UTS UNIVERSITY OF TECHNOLOGY SYDNEY

AI-Driven Bridge Condition Assessment under Moving Vehicles Utilising Transfer Learning Techniques

by Saeid Talaei

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Prof. Xinqun Zhu and Prof. Jianchun Li

University of Technology Sydney

Faculty of Engineering and Information Technology

October 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Saeid Talaei, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Civil and Environmental Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

Signature

Production Note:
Signature removed prior to publication.

Date: 22/10/2025

ACKNOWLEDGEMENTS

My deepest gratitude belongs to my knowledgeable, intellectual, and gentle principal supervisor, Dr. Xinqun Zhu. His continuous technical and moral support was the main reason I could pursue my PhD studies. A considerate supervisor who helped me through the tough times. He was consistently supportive and responsive, enlightening me with novel ideas. His kindness and expertise in technical details kept me on track. A dedicated supervisor like him is all a PhD student could ever hope for.

I want to express my heartfelt thanks to my other supervisor, Prof. Jianchun Li, a true mentor with a big heart. He is a teacher who helps his students grow in small steps and was far kinder to me than I deserve. Our chats by phone or over coffee always taught me how to see the big picture, both in research and in life.

I want to extend my gratitude to Prof. Bijan Samali, my mentor and role model. I am proud to know him and have learned peace and mindfulness from his grace. He supported me during my time of need and treated me like family.

I would also like to thank Prof. Tommy Chan, who helped me move to Brisbane and welcomed me into his research group at QUT. His vast knowledge and discipline guided me throughout my PhD studies.

I am grateful to Dr Yang Yu, who helped me polish my ideas and find my errors.

To my research group and friends: Jiajia, Xutong, Bing, Mingzhe, and Ye, thank you. We learned a lot together and shared our ideas in weekly meetings. I believe you all deserve the best, because you are the best.

I, with all of my heart, thank my dear parents; their support has shaped who I am, and their pride drives my constant pursuit. They taught me to move forward without looking back, to aim for the best, and to strive to achieve it. I also thank my dear sister Sima, who refreshes my spirit with her smiles.

Finally, I dedicate this thesis to my wife, my heart and love, Sage. Without her, everything is just meaningless.

LIST OF PUBLICATIONS

1. **Talaei, S.**, Zhu, X., and Li, J. (2022). "Transfer learning-based condition assessment for bridges". *Proceedings of the 3rd International Conference on Structural Engineering Research*, pp. 190-195.

2. **Talaei, S.**, Zhu, X., Li, J., Yu, Y., and Chan, T. H. T. (2023). "Transfer learning-based bridge damage detection: Leveraging time-frequency features". *Structures*, 57, 105052.

3. **Talaei, S.**, Zhu, X., Li, J., Yu, Y., & Chan, T. H. (2025). A hybrid domain adaptation approach for estimation of prestressed forces in prestressed concrete bridges under moving vehicles. *Engineering Structures*, 330, 119904.

4. **Talaei, S.**, Zhu, X., Li, J., Yu, Y., and Chan, T. H. T. (2025). Domain-Generalised Coupled Multi-Output Autoencoder for Vehicle-Bridge Interaction Feature Extraction and Damage Identification *(Under Preparation)*

TABLE OF CONTENTS

CERTIFICATE OF ORIGINAL AUTHORSHIP ..... **i**
ACKNOWLEDGEMENTS ..... **ii**
LIST OF PUBLICATIONS ..... **iii**
TABLE OF CONTENTS ..... **iv**
LIST OF FIGURES ..... **viii**
LIST OF TABLES ..... **xiii**
ABSTRACT ..... **xiv**
1 CHAPTER 1 - INTRODUCTION **1**
1.1 Background **1**
1.2 Aim and Objectives **3**
1.3 Research Significance **4**
1.4 Structure of the Thesis **4**
2 CHAPTER 2 - LITERATURE REVIEW **6**
2.1. Overview **6**
2.2. Structural Health Monitoring **7**
2.3. Vibration-Based Structural Damage Detection **8**
2.3.1. Model-Based and Data-Driven Approaches 9
2.3.2. Damage-Sensitive Parameters 11
2.4. AI-Driven Structural Damage Detection **15**
2.4.1. Unsupervised Structural Damage Detection 16
2.4.1.1. Novelty Detection Methods 16
2.4.1.2. Clustering Algorithms 17
2.4.2. Supervised Structural Damage Detection 17
2.4.2.1. Artificial Neural Network (ANN) 18
2.4.2.2. Support Vector Machine (SVM) 18
2.4.2.3. K-Nearest Neighbours (KNN) 19

