAGE PublicationsSage UK: London, England, 12(6) p. 4262565.

Waqar, A. (2024) 'Intelligent decision support systems in construction engineering: An artificial intelligence and machine learning approaches.' *Expert Systems with Applications*. Pergamon, 249, September, p. 123503.

Haider, S. W., Baladi, G. Y., Chatti, K., & Dean, C. M. (2010) 'Effect of Frequency of Pavement Condition Data Collection on Performance Prediction.' *Transportation Research Record*, 2153(1) pp. 67–80.

Wasiq, S. and Golroo, A. (2024) 'Probabilistic pavement performance modeling using hybrid Markov Chain: A case study in Afghanistan.' *Case Studies in Construction Materials*. Elsevier, 20, July, p. e03023.

Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., Augspurger, T., Halchenko, Y., Cole, J. B., Warmenhoven, J., de Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M., Meyer, K., Miles, A., Ram, Y., Yarkoni, T., Williams, M. L., Evans, C., Fitzgerald, C., Brian, Fonnesbeck, C., Lee, A. and Qalieh, A. (2017) 'mwaskom/seaborn: v0.8.1 (September 2017).' Zenodo, September.

Willard, J., Jia, X., Steinbach, M., Kumar, V. and Xu, S. (2020) 'Integrating Physics-Based Modeling With Machine Learning: A Survey.' *arXiv preprint arXiv:2003.04919*, 1(1) pp. 1–34.

Willard, J., Jia, X., Xu, S., Steinbach, M. and Kumar, V. (2022) 'Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems.' *ACM Computing Surveys*. ACM PUB27 New York, NY , 55(4).

Wiman, L. G. (2008) 'Preliminary Relationship Between Deflection and Rut Depth Propagation for Flexible Pavement Using ALT.' *In APT'08. Third International ConferenceCentro de Estudios y Experimentación de Obras Públicas (CEDEX) Transportation Research Board*.

Won, J., Park, J., Park, J. W. and Kim, I. (2020) 'BLESeis: Low-Cost IoT Sensor for Smart Earthquake Detection and Notification.' *Sensors*. Multidisciplinary Digital Publishing Institute, 20(10) p. 2963.

Kouraytem, N., Li, X., Tan, W., Kappes, B., & Spear, A. D. (2021) 'Modeling process–structure–property relationships in metal additive manufacturing: a review on physics-driven versus data-driven approaches.' *Journal of Physics: Materials*. IOP Publishing, 4(3) p. 032002.

Wu, C., Zhu, X. and Si, W. (2023) ‘Sensitivity analysis of asphalt pavement performance under freeze-thaw cycles by applying reliability method.’ *Case Studies in Construction Materials*. Elsevier, 19, December, p. e02656.

Wu, Y., Sicard, B. and Gadsden, S. A. (2024) ‘Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring.’ *Expert Systems with Applications*. Pergamon, December, p. 124678.

Van Wyk, F., Khojandi, A., Kamaleswaran, R., Akbilgic, O., Nemati, S. and Davis, R. L. (2017) ‘How much data should we collect? A case study in sepsis detection using deep learning.’ *In 2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*. IEEE, pp. 109–112.

Xu, Y., He, J., Ji, J. and Xu, S. (2018) ‘Effect of Pavement Condition Monitoring Frequency with Unequal Interval on Determining Pavement Service Life.’ *Advances in Civil Engineering Materials*. American Society of Mechanical Engineers Digital Collection, 7(2) pp. 261–271.

Xu, Y. and Zhang, Z. (2022) ‘Review of Applications of Artificial Intelligence Algorithms in Pavement Management.’ *Journal of Transportation Engineering, Part B: Pavements*. American Society of Civil Engineers (ASCE), 148(3) p. 03122001.

Xu, Z., Kang, R. and Lu, R. (2020) ‘3D Reconstruction and Measurement of Surface Defects in Prefabricated Elements Using Point Clouds.’ *Journal of Computing in Civil Engineering*. American Society of Civil Engineers (ASCE), 34(5) p. 04020033.

Yang, J., Lu, J. J., Gunaratne, M. and Xiang, Q. (2003) ‘Forecasting Overall Pavement Condition with Neural Networks: Application on Florida Highway Network.’ *Transportation Research Record*. National Research Council, 1853(1) pp. 3–12.

Yang, L. and Özel, T. (2021) ‘Physics-based simulation models for digital twin development in laser powder bed fusion.’ *International Journal of Mechatronics and Manufacturing Systems*. Inderscience Publishers, 14(2) pp. 143–163.

Yao, L., Dong, Q., Jiang, J. and Ni, F. (2020) ‘Deep reinforcement learning for long-term pavement maintenance planning.’ *Computer-Aided Civil and Infrastructure Engineering*. Blackwell Publishing Inc., 35(11) pp. 1230–1245.

Yao, L., Leng, Z., Jiang, J. and Ni, F. (2021) ‘Modelling of pavement performance evolution considering uncertainty and interpretability: a machine learning based framework.’ *International Journal of Pavement Engineering*. Taylor & Francis, 23(14) pp. 5211–5226.

Ye, C., Butler, L., Bartek, C., Iangurazov, M., Lu, Q., Gregory, A., Girolami, M. and Middleton, C. (2019) ‘A Digital Twin of Bridges for Structural Health Monitoring.’ *In 12th International Workshop on Structural Health Monitoring*. Stanford University.

Ye, Z., Wei, Y., Li, J., Yan, G. and Wang, L. (2022) ‘A distributed pavement monitoring system based on Internet of Things.’ *Journal of Traffic and Transportation Engineering (English Edition)*. Elsevier, 9(2) pp. 305–317.

Ye, Z., Wei, Y., Yang, S., Li, P., Yang, F., Yang, B. and Wang, L. (2024) 'IoT-enhanced smart road infrastructure systems for comprehensive real-time monitoring.' *Internet of Things and Cyber-Physical Systems*. Elsevier, 4, January, pp. 235–249.

Yin, M. and Kumar Reja, V. (2024) 'How Can Digital Twins Be Used in Highway Maintenance? A Questionnaire Survey for Industry Practitioners.'

Yoon, S., Lee, S., Kye, S., Kim, I. H., Jung, H. J. and Spencer, B. F. (2022) 'Seismic fragility analysis of deteriorated bridge structures employing a UAV inspection-based updated digital twin.' *Structural and Multidisciplinary Optimization*. Springer Science and Business Media Deutschland GmbH, 65(12) pp. 1–17.

Yu, G., Wang, Y., Mao, Z., Hu, M., Sugumaran, V. and Wang, Y. K. (2021) 'A digital twin-based decision analysis framework for operation and maintenance of tunnels.' *Tunnelling and Underground Space Technology*. Pergamon, 116, October, p. 104125.

Yu, G., Zhang, S., Hu, M. and Ken Wang, Y. (2020) 'Prediction of highway tunnel pavement performance based on digital twin and multiple time series stacking.' *Advances in Civil Engineering*. Hindawi Limited, 2020(1).

Yu, T., Pei, L. I., Li, W., Sun, Z. Y. and Huyan, J. (2021) 'Pavement Surface Condition Index Prediction Based on Random Forest Algorithm.' *Journal of Highway and Transportation Research and Development (English Edition)*. Research Institute of Highway, Ministry of Transport, Beijing (RIOH), 15(4) pp. 1–11.

Zabielski, J. and Srokosz, P. (2020) 'Monitoring of Structural Safety of Buildings Using Wireless Network of MEMS Sensors.' *Buildings*. Multidisciplinary Digital Publishing Institute, 10(11) p. 193.

Zakir, D., Greenwood, I. and Ayoobi, A. (2024) 'Sustainable transport infrastructure: Road asset management in the CAREC region.' Tokyo: Asian Development Bank Institute (ADBI), March.

Zeiada, W., Hamad, K., Omar, M., Underwood, B. S., Khalil, M. A. and Karzad, A. S. (2019) 'Investigation and modelling of asphalt pavement performance in cold regions.' *International Journal of Pavement Engineering*. Taylor & Francis, 20(8) pp. 986–997.

Zemouri, R., Ibrahim, R. and Tahan, A. (2023) 'Hydrogenerator early fault detection: Sparse Dictionary Learning jointly with the Variational Autoencoder.' *Engineering Applications of Artificial Intelligence*. Pergamon, 120, April, p. 105859.

Zhang, A., Yang, J. and Wang, F. (2023) 'Application and enabling digital twin technologies in the operation and maintenance stage of the AEC industry: A literature review.' *Journal of Building Engineering*. Elsevier, December, p. 107859.

Zhang, H., Liu, Q., Chen, X. and Zhang, D. (2017) 'A digital twin-based approach for designing and multi-objective optimization of hollow glass production line.' *Ieee Access*, 5 pp. 26901–26911.

Zhang, H., Qi, Q., Ji, W. and Tao, F. (2023) ‘An update method for digital twin multi-dimension models.’ *Robotics and Computer-Integrated Manufacturing*. Pergamon, 80, April, p. 102481.

Zhang, M., Tao, F., Huang, B., Liu, A., Wang, L., Anwer, N. and Nee, A. Y. C. (2022) ‘Digital twin data: methods and key technologies.’ *Digital Twin*. F1000 Research Ltd, 1, February, p. 2.

Zhang, Teng, Peng, F., Yan, R., Tang, X., Deng, R. and Yuan, J. (2024) ‘Quantification of uncertainty in robot pose errors and calibration of reliable compensation values.’ *Robotics and Computer-Integrated Manufacturing*. Pergamon, 89, October, p. 102765.

Zhang, Tianren, Wang, W., Dong, R., Wang, Y., Peng, T., Zheng, P. and Yang, Z. (2024) ‘Physics-guided neural network for grinding temperature prediction.’ *Journal of Engineering Design*. Taylor & Francis, May, pp. 1–24.

Zhang, Y. and Cheng, L. (2023) ‘The role of transport infrastructure in economic growth: Empirical evidence in the UK.’ *Transport Policy*. Pergamon, 133, March, pp. 223–233.

Zhang, Z., Huang, Y., Bridgelall, R., Palek, L. and Strommen, R. (2015) ‘Sampling optimization for high-speed weigh-in-motion measurements using in-pavement strain-based sensors.’ *Measurement Science and Technology*, 26(6) p. 065003.

Zheng, M. and Guo, Y. (2024) ‘Exploring latent profiles in the relationship between learning atmosphere, negative emotions, and willingness to communicate among EFL learners.’ *Innovation in Language Learning and Teaching*. Routledge, September, pp. 1–20.

Zhou, T., Li, Y., Wu, Y. and Carlson, D. (2021) ‘Estimating Uncertainty Intervals from Collaborating Networks.’ *Journal of Machine Learning Research*, 22 pp. 1–47.

Zhu, Z., Liu, C. and Xu, X. (2019) ‘Visualisation of the Digital Twin data in manufacturing by using Augmented Reality.’ *Procedia CIRP*. Elsevier, 81, January, pp. 898–903.

Ziari, H., Maghrebi, M., Ayoubinejad, J. and Waller, S. T. (2016) ‘Prediction of pavement performance: Application of support vector regression with different kernels.’ *Transportation Research Record*. National Research Council, 2589(1) pp. 135–145.

Ziari, H., Sobhani, J., Ayoubinejad, J. and Hartmann, T. (2015) ‘Prediction of IRI in short and long terms for flexible pavements: ANN and GMDH methods.’ *International Journal of Pavement Engineering*. Taylor and Francis Ltd., 17(9) pp. 776–788.

Zimmerman, K. A. and Peshkin, D. G. (2003) ‘Applying Pavement Preservation Concepts to Low-Volume Roads.’ *Transportation research record*. SAGE PublicationsSage CA: Los Angeles, CA, 1819(1) pp. 81–87.