2.4.2.4. Decision Trees and Random Forest 20
2.4.2.5. Gaussian Process (GP) 20
2.4.2.6. Sparse Representation Classification (SRC) 21
2.4.2.7. Ensemble Methods 21
2.5. Deep Learning-Based Structural Damage Detection 21
2.5.1. Deep Autoencoders 22
2.5.2. Deep Convolutional Neural Network (DCNN) 23
2.5.3. Recurrent Neural Network (RNN) 25
2.6. Transfer Learning 26
2.6.1. When to Transfer? 27
2.6.2. What to Transfer? 27
2.6.2.1. Instance-Based TL Approaches 27
2.6.2.2. Relational-Based TL Approaches 27
2.6.2.3. Parameter-Based TL Approaches 28
2.6.2.4. Feature-Based TL Approaches 28
2.6.3. How to Transfer? 28
2.6.4. Domain Adaptation-Based Structural Damage Detection 34
2.6.5. Transfer the Knowledge from FE Models to Real Structures 36
2.7. Summary 38
3 CHAPTER 3 - STRUCTURAL PARAMETER IDENTIFICATION FOR BRIDGE STRUCTURES USING FINE-TUNING TECHNIQUE AND FREQUENCY RESPONSE FUNCTIONS 40
3.1 Overview 40
3.2 Proposed Methodology 41
3.2.1 Deep Learning Framework for Bridge Condition Assessment 41
3.2.2 Convolutional Neural Networks (CNNs) 42
3.2.3 Transfer Learning Technique 47
3.3 Case Study 50
3.3.1 Bridge Finite Element Model 50
3.3.2 Vehicle Location Estimation from Bridge Responses Using DL 52
3.3.3 Fine-Tuning 59

3.4 Summary **63**

4 CHAPTER 4 - TRANSFER LEARNING BASED BRIDGE DAMAGE DETECTION: LEVERAGING TIME-FREQUENCY FEATURES **64**

4.1 Overview **64**

4.2 Related Works **65**

4.3 Proposed Methodology **67**

4.3.1 Vehicle-Bridge Interaction (VBI) Analysis 68

4.4 Damage Simulation **70**

4.5 Time-Frequency Features **71**

4.6 Transfer Learning-Based Structural Damage Detection Method **73**

4.6.1 Pretrained Convolutional Neural Networks 74

4.6.2 Classification Algorithms 75

4.7 Case Study **77**

4.7.1 Damage Classification Using Transfer Learning 79

4.7.2 Investigating the Effect of Noise in Responses on Performance 85

4.7.3 Investigating the Effect of Sensor Location on Classification Performance 88

4.7.4 Investigating the Effect of Vehicle Speed on Classification Performance ... 89

4.7.5 Investigating the Effect of Surface Roughness on Performance 94

4.8 Discussions **98**

4.9 Summary **99**

5 CHAPTER 5 - HYBRID DOMAIN ADAPTATION APPROACH FOR ESTIMATION OF PRESTRESSED FORCES IN PSC BRIDGES **100**

5.1 Overview **100**

5.2 Related Works **101**

5.3 Proposed Methodology **104**

5.3.1 Prestressed Force Simulation in VBI Model 104

5.3.2 DL-Based PF-Sensitive Feature Extraction 105

5.4 A Novel Hybrid Domain Adaptation Method for PF Prediction **108**

5.4.1 Domain Adaptation (DA) 108

5.4.2 A Novel Hybrid DA-Based Method 109

5.5 Evaluation of the Proposed Method **111**

5.5.1 Training Data Setup 111

5.5.2 Model Architecture and Training Process 112

5.5.3 Testing and Evaluation 114
5.5.4 Further Explorations on a Target with Multiple Deviations 119
5.5.5 Network Hyperparameters Study 120
5.5.6 Evaluating Model Robustness Across Unseen Domains 122
5.6 Discussions 124
5.7 Summary 125
6 CHAPTER 6 - DOMAIN-GENERALISED AUTOENCODER FOR BRIDGE DAMAGE IDENTIFICATION 126
6.1 Overview 126
6.2 Related Works 127
6.3 Data Acquisition and Preprocessing 130
6.3.1 Data Acquisition from the VBI Finite-Element Model 130
6.3.2 Signal Preprocessing Utilising Empirical Mode Decomposition 130
6.4 AE Architecture Study for Reliable Bridge Acceleration Reconstruction 131
6.4.1 Feature Extraction and Crack Severity Detection 138
6.5 Domain Generalisation 139
6.5.1 Domain Generalised Dataset Generation Utilising LHS 140
6.6 Autoencoder Loss and Hyperparameter Optimisation 143
6.6.1 Hybrid Loss Function Definition 143
6.6.2 Hyperparameter Optimisation Utilising Optuna 144
6.7 Evaluation of the Autoencoder Performance 146
6.8 Dual-Head Autoencoder for Crack Signature Function Reconstruction 150
6.8.1 Crack Signature Function 151
6.8.2 Dual-Head Autoencoder Model Description and Training 152
6.8.3 Crack Signature Function Reconstruction Results 154
6.9 Conclusion and Discussion 161
6.10 Summary 162
7 CHAPTER 7 - CONCLUSIONS AND RECOMMENDATIONS 164
7.1 Conclusions 164
7.2 Recommendations for Future Works 166
REFERENCES 167