Appendix A: Data Pre-processing and Cleaning Python Script

The script was written in Python Jupyter Notebook to perform data pre-processing and cleaning on all the data collected, following similar procedures. Herein, an exemplary code snippet for data pre-processing on Rutting data is presented.

```
def get_rutting_values(filenames, dataset):
    dfs_rutting = []
    for file in filenames:
        dfs_rutting.append(pd.read_excel(file, sheet_name='Distress ACP & CRCP', skiprows=2, header=None))
    ### Read rutting data for each road section
    def create_dataframe_for_each_section(df):
        ### Rename the dataframe columns
        df.columns=["Year", "CN Event Description", "Fatigue", "Longitudinal Cracking", "Transverse Cracking", "Rutting", "Punchouts", "Longitudinal Cracking", "Transverse Cracking", "Spalling of Long. Joints"]
        df = df.drop(columns=["Fatigue", "Longitudinal Cracking", "Transverse Cracking","Punchouts", "Longitudinal Cracking", "Transverse Cracking", "Spalling of Long. Joints"])
        ### Outlier detection and removal function
        def outlierDetectionAndRemoval(df):
            ### Get the outliers
            rutting = df['Rutting']
            rutting_dropna = df['Rutting'].dropna().tolist()
            mean = np.mean(rutting_dropna)
            std = np.std(rutting_dropna)
            threshold = 2 ### set up a threshold
            outlier = []
            for i in rutting_dropna:
                if (i == 0):
                    continue
                if std != 0:
                    z = (i-mean)/std
                    if z > threshold:
                        outlier.append(i)
            print('Outlier in the dataset is:', outlier)
            ### Set outliers to NaN
            for i in rutting:
                if i in outlier:
                    index = rutting[rutting == i].index[0]
                    rutting = rutting.drop(index)
            ### df.drop(columns=['Rutting'])
            df['Rutting']=rutting
            return df

        def objective(x,a,b):
            return a * x + b
```

```
### 1. Check if the dataframe needs to be separated into multiple sub-dataframes
        maintenance_type_1 = "Overlay"
        maintenance_type_2 = "Surface Treatment"
        for i in range(0,len(df)):
            if (type(df.iloc[i,1]) != float and ((maintenance_type_1 in df.iloc[i,1]) or
(maintenance_type_2 in df.iloc[i,1]))):
                df.iloc[i,2] = 0
### 2. Drop all the NaNs in the DataFrame
        df = df[df['Rutting'].notna()]
        df = df.reset_index(inplace = False, drop = True)
### 3. Address values for duplicate years
        years_temp = []
        years_temp = pd.to_datetime(df['Year']).dt.year.tolist()
        df = df.assign(Year=years_temp)

    ### 3.1. If it is the year with 0, keep the year with 0. Other years, getting an average.
        year_with_maintenance = list(df[df['Year'].map(df.groupby('Year').apply(lambda x:
x['Rutting'].eq(0).any()))]['Year'])
        year_with_maintenance = list(dict.fromkeys(year_with_maintenance))
        for year in year_with_maintenance:
            indexNames = df[df['Year'] == year].index
            ### Delete these row indexes from dataFrame = the ones with value 0
            df.drop(indexNames,inplace=True)
        df = df.groupby('Year').mean().reset_index()
        for year in year_with_maintenance:
            df_to_add = pd.DataFrame([[year, 0]], columns=['Year', 'Rutting'])
            df = df.append(df_to_add, ignore_index=True)
            df.sort_values(by=['Year'], inplace=True)
            df = df.reset_index(inplace = False, drop = True)
### 4. Separate dataframe into multiple sub-dataframes if needed
        index_for_separation = df[df['Rutting'] == 0].index.tolist()
        dataframes = []
        if (len(index_for_separation) == 0):
            dataframes.append(df)
        if (len(index_for_separation) != 0):
            length = len(index_for_separation)
            if (length == 1):
                dataframes.append(df.iloc[0:index_for_separation[0],:])
                dataframes.append(df.iloc[index_for_separation[0]:,:])
            else:
                dataframes.append(df.iloc[0:index_for_separation[0],:])
                for i in range(0, length-1):
                    dataframes.append(df.iloc[index_for_separation[i]:index_for_separation[i+1],:])
                dataframes.append(df.iloc[index_for_separation[length-1]:,:])
        for element in dataframes:
            if element.empty:
                dataframes.remove(element)
        df_full = []
        combined_dataframe = []

        ### Operations within each sub-dataframes:
        for subset in dataframes:
```

```
            ### 4.1. Detect outlier and remove outliers
            outlierDetectionAndRemoval(subset)
            ### 4.2. Remove NaNs
            subset_dropna = subset[subset['Rutting'].notna()]
            subset = subset_dropna.reset_index(inplace = False, drop = True)
            ### 4.3 Address missing years
            year_s = pd.Series(np.arange(subset['Year'].min(), subset['Year'].max()+1))
            years_df = pd.DataFrame(year_s, columns =['Year'])
            dict1 = subset.set_index('Year').to_dict()
            rutting_dict = dict1['Rutting']
            rutting_full = []
            full_years = years_df['Year'].tolist()
            for item in full_years:
               if item in rutting_dict:
                  rutting_full.append(rutting_dict[item])
               else:
                  rutting_full.append(np.nan)
            years_df['Rutting'] = rutting_full
            modified_rutting = []
            ### Interpolation happens based on moving average for each subset
            for j in range(0, len(years_df)):
               if(years_df['Rutting'].iloc[j] == 0):
                  modified_rutting.append(0)
               elif (j == 0): ### First record
                  modified_rutting.append(years_df['Rutting'].iloc[j:j+2].mean())
               elif (j == len(years_df)-1): ### Last record
                  modified_rutting.append(years_df['Rutting'].iloc[j-1:j+1].mean())
               else: ### Records in between
                  modified_rutting.append(years_df['Rutting'].iloc[j-1:j+2].mean()) # it was 2 before
            years_df['Rutting_averaged'] = modified_rutting
            years_df['Rutting_averaged'] = years_df['Rutting_averaged'].interpolate(method='spline',
limit_direction ='both', order = 1) # interpolate any NaN
            df_full.append(years_df)
         ### 5. Combine separate dataframes back to one
         years_df_all = df_full[0]
         for k in range(1, len(df_full)):
            years_df_all = years_df_all.append(df_full[k],ignore_index = True)
         ### 6. Fill missing years in between after combining multiple sub-DataFrames
         year_s = pd.Series(np.arange(years_df_all['Year'].min(), years_df_all['Year'].max()+1))
         years_df = pd.DataFrame(year_s, columns =['Year'])
         dict1 = years_df_all.set_index('Year').to_dict()
         rutting_dict = dict1['Rutting_averaged']
         rutting_full = []
         full_years = years_df['Year'].tolist()
         for item in full_years:
            if item in rutting_dict:
               rutting_full.append(rutting_dict[item])
            else:
               rutting_full.append(np.nan)
         years_df['Rutting_averaged'] = rutting_full
         ### 6.1. Logic to interpolate the missing values
         modified_rutting_all = []
         count = 0
```

```
        for j in range(0, len(years_df)): ### Use the best fit line equation to interpolate
            if (pd.isnull(years_df['Rutting_averaged'].iloc[j])):
                if (years_df['Year'].iloc[j] >= 1995 and years_df['Year'].iloc[j] <= 2007):
                    ### calculate the equation and interpolate the missing year
                    index = 0
                    if (len(years_df.iloc[:j-count].index[years_df.iloc[:j-count]['Rutting_averaged'] ==
0].tolist())>0):
                        index = years_df.iloc[:j-count].index[years_df.iloc[:j-count]['Rutting_averaged'] ==
0].tolist()[-1]
                    x = years_df.iloc[index:(j-count)]['Year']
                    y = years_df.iloc[index:(j-count)]['Rutting_averaged']
                    popt, _ = curve_fit(objective, x, y)
                    a, b = popt
                    newValue = objective(years_df['Year'].iloc[j],a,b)
                    modified_rutting_all.append(newValue)
                    count += 1
                else:
                    count += 1
                    previousValue = years_df['Rutting_averaged'].iloc[j-count]
                    presentValue = previousValue + 1*count
                    modified_rutting_all.append(presentValue)
            else:
                count = 0
                modified_rutting_all.append(years_df['Rutting_averaged'].iloc[j])

        ### 6.1. Logic to interpolate the missing values
        years_df['Rutting (mm)'] = modified_rutting_all

        ### 7. Final checks on missing years between 1995 and 2007
        df = years_df
        ### Remove the rows before the year of 1995
        df = df[df['Year'] >= 1995]
        ### Remove the rows after the year of 2007
        df = df[df['Year'] < 2008]
        ### Resetting index
        df = df.reset_index(inplace = False, drop = True)
        years = pd.Series(np.arange(1995, 2008))
        years_df = pd.DataFrame(years, columns =['Year'])
        dict_final = df.set_index('Year').to_dict()
        rutting_dict = dict_final['Rutting (mm)']
        rutting_final = []
        full_years = years_df['Year'].tolist()
        for item in full_years:
            if item in rutting_dict:
                rutting_final.append(rutting_dict[item])
            else:
                rutting_final.append(np.nan)
        years_df['Rutting (mm)'] = rutting_final

        ### 7.1 Address missing values from top, or at the bottom
        years_df['Rutting (mm)'] = years_df['Rutting (mm)'].interpolate(method ='spline',
limit_direction ='both', order = 1)
        for k in range(0, len(years_df)):
```

```
            if (years_df['Rutting (mm)'].iloc[k] == 0 and years_df['Rutting (mm)'].iloc[k+1] == 0):
                years_df['Rutting (mm)'].iloc[k+1] = (years_df['Rutting (mm)'].iloc[k] +
years_df['Rutting (mm)'].iloc[k+2])/2
        return years_df
    clean_df = []
    for i in range(0, len(dfs_rutting)):
        clean_df.append(create_dataframe_for_each_section(dfs_rutting[i]))
    whole_data_rutting = pd.concat(clean_df)
    dataset['Rutting (mm)'] = whole_data_rutting['Rutting (mm)'].tolist()
```

Appendix B: Exhaustive Feature Selection for Rutting and IRI

The feature selection process used the mlxtend (machine learning extensions). It is a Python library of useful tools for data science tasks.

```
### exhaustive feature selection process
from sklearn.linear_model import LinearRegression
from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS

X = dataset_for_ml_shifted[dataset_for_ml_shifted.columns[1:].tolist()]
y = dataset_for_ml_shifted[['IRI (t+1)']]

lr = LinearRegression()

efs = EFS(lr,
        min_features=5,
        max_features=22, ### adjustable depending on predicting IRI or Rutting
        scoring='neg_mean_squared_error',
        cv=10)

efs.fit(X, y.values.ravel())

print('Best MSE score: %.2f' % efs.best_score_ * (-1))
print('Best subset:', efs.best_idx_)

df = pd.DataFrame.from_dict(efs.get_metric_dict()).T
df.sort_values('avg_score', inplace=True, ascending=False)
df.head(10)
```

Appendix C: RF Model Development Including Hyperparameter Tuning

After defining the hyperparameter space, grid search method was used to identify the best set of hyperparameters which were used as values for the model attributes. Grid search is one of the most common methods for hyperparameter tuning. It exhaustively tries every combination of the provided hyper-parameter values to find the best model.

```
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import GridSearchCV
### Number of trees in random forest
n_estimators = [int(x) for x in range(50,500,50)]
### Number of features to consider at every split
### max_features = ['auto', 'sqrt' , 'log2', 1.0]
### Maximum number of levels in tree
max_depth = [int(x) for x in range(3,50,1)]
max_depth.append(None)
### Minimum number of samples required to split a node
min_samples_split = [int(x) for x in range(2,30,2)]
### Minimum number of samples required at each leaf node
min_samples_leaf = [1,2,5,10,50,100] # 3 papers justify the choices made here.
### Method of selecting samples for training each tree
###bootstrap = [True, False]
### Create the random grid
random_grid = {'n_estimators': n_estimators,
               #'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}}
pprint(random_grid)
### Use the grid search to search for best hyperparameters
### First create the base model to tune
rf = RandomForestRegressor()
### Random search of parameters, using 3-fold cross validation,
### search across 100 different combinations, and use all available cores
rf_grid = GridSearchCV(estimator = rf, param_grid = random_grid, cv = 5, verbose=2, n_jobs = -1)
### Fit the random search model
random_forest_reg_model = RandomForestRegressor(n_estimators = 100, min_samples_split = 12,
min_samples_leaf = 5, max_depth = 5)
random_forest_reg_model.fit(x,y)
cv = RepeatedKFold(n_splits=10, n_repeats=1)
n_scores = cross_validate(random_forest_reg_model, x, y, scoring='r2', cv=cv, n_jobs=-1,
error_score='raise', return_estimator=True)
```

Appendix D: ANN Model Development Including Hyperparameter Tuning

The Bayesian Optimisation method was used for identifying the best parameters for the ANN. It is a global optimization method for noisy black box functions, build a probability model of the objective function and use it to select the most promising hyperparameters to evaluate in the true objective function.

```
def build_model(hp):
   model = keras.Sequential()
   for i in range(hp.Int('num_layers', 1, 10)):
```

```
        model.add(layers.Dense(units=hp.Int('units_' + str(i),
                                            min_value=32,
                                            max_value=512,
                                            step=32),
                               activation='relu'))
    model.add(layers.Dense(1, activation='linear'))
    model.compile(
        optimizer=keras.optimizers.Adam(
            hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])),
        loss='mean_absolute_error',
        metrics=['mean_absolute_error'])
    return model

tuner = BayesianOptimization(
        build_model,
        objective='val_mean_absolute_error',
        max_trials=50)
 tuner.search(x,y,epochs=300,validation_split = 0.1,shuffle = True)
 models = tuner.get_best_models(num_models=2)
 mlp_reg_model = models[0]
 print("[INFO] training the best model...")
 mlp_reg_model.fit(x, y, epochs=300)
 print("[INFO] training the best model...COMPLETED")
 y_pred = mlp_reg_model.predict(x)
 y_pred = round(y_pred[0][0],3)
```

Appendix E: Script to Generate Prediction Intervals Based on a List of Actual and Predicted Values Produced by Models

```
from matplotlib import pyplot
### Read results saved in excel files
df_ml = pd.read_excel(filepath, sheet_name='ML', skiprows=0, header=None)
df_ml.columns = ["ML_Actual Rutting","ML_Predicted Rutting"]

### Save results into Pandas DataFrames for both actual and prediction values
df_ml_actual = df_ml["ML_Actual Rutting"]
df_ml_prediction = df_ml["ML_Predicted Rutting"]

x = df_ml_actual
y = df_ml_prediction
pyplot.scatter(x, y)
xpoints = ypoints = plt.xlim(0,16)
xpoints = ypoints = plt.ylim(0,16)
pyplot.plot(xpoints, ypoints, linestyle='--', color='b', lw=1.5)
pyplot.show()

### simple linear regression model
from numpy.random import randn
from numpy.random import seed
from scipy.stats import linregress
```

```
from matplotlib import pyplot
from numpy import sum as arraysum

### fit linear regression model
b1, b0, r_value, p_value, std_err = linregress(x, y)
print('b0=%.3f, b1=%.3f' % (b0, b1))

### make prediction
yhat = b1 * x + b0

### plot data and predictions
pyplot.scatter(x, y)
xpoints = ypoints = plt.xlim(0,16)
xpoints = ypoints = plt.ylim(0,16)
pyplot.plot(xpoints, ypoints, linestyle='--', color='b', lw=1.5)
pyplot.plot(x, yhat, color='r')
pyplot.show()

### define new input, expected value and prediction
x_in = x[0]
y_out = y[0]
yhat_out = yhat[0]

### estimate stdev of yhat
print (y-yhat)

### pyplot.scatter(y-yhat)


sum_errs = arraysum((y - yhat)**2)
stdev = sqrt(1/(len(y)-2) * sum_errs)

### calculate prediction interval
interval = 1.645 * stdev
print('Prediction Interval: %.3f' % interval)
lower, upper = yhat_out - interval, yhat_out + interval
print('90%% likelihood of the prediction value is between %.3f and %.3f' % (lower, upper))
print('The actual value: %.3f' % x_in)
print('The prediction value: %.3f' % y_out)
print('The prediction fitting line value: %.3f' % yhat_out)
### plot dataset and prediction with interval
### Prediction interval is displayed in the plot.
pyplot.scatter(x, y)
xpoints = ypoints = plt.xlim(0,16)
xpoints = ypoints = plt.ylim(0,16)
pyplot.plot(xpoints, ypoints, linestyle='--', color='b', lw=1.5)
pyplot.plot(x, yhat, color='red')
pyplot.plot(x, yhat+interval, '--', color='yellow', label="Upper")
pyplot.plot(x, yhat-interval, '--', color='black', label = "Lower")
pyplot.xlabel("Actual")
pyplot.ylabel("Prediction")
pyplot.errorbar(x_in, yhat_out, yerr=interval, color='green', fmt='o')
pyplot.legend(['45 degree line','Goodness of fit line', 'Upper bound', 'Lower bound'], loc=0)
```

```
pyplot.show()

import random
errors = y - x
errors_new = errors.sort_values().reset_index(drop=True)
errors_new.index += 1
numerical_probability = []
for i in range(1,len(errors_new)+1):
    numerical_probability.append((i-0.5)/len(errors_new))
df_ml_errors = pd.DataFrame(list(zip(numerical_probability, errors_new)),columns =['numerical probability', 'errors'])
```

The variable df_ml_errors are the residuals accumulated from repetitive runs with different training and testing data each run. It shows the trend like the Figure E.1 below. Figure E.2. shows the residuals CDF. Based on this, a uniform selection along the curve was done to select the residual to further modify the prediction to enable a probabilistic approach.

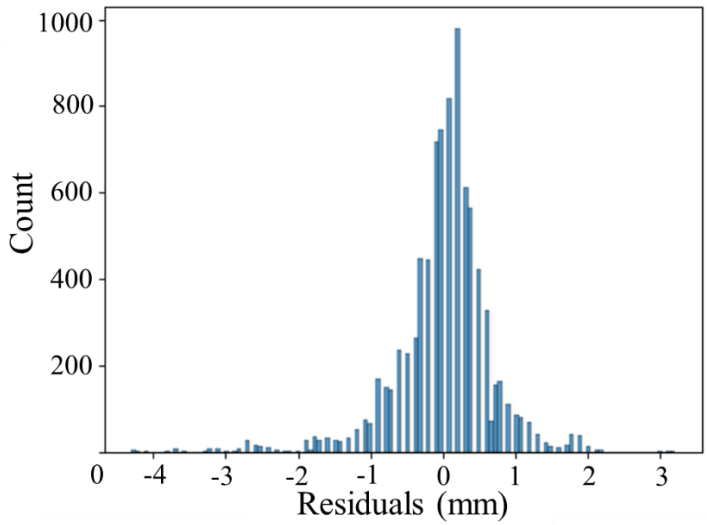


Figure E.1. Residuals distribution

```
### Script for obtaining Cumulative Distribution Function (CDF)
data = df_ml_errors["errors"]
### sort data
x = np.sort(data)
### calculate CDF values
y = 1. * np.arange(len(data)) / (len(data) - 1
### plot CDF
plt.plot(x, y)
```