LIST OF FIGURES

Figure 1.1 Flowchart of the thesis structure 5
Figure 2.1 Specialisation of AI algorithms 16
Figure 2.2 An illustration of the hyperplane for separation in the non-linear case 19
Figure 2.3 A schematic illustration of kNN algorithm with k = 3 and k = 6 20
Figure 2.4 Deep Auto Encoders (Jiang et al., 2021) 22
Figure 2.5 Structure of Wang et al. proposed DCNN-based DI model 24
Figure 2.6 An example of fine-tuning for vision-based structural damage recognition (Gao & Mosalam, 2018) 29
Figure 2.7 Domain Adaptation 31
Figure 2.8 Domain adaptation categorisation based on the shared classes between domains (You et al., 2019) 32
Figure 2.9 Deep Reconstruction-based Domain Adaptation (Ghifary et al., 2016) 33
Figure 2.10 Adversarial deep domain adaptation (Ganin & Lempitsky, 2015) 33
Figure 2.11 Domain adaptation between source and target 35
Figure 2.12 Domain adaptation between numerical model and real structure 36
Figure 2.13 (a) closed set domain adaptation in (b) partial domain adaptation for damage detection (Poole et al., 2022) 37
Figure 3.1 Deep Learning Framework 42
Figure 3.2 A typical convolutional neural network (Azimi & Pekcan, 2020) 42
Figure 3.3 An example of a convolution kernel (Lin et al., 2017) 43
Figure 3.4 Common activation functions: (a)Sigmoid, (b) ReLU and (c)leaky ReLU 44
Figure 3.5 Demonstration of a 1-D max pooling (Zhang et al., 2019) 46
Figure 3.6 Dropout operation (Yu et al., 2019) 46
Figure 3.7 (a)Inductive and (b)Transductive TL approaches 48
Figure 3.8 Different fine-tuning strategies based on target domain data size and its similarity to source domain data (Azimi & Pekcan, 2020) 49
Figure 3.9 Finite element model of the composite bridge in ANSYS 2021 50

Figure 3.10 First three bending and torsional modeshapes and natural frequencies 51
Figure 3.11 Location of a) accelerometers, and b) impact force on the bridge deck..... 52
Figure 3.12 Time history of a) the impact force and b) bridge acceleration response in Y direction captured by sensor 13..... 52
Figure 3.13 Added mass on the bridge deck..... 53
Figure 3.14 Samples of the CNN input dataset..... 54
Figure 3.15 a sample of the CNN structure 55
Figure 3.16 Differences between source and target domains 60
Figure 3.17 t-SNE visualisation of source and target domain features..... 61
Figure 3.18 Confusion matrix of the target classification after fine-tuning using DNN FC classifier 62
Figure 3.19 Confusion matrix of the target classification after fine-tuning using a random forest classifier 62
Figure 3.20 Confusion matrix of the target classification after fine-tuning using an ensemble of classifiers 62
Figure 4.1: The VBI model (Li et al., 2020) 68
Figure 4.2 Equivalent nodal loads for a beam element loaded by the bridge-vehicle interaction force 69
Figure 4.3 Deflections at 3/8 span of the beam subjected to: (a) Light moving loads; (b) heavy moving loads (___measured; ... VBI model) (Law & Zhu, 2004) 70
Figure 4.4 Representation of VBI acceleration response of the bridge in time, frequency, and time-frequency domains..... 72
Figure 4.5 Structure of the proposed method..... 74
Figure 4.6 Two-axle moving vehicle-bridge system..... 77
Figure 4.7 Samples of VBI acceleration response in time-domain for the Intact structure, 2nd, and 8th damage classes..... 78
Figure 4.8 Time-frequency representation of VBI acceleration responses for different damage scenarios 79
Figure 4.9 comparison of classification metrics of different classifiers utilised to fine-tune each pretrained network..... 83
Figure 4.10 MCC comparison of damage classification..... 84
Figure 4.11 ROC curve comparison of damage classification utilising the features extracted from AlexNet..... 84