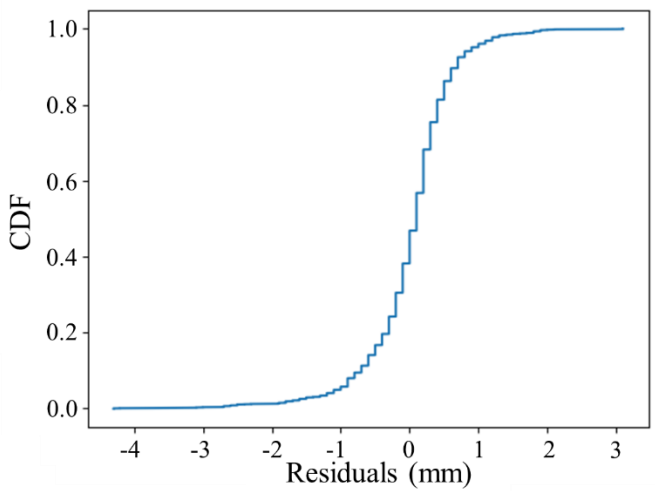


Figure E.2. Residuals and their probability

```
dict_error_vs_cdf = {}
for A, B in zip(y, x):
    dict_error_vs_cdf[A] = B
### dict_error_vs_cdf
dict_error_vs_cdf[np.random.choice(cdf_probability, 1)[0]]
```

Appendix F: Uncertainty Quantification with 90th percentile range predictions

Give the CDF, for 90th percentile range predictions, each year's single value prediction can be modified by a residual uniformly sampled from the CDF (Figure E.2). By repeating this process 2000 times, a 90th percentile range in predictions is achieved. A Python code snippet showing how the prediction is modified is presented below.

```
y_pred = mlp_reg_model.predict(x)
while True:
    error_generated = dict_error_vs_cdf[np.random.choice(cdf_probability, 1)[0]]
    y_pred = y_pred - error_generated
    y_pred = np.round(y_pred[0],3)
    if y_pred > 0:
        break
    else:
        y_pred = mlp_reg_model.predict(x)
```

Appendix G: Python Scripts for NBIF Data Analysis

In this Appendix, the data analysis and RF model development done using Python scripts have been presented, with the example of sampling rate every 20 minutes for Experiment 1 including following processes: 1) Reading sensor data; 2) Data fusion; 3) Data arrangement

for different prediction horizons; 4) Feature Selection; 5) RF model training and testing with visualisation.

G.1 Reading Sensor Data

```
### Read Strain Gauges & LVDTs data
import pandas as pd
sampling_rate = 20
sg_no_lines_to_skip = 23
sg_frequency = 1613
sg_no_of_records_per_cycle = 323
multiplier_sg = 1000000
multiplier_lvdt_1_2 = 2.5
multiplier_lvdt_1_2_base = 12.5
multiplier_lvdt_3_4 = 7.5
multiplier_lvdt_3_4_base = 37.5
df_variables = {}
### Columns to drop
columns_to_drop = ['Maximum SG_v_1','Minimum SG_v_1','SG_v_1 (Max - Min)',
                   'Maximum SG_v_2', 'Minimum SG_v_2', 'SG_v_2 (Max - Min)',
                   'Maximum SG_45_2', 'Minimum SG_45_2', 'SG_45_2 (Max - Min)',
                   'Maximum SG_h_2', 'Minimum SG_h_2', 'SG_h_2 (Max - Min)']
### Every 20 minute
for location_of_file in range(len(sg_files_e1)):
    for index in range(len(sg_files_e1[location_of_file:location_of_file+1])):
        with open(sg_files_e1[location_of_file:location_of_file+1][index]) as f:
            content = f.readlines()
        index_four_mins = []
        every_four_minute = []
        for i in range(len(content)):
            if (i % (sg_frequency*60*sampling_rate-sg_no_lines_to_skip) == sg_no_lines_to_skip+1):
                index_four_mins.append(i)
        for j in index_four_mins:
            every_four_minute.append(content[j:j+323])
        minimum_x_value, maximum_lvdt3, minimum_lvdt3, permanent_surface_displacement_3 = [],
[], [], []
        maximum_lvdt1, minimum_lvdt1, permanent_surface_displacement_1 = [], [], []
        maximum_lvdt2, minimum_lvdt2, permanent_surface_displacement_2 = [], [], []
        maximum_lvdt4, minimum_lvdt4, permanent_surface_displacement_4 = [], [], []
        maximum_sg1_v, minimum_sg1_v, sg1_v_max_min = [], [], []
        maximum_sg1_45, minimum_sg1_45, sg1_45_max_min = [], [], []
        maximum_sg1_h, minimum_sg1_h, sg1_h_max_min = [], [], []
        maximum_sg2_v, minimum_sg2_v, sg2_v_max_min = [], [], []
        maximum_sg2_45, minimum_sg2_45, sg2_45_max_min = [], [], []
        maximum_sg2_h, minimum_sg2_h, sg2_h_max_min = [], [], []

        for interval in every_four_minute:
            x_value, lvdt1, lvdt2, lvdt3, lvdt4, sg1_v, sg1_45, sg1_h, sg2_v, sg2_45, sg2_h =
[],[],[],[],[],[],[],[],[],[],[]
            for each_line in interval:
                x_value.append(float(each_line.split('\t')[0]))
```

```
            lvdt1.append((float(each_line.split('\t')[1])/multiplier_lvdt_1_2_base)*multiplier_lvdt_1_2+
multiplier_lvdt_1_2)
            lvdt2.append((float(each_line.split('\t')[2])/multiplier_lvdt_1_2_base)*multiplier_lvdt_1_2+
multiplier_lvdt_1_2)
            lvdt3.append((float(each_line.split('\t')[3])/multiplier_lvdt_3_4_base)*multiplier_lvdt_3_4+
multiplier_lvdt_3_4)
            lvdt4.append((float(each_line.split('\t')[4])/multiplier_lvdt_3_4_base)*multiplier_lvdt_3_4+
multiplier_lvdt_3_4)
            sg1_v.append(-float(each_line.split('\t')[5])*multiplier_sg)
            sg1_45.append(-float(each_line.split('\t')[6])*multiplier_sg)
            sg1_h.append(-float(each_line.split('\t')[7])*multiplier_sg)
            sg2_v.append(-float(each_line.split('\t')[8])*multiplier_sg)
            sg2_45.append(-float(each_line.split('\t')[9])*multiplier_sg)
            sg2_h.append(-float(each_line.split('\t')[10])*multiplier_sg)
        minimum_x_value.append(min(x_value))
        maximum_lvdt1.append(max(lvdt1))
        minimum_lvdt1.append(min(lvdt1))
        permanent_surface_displacement_1.append(max(lvdt1)-min(lvdt1))
        maximum_lvdt2.append(max(lvdt2))
        minimum_lvdt2.append(min(lvdt2))
        permanent_surface_displacement_2.append(max(lvdt2)-min(lvdt2))
        maximum_lvdt3.append(max(lvdt3))
        minimum_lvdt3.append(min(lvdt3))
        permanent_surface_displacement_3.append(max(lvdt3)-min(lvdt3))
        maximum_lvdt4.append(max(lvdt4))
        minimum_lvdt4.append(min(lvdt4))
        permanent_surface_displacement_4.append(max(lvdt4)-min(lvdt4))

        maximum_sg1_v.append(max(sg1_v))
        minimum_sg1_v.append(min(sg1_v))
        sg1_v_max_min.append(max(sg1_v)-min(sg1_v))
        maximum_sg1_45.append(max(sg1_45))
        minimum_sg1_45.append(min(sg1_45))
        sg1_45_max_min.append(max(sg1_45)-min(sg1_45))
        maximum_sg1_h.append(max(sg1_h))
        minimum_sg1_h.append(min(sg1_h))
        sg1_h_max_min.append(max(sg1_h)-min(sg1_h))
        maximum_sg2_v.append(max(sg2_v))
        minimum_sg2_v.append(min(sg2_v))
        sg2_v_max_min.append(max(sg2_v)-min(sg2_v))
        maximum_sg2_45.append(max(sg2_45))
        minimum_sg2_45.append(min(sg2_45))
        sg2_45_max_min.append(max(sg2_45)-min(sg2_45))
        maximum_sg2_h.append(max(sg2_h))
        minimum_sg2_h.append(min(sg2_h))
        sg2_h_max_min.append(max(sg2_h)-min(sg2_h))

    value = pd.DataFrame(list(zip(minimum_x_value, minimum_lvdt1,
permanent_surface_displacement_1, maximum_lvdt1,
                              minimum_lvdt2, permanent_surface_displacement_2, maximum_lvdt2,
                              minimum_lvdt3, permanent_surface_displacement_3, maximum_lvdt3,
                              minimum_lvdt4, permanent_surface_displacement_4, maximum_lvdt4,
                              maximum_sg1_v, minimum_sg1_v, sg1_v_max_min,
```

```
                        maximum_sg1_45, minimum_sg1_45, sg1_45_max_min,
                        maximum_sg1_h, minimum_sg1_h, sg1_h_max_min,
                        maximum_sg2_v, minimum_sg2_v, sg2_v_max_min,
                        maximum_sg2_45, minimum_sg2_45, sg2_45_max_min,
                        maximum_sg2_h, minimum_sg2_h, sg2_h_max_min)), columns =['Time',
'Minimum LVDT_1', 'Permanent Surface Displacement from LVDT_1', 'Maximum LVDT_1',
                                                        'Minimum LVDT_2', 'Permanent Surface
Displacement from LVDT_2', 'Maximum LVDT_2',
                                                        'Minimum LVDT_3', 'Permanent Surface
Displacement from LVDT_3', 'Maximum LVDT_3',
                                                        'Minimum LVDT_4', 'Permanent Surface
Displacement from LVDT_4', 'Maximum LVDT_4',
                                                        'Maximum SG_v_1', 'Minimum SG_v_1',
'SG_v_1 (Max - Min)',
                                                        'Maximum SG_45_1', 'Minimum SG_45_1',
'SG_45_1 (Max - Min)',
                                                        'Maximum SG_h_1', 'Minimum SG_h_1',
'SG_h_1 (Max - Min)',
                                                        'Maximum SG_v_2', 'Minimum SG_v_2',
'SG_v_2 (Max - Min)',
                                                        'Maximum SG_45_2', 'Minimum SG_45_2',
'SG_45_2 (Max - Min)',
                                                        'Maximum SG_h_2', 'Minimum SG_h_2',
'SG_h_2 (Max - Min)'])
        value_dropped = value.drop(columns=columns_to_drop)
    df_variables["df{0}".format(location_of_file+1)] = value_dropped

df_combined_e1_full_s20 = pd.concat([df_variables["df1"],df_variables["df2"],
df_variables["df3"],df_variables["df4"],
                    df_variables["df5"],df_variables["df6"], df_variables["df7"],df_variables["df8"],
                    df_variables["df9"],df_variables["df10"], df_variables["df11"],df_variables["df12"],
                    df_variables["df13"],df_variables["df14"], df_variables["df15"],df_variables["df16"],
                    df_variables["df17"],df_variables["df18"],df_variables["df19"],df_variables["df20"],
                    df_variables["df21"],df_variables["df22"],df_variables["df23"],df_variables["df24"],
                     df_variables["df25"],df_variables["df26"], df_variables["df27"]], axis=0)
df_combined_e1_full_s20 = df_combined_e1_full_s20.reset_index().drop(columns=['index'])

### Read Pressure Cell data
with open(pc_file_path_e5) as f:
    pc_all_test_e5 = f.readlines() # Pressure cell readings

multiplier_sg = 1000000
multiplier_lvdt = 10
pc_no_lines_to_skip = 4
pc_frequency = 20
pc_no_of_records_per_cycle = 4

index_four_mins = []
every_four_minute = []

for i in range(len(pc_all_test_e5)):
    if (i % (pc_frequency*60*sampling_rate-pc_no_lines_to_skip) == pc_no_lines_to_skip+1):
        index_four_mins.append(i)
```

```
for j in index_four_mins:
    every_four_minute.append(pc_all_test_e5[j:j+pc_no_of_records_per_cycle])

pc1_frequency = 3088.369
pc2_frequency = 3113.477
initial_temp1 = 15.28195
initial_temp2 = 15.2627

## Pressure Cell 1
pc1_a= -6.18796E-7
pc1_b= -8.71533E-2
pc1_c= -pc1_a*(pc1_frequency)**2 - pc1_b*(pc1_frequency)
pc1_t= 2.83717E-1

## Pressure Cell 2
pc2_a= -2.46989E-7
pc2_b= -9.31967E-2
pc2_c= -pc2_a*(pc2_frequency)**2 - pc2_b*(pc2_frequency)
pc2_t= 3.30021E-1

minimum_timestamp, min_no_of_records = [], []
maximum_converted_load_middle, minimum_converted_load_middle,
permanent_converted_load_middle_max_min = [],[],[]
maximum_load_middle, minimum_load_middle, permanent_load_middle_max_min = [],[],[]
maximum_load_corner, minimum_load_corner, permanent_load_corner_max_min = [],[],[]
maximum_pc_temp_middle, minimum_pc_temp_middle, permanent_pc_temp_middle_max_min =
[],[],[]
maximum_pc_temp_corner, minimum_pc_temp_corner, permanent_pc_temp_corner_max_min =
[],[],[]
maximum_converted_load_corner, minimum_converted_load_corner,
permanent_converted_load_corner_max_min = [],[],[]

for interval in every_four_minute:
    timestamp, no_of_records, load_middle, load_corner, pc_temp_middle, pc_temp_corner,
converted_load_middle, converted_load_corner = [],[],[],[],[],[],[],[]

    for each_line in interval:
        timestamp.append(str(each_line.split('\t')[0]))
        no_of_records.append(int(each_line.split(',')[1]))
        load_middle.append(float(each_line.split(',')[2]))
        load_corner.append(float(each_line.split(',')[3]))
        pc_temp_middle.append(float(each_line.split(',')[10]))
        pc_temp_corner.append(float(each_line.split(',')[11]))
        converted_load_middle.append((pc1_a * ((float(each_line.split(",")[2]))**2)+
pc1_b*(float(each_line.split(",")[2]))+ pc1_c + pc1_t * (float(each_line.split(",")[10])-initial_temp1)))
        converted_load_corner.append((pc2_a * ((float(each_line.split(",")[3]))**2)+
pc2_b*(float(each_line.split(",")[3]))+ pc2_c + pc2_t * (float(each_line.split(",")[11])-initial_temp2)))

    minimum_timestamp.append(min(timestamp))
    min_no_of_records.append(min(no_of_records))
    maximum_converted_load_middle.append(max(converted_load_middle))
    minimum_converted_load_middle.append(min(converted_load_middle))
```

```
    permanent_converted_load_middle_max_min.append(max(converted_load_middle)-
min(converted_load_middle))
    maximum_converted_load_corner.append(max(converted_load_corner))
    minimum_converted_load_corner.append(min(converted_load_corner))
    permanent_converted_load_corner_max_min.append(max(converted_load_corner)-
min(converted_load_corner))
    maximum_load_middle.append(max(load_middle))
    minimum_load_middle.append(min(load_middle))
    permanent_load_middle_max_min.append(max(load_middle)-min(load_middle))
    maximum_load_corner.append(max(load_corner))
    minimum_load_corner.append(min(load_corner))
    permanent_load_corner_max_min.append(max(load_corner)-min(load_corner))
    maximum_pc_temp_middle.append(max(pc_temp_middle))
    minimum_pc_temp_middle.append(min(pc_temp_middle))
    permanent_pc_temp_middle_max_min.append(max(pc_temp_middle)-min(pc_temp_middle))
    maximum_pc_temp_corner.append(max(pc_temp_corner))
    minimum_pc_temp_corner.append(min(pc_temp_corner))
    permanent_pc_temp_corner_max_min.append(max(pc_temp_corner)-min(pc_temp_corner))

df_pc_e5_s20 = pd.DataFrame(list(zip(minimum_timestamp, min_no_of_records,
                        maximum_converted_load_middle, minimum_converted_load_middle,
permanent_converted_load_middle_max_min,
                        maximum_converted_load_corner, minimum_converted_load_corner,
permanent_converted_load_corner_max_min,
                        maximum_load_middle, minimum_load_middle,
permanent_load_middle_max_min,
                        maximum_load_corner, minimum_load_corner, permanent_load_corner_max_min,
                        maximum_pc_temp_middle, minimum_pc_temp_middle,
permanent_pc_temp_middle_max_min,
                        maximum_pc_temp_corner, minimum_pc_temp_corner,
permanent_pc_temp_corner_max_min
                       )), columns =['Time', 'No. of Records',
                                'Converted Load (Middle) (Max)', 'Converted Load (Middle) (Min)',
'Converted Load (Middle) (Max - Min)',
                                'Converted Load (Corner) (Max)', 'Converted Load (Corner) (Min)',
'Converted Load (Corner) (Max - Min)',
                                'Load (Middle) (Max)', 'Load (Middle) (Min)', 'Load (Middle) (Max -
Min)',
                                'Load (Corner) (Max)', 'Load (Corner) (Min)', 'Load (Corner) (Max - Min)',
                                'PC Temperature (Middle) (Max)', 'PC Temperature (Middle) (Min)', 'PC
Temperature (Middle) (Max - Min)',
                                'PC Temperature (Corner) (Max)', 'PC Temperature (Corner) (Min)', 'PC
Temperature (Corner) (Max - Min)'])
```