Figure 4.12 Confusion Matrix of the selected pretrained network (AlexNet+ESD) 84
Figure 4.13 effect of noise in acceleration data on classification metrics 86
Figure 4.14 effect of noise in acceleration data on the ROC curve of the classification test results 86
Figure 4.15 Confusion Matrices of the selected pretrained network (AlexNet+ESD) with different SNR in acceleration data 87
Figure 4.16 explored potential transducer installation locations on the bridge span 88
Figure 4.17 classification metrics representing the effect of sensor location 89
Figure 4.18 effect of sensor location on the ROC curve of the classification test results 89
Figure 4.19 Samples of time-frequency damage-sensitive features in different vehicle speeds 90
Figure 4.20 VBI interaction force of the front and rear wheels in different vehicle speeds 90
Figure 4.21 classification metrics of models trained with different vehicle speeds 91
Figure 4.22 ROC curves of models trained with different vehicle speeds..... 91
Figure 4.23 Confusion Matrices of the selected pretrained network trained and tested (AlexNet+ESD) with different vehicle speeds..... 92
Figure 4.24 Effect of vehicle speed on classification metrics..... 93
Figure 4.25 Effect of vehicle speed on the ROC curve of the classification test results 93
Figure 4.26 Classification metrics of models trained with different surface roughness . 95
Figure 4.27 ROC curves of models trained with different surface roughness 95
Figure 4.28 Confusion Matrices of the selected pretrained network (AlexNet+ESD) trained and tested with different surface roughness classes..... 96
Figure 4.29 Effect of surface roughness on classification metrics..... 97
Figure 4.30 Effect of surface roughness on the ROC curve of the classification test results 97
Figure 5.1 Overall methodology of the proposed approach..... 104
Figure 5.2 Composing 3-band colored image by combining CWT data from three sensors 106
Figure 5.3 The flowchart for evaluating the performance of the AlexNet model..... 107
Figure 5.4 Feature extraction using AlexNet and correlation of features to PF 107
Figure 5.5 Framework of the proposed hybrid DA method..... 109

Figure 5.6 Evolution of loss values during training the proposed hybrid DA model ... 112
Figure 5.7 Feature distribution of source and target domain representative features at 100% PF condition before and after hybrid DA 113
Figure 5.8 Feature distribution of target domain representative features for the targets with bridge modelling error at 100% PF condition..... 115
Figure 5.9 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 1 (bridge's length modelling error) 115
Figure 5.10 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 2 (bridge's rigidity modelling error)..... 115
Figure 5.11 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 3 (bridge's mass modelling error) 116
Figure 5.12 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 4 (bridge's damping modelling error) 116
Figure 5.13 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 5 (bridge's boundary conditions modelling error)..... 116
Figure 5.14 Feature distribution of target domain representative features for the target with road roughness modelling error at 100% PF condition..... 117
Figure 5.15 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 6 (road roughness modelling error) 117
Figure 5.16 Feature distribution of target domain representative features for the target with vehicle modelling error at 100% PF condition 118
Figure 5.17 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 7 (vehicle speed error) 118
Figure 5.18 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 8 (vehicle modelling error) 119
Figure 5.19 Feature distribution of target domain representative features for the target with multiple deviations at 100% PF condition 119
Figure 5.20 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 9 (multiple deviations)..... 120
Figure 5.21 Regression Results for Target 9 with Carefully Selected Hyperparameters 122
Figure 5.22 Model performance evaluation across unseen domains 123

Figure 6.1 Baseline correction of the vehicle acceleration responses 131
Figure 6.2 A sample of the reconstructed bridge acceleration signals using CNN-based AE 133
Figure 6.3 A sample of the reconstructed bridge acceleration signals using LSTM-based AE 134
Figure 6.4 A sample of the reconstructed bridge acceleration signals using CNN+LSTM-based AE 135
Figure 6.5 The selected AE architecture combining CNN and LSTM layers 136
Figure 6.6 Extracted Features from the AE latent space 138
Figure 6.7 Damage severity regression results 139
Figure 6.8 LHS generated samples across different labels 142
Figure 6.9 Train and validation loss curves for different loss components 147
Figure 6.10 Examples of the reconstructed signals by the optimised AE 148
Figure 6.11 The first 3 PCA components (left) and the tSNE representation (right) of the extracted features coloured by the different domain-variation labels. 149
Figure 6.12 Damage severity regression results on the generalised dataset 150
Figure 6.13 The proposed dual-head autoencoder architecture 150
Figure 6.14 Crack stiffness reduction function (Wahab et al. 1999) 151
Figure 6.15 Total Train and validation loss curves for the dual-head AE 152
Figure 6.16 Examples of the dual-head AE reconstruction results 155
Figure 6.17 Examples of the dual-head AE reconstruction results 156
Figure 6.18 Examples of the dual-head AE reconstruction results 157
Figure 6.19 Examples of the dual-head AE reconstruction results 158
Figure 6.20 Examples of the dual-head AE reconstruction results 159
Figure 6.21 Examples of the dual-head AE reconstruction results 160

LIST OF TABLES