Read Temperature probes data

```
with open(temperature_file_path_e5) as f:
    temp_all_test = f.readlines() # Temperature readings
temp_no_lines_to_skip = 7
temp_every_four_minute = []
temp_all_test_filtered = temp_all_test[temp_no_lines_to_skip:len(temp_all_test)-2]
for i in range(0,len(temp_all_test_filtered),sampling_rate):
    temp_every_four_minute.append(temp_all_test_filtered[i])
```

```
timestamp, no_of_records, t1, t2, t3, t4, t5, t6, t7 = [],[],[],[],[],[],[],[],[]
for each_line in temp_every_four_minute:
        timestamp.append(str(each_line.split(',')[0]))
        no_of_records.append(int(each_line.split(',')[1]))
        t1.append(float(each_line.split(',')[2]))
        t2.append(float(each_line.split(',')[3]))
        t3.append(float(each_line.split(',')[4]))
        t4.append(float(each_line.split(',')[5]))
        t5.append(float(each_line.split(',')[6]))
        t6.append(float(each_line.split(',')[7]))
        t7.append(float(each_line.split(',')[8]))
df_temp_e5_s20 = pd.DataFrame(list(zip(timestamp, no_of_records, t1,t2,t3,t4,t5,t6,t7)), columns
=['Time', 'No_of_records', 't1', 't2', 't3', 't4', 't5', 't6', 't7 (ambient temperature)'])
df_temp_e5_s20 = df_temp_e5_s20.iloc[0:len(df_temp_e5_s20)-2,:]
```

G.2 Sensor Data Fusion

Pressure cell

```
time_stamp_pc = []
for record in df_pc_e5_s20["Time"]:
   time_stamp_pc.append(record.split(",")[0][1:-1])
df_pc_e5_s20["Time_stamp"] = time_stamp_pc
#df_pc_e5_s20 = df_pc_e5_s20.drop(columns = ["Time"])

df_pc_e5_s20_1 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-05 12:02:00"]
df_pc_e5_s20_1 = df_pc_e5_s20_1[df_pc_e5_s20_1["Time_stamp"] <= "2024-02-05 16:02:00"]

df_pc_e5_s20_2 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-06 09:10:00"]
df_pc_e5_s20_2 = df_pc_e5_s20_2[df_pc_e5_s20_2["Time_stamp"] <= "2024-02-06 15:48:00"]

df_pc_e5_s20_3 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-07 08:28:00"]
df_pc_e5_s20_3 = df_pc_e5_s20_3[df_pc_e5_s20_3["Time_stamp"] <= "2024-02-07 16:17:00"]

df_pc_e5_s20_4 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-08 08:44:00"]
df_pc_e5_s20_4 = df_pc_e5_s20_4[df_pc_e5_s20_4["Time_stamp"] <= "2024-02-08 16:00:00"]

df_pc_e5_s20_5 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-09 08:35:00"]
df_pc_e5_s20_5 = df_pc_e5_s20_5[df_pc_e5_s20_5["Time_stamp"] <= "2024-02-09 15:39:00"]

df_pc_e5_s20_6 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-12 09:00:00"]
df_pc_e5_s20_6 = df_pc_e5_s20_6[df_pc_e5_s20_6["Time_stamp"] <= "2024-02-12 16:07:00"]

df_pc_e5_s20_7 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-13 08:30:00"]
df_pc_e5_s20_7 = df_pc_e5_s20_7[df_pc_e5_s20_7["Time_stamp"] <= "2024-02-13 15:46:00"]

df_pc_e5_s20_8 = df_pc_e5_s20[df_pc_e5_s20["Time_stamp"] >= "2024-02-14 08:33:00"]
df_pc_e5_s20_8 = df_pc_e5_s20_8[df_pc_e5_s20_8["Time_stamp"] <= "2024-02-14 16:13:00"]

df_pc_e5_s20_combined = pd.concat([df_pc_e5_s20_1, df_pc_e5_s20_2, df_pc_e5_s20_3,
df_pc_e5_s20_4, df_pc_e5_s20_5,
                    df_pc_e5_s20_6, df_pc_e5_s20_7, df_pc_e5_s20_8], axis=0)
```

```
df_pc_e5_s20_combined = df_pc_e5_s20_combined.reset_index().drop(columns=['index'])
```

Temperature:

```
time_stamp_temp = []
for record in df_temp_e5_s20["Time"]:
    time_stamp_temp.append(record.split(",")[0][1:-1])
df_temp_e5_s20["Time_stamp"] = time_stamp_temp
df_temp_e5_s20_1 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-05 12:02:00"]
df_temp_e5_s20_1 = df_temp_e5_s20_1[df_temp_e5_s20_1["Time_stamp"] <= "2024-02-05
16:02:00"]

df_temp_e5_s20_2 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-06 09:10:00"]
df_temp_e5_s20_2 = df_temp_e5_s20_2[df_temp_e5_s20_2["Time_stamp"] <= "2024-02-06
15:48:00"]
df_temp_e5_s20_3 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-07 08:28:00"]
df_temp_e5_s20_3 = df_temp_e5_s20_3[df_temp_e5_s20_3["Time_stamp"] <= "2024-02-07
16:17:00"]
df_temp_e5_s20_4 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-08 08:44:00"]
df_temp_e5_s20_4 = df_temp_e5_s20_4[df_temp_e5_s20_4["Time_stamp"] <= "2024-02-08
16:00:00"]
df_temp_e5_s20_5 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-09 08:35:00"]
df_temp_e5_s20_5 = df_temp_e5_s20_5[df_temp_e5_s20_5["Time_stamp"] <= "2024-02-09
15:39:00"]
df_temp_e5_s20_6 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-12 09:00:00"]
df_temp_e5_s20_6 = df_temp_e5_s20_6[df_temp_e5_s20_6["Time_stamp"] <= "2024-02-12
16:07:00"]
df_temp_e5_s20_7 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-13 08:30:00"]
df_temp_e5_s20_7 = df_temp_e5_s20_7[df_temp_e5_s20_7["Time_stamp"] <= "2024-02-13
15:46:00"]
df_temp_e5_s20_8 = df_temp_e5_s20[df_temp_e5_s20["Time_stamp"] >= "2024-02-14 08:33:00"]
df_temp_e5_s20_8 = df_temp_e5_s20_8[df_temp_e5_s20_8["Time_stamp"] <= "2024-02-14
16:13:00"]
df_temp_e5_s20_combined = pd.concat([df_temp_e5_s20_1, df_temp_e5_s20_2, df_temp_e5_s20_3,
df_temp_e5_s20_4, df_temp_e5_s20_5, df_temp_e5_s20_6, df_temp_e5_s20_7, df_temp_e5_s20_8],
axis=0)
df_temp_e5_s20_combined = df_temp_e5_s20_combined.reset_index().drop(columns=['index'])
df_temp_and_pc_e5_s20_combined = pd.concat([df_temp_e5_s20_combined,
df_pc_e5_s20_combined], axis=1)
df_temp_and_pc_e5_s20_combined =
df_temp_and_pc_e5_s20_combined.reset_index().drop(columns=['index'])
```

Strain Gauges & LVDTs

```
df_combined_e5_full_s20_1 = df_combined_e5_full_s20.iloc[:12,:]
df_combined_e5_full_s20_2 = df_combined_e5_full_s20.iloc[12:12+20,:] # 200
df_combined_e5_full_s20_3 = df_combined_e5_full_s20.iloc[12+20:12+20+23,:] # 235
df_combined_e5_full_s20_4 = df_combined_e5_full_s20.iloc[12+20+23:12+20+23+22,:] # 218
df_combined_e5_full_s20_5 = df_combined_e5_full_s20.iloc[12+20+23+22:12+20+23+22+22,:]
df_combined_e5_full_s20_6 =
df_combined_e5_full_s20.iloc[12+20+23+22+22:12+20+23+22+22+21,:] # 214
df_combined_e5_full_s20_7 =
df_combined_e5_full_s20.iloc[12+20+23+22+22+21:12+20+23+22+22+21+22,:]
```

```
df_combined_e5_full_s20_8 =
df_combined_e5_full_s20.iloc[12+20+23+22+22+21+22:12+20+23+22+22+21+22+23,:]
df_combined_e5_full_s20_combined = pd.concat([df_combined_e5_full_s20_1,
df_combined_e5_full_s20_2, df_combined_e5_full_s20_3, df_combined_e5_full_s20_4,
df_combined_e5_full_s20_5, df_combined_e5_full_s20_6, df_combined_e5_full_s20_7,
df_combined_e5_full_s20_8], axis=0)
df_combined_e5_full_s20_combined =
df_combined_e5_full_s20_combined.reset_index().drop(columns=['index'])
df_combined_e5_s20_all = pd.concat([df_combined_e5_full_s20_combined,
df_pc_e5_s20_combined, df_temp_e5_s20_combined], axis=1)
df_combined_e5_s20_all = df_combined_e5_s20_all.reset_index().drop(columns=['index'])
```

G.3 Data Preparation for Different Prediction Scenarios

```
df_combined_e5_s20_all = df_combined_e5_s20_all.drop(columns = ["Time", "No. of Records",
"Time_stamp", "Time", "No_of_records", "Time_stamp", "PC Temperature (Middle) (Max - Min)",
"PC Temperature (Corner) (Max - Min)"])

### To predict the permanent strain
cols_to_move = ['Minimum LVDT_3']

### move columns to front
df_combined_e5_s20_all = df_combined_e5_s20_all[cols_to_move + [x for x in
df_combined_e5_s20_all.columns if x not in cols_to_move]]
df_combined_e5_s20_all = df_combined_e5_s20_all.drop(columns= ["Maximum LVDT_3",
"Permanent Surface Displacement from LVDT_3"])
df_combined_e5_s20_all = df_combined_e5_s20_all.drop(columns = ["Permanent Surface
Displacement from LVDT_4"])
df_combined_e5_s20_all = df_combined_e5_s20_all.drop(columns = ["Maximum LVDT_4",
"Minimum LVDT_4"])

### Prepare to predict next 20, 40, and 60 mins
df_combined_e5_s20_all_copy_20 = df_combined_e5_s20_all.copy(deep = True)
df_combined_e5_s20_all_copy_40 = df_combined_e5_s20_all.copy(deep = True)
df_combined_e5_s20_all_copy_60 = df_combined_e5_s20_all.copy(deep = True)

df_combined_e5_s20_all_copy_20['Minimum LVDT_3'] =
df_combined_e5_s20_all_copy_20['Minimum LVDT_3'].shift(-1)
df_combined_e5_s20_all_copy_40['Minimum LVDT_3'] =
df_combined_e5_s20_all_copy_40['Minimum LVDT_3'].shift(-2)
df_combined_e5_s20_all_copy_60['Minimum LVDT_3'] =
df_combined_e5_s20_all_copy_60['Minimum LVDT_3'].shift(-3)

df_combined_e5_s20_all_copy_20 = df_combined_e5_s20_all_copy_20.dropna()
df_combined_e5_s20_all_copy_40 = df_combined_e5_s20_all_copy_40.dropna()
df_combined_e5_s20_all_copy_60 = df_combined_e5_s20_all_copy_60.dropna()
==========================================================================
==
```

The same scripts were written for Experiment 2, then both data was combined together before the next step.

```
df_combined_e5_e6_s20_all = pd.concat([df_combined_e5_s20_all, df_combined_e6_s20_all],
axis=0)
```

```
df_combined_e5_e6_s20_all = df_combined_e5_e6_s20_all.reset_index().drop(columns=['index'])
df_combined_e5_e6_s20_all_copy_20 = pd.concat([df_combined_e5_s20_all_copy_20,
df_combined_e6_s20_all_copy_20], axis=0)
df_combined_e5_e6_s20_all_copy_20 =
df_combined_e5_e6_s20_all_copy_20.reset_index().drop(columns=['index'])
df_combined_e5_e6_s20_all_copy_40 = pd.concat([df_combined_e5_s20_all_copy_40,
df_combined_e6_s20_all_copy_40], axis=0)
df_combined_e5_e6_s20_all_copy_40 =
df_combined_e5_e6_s20_all_copy_40.reset_index().drop(columns=['index'])
df_combined_e5_e6_s20_all_copy_60 = pd.concat([df_combined_e5_s20_all_copy_60,
df_combined_e6_s20_all_copy_60], axis=0)
df_combined_e5_e6_s20_all_copy_60 =
df_combined_e5_e6_s20_all_copy_60.reset_index().drop(columns=['index'])
```

G.4 Feature Selection

```
### pearson's correlation feature selection for numeric input and numeric output
from sklearn.datasets import make_regression
from sklearn.feature_selection import SelectKBest

X = df_combined_e5_e6_s20_all[df_combined_e5_e6_s20_all.columns[1:].tolist()]
y = df_combined_e5_e6_s20_all[['Minimum LVDT_3']]

### define feature selection
fs = SelectKBest(score_func=f_regression, k=10)

### apply feature selection
X_selected_copy_20 = fs.fit_transform(X, y.values.ravel())
print(X_selected_copy_20.shape)
```

G.5 RF Model Training and Testing on Sensor Data

```
import numpy as np
import matplotlib.pyplot as plt
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression

X = X_selected_copy_20
y = y

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

### fit the model
model = RandomForestRegressor(random_state=1).fit(X_train, y_train.values.ravel())
```

```
### make predictions
yhat = model.predict(X_test)
### evaluate predictions
mae = mean_absolute_error(y_test, yhat)
print('MAE: %.4f' % mae)
r2 = r2_score(y_test, yhat)
print('R2: %.4f' % r2)
rmse = mean_squared_error(y_test, yhat, squared = False)
print('RMSE: %.4f' % rmse)
### find line of best fit
a, b = np.polyfit(y_test["Minimum LVDT_3"], yhat, 1)

### add points to plot
plt.scatter(y_test["Minimum LVDT_3"], yhat)

### Define the data for the x and y axis
x_values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y_values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

### Plot the line
plt.plot(x_values, y_values, linestyle='--', color='gold', label = '45 degree line')

### add line of best fit to plot
plt.plot(y_test["Minimum LVDT_3"], a*y_test["Minimum LVDT_3"]+b, color = 'r', label =
'Goodness of fit')
plt.scatter(y_test, yhat, color = 'b')
plt.xlim(1, 9.25)
plt.ylim(1, 9.25)
plt.xlabel('Actual Permanent Surface Displacement from LVDT_3')
plt.ylabel('Predicted Permanent Surface Displacement from LVDT_3')
plt.title('Actual vs Predicted Permanent Surface Displacement from LVDT_3 - (sampling rate every
20 minute - predict concurrently)')
plt.legend(loc="upper left")
plt.show()
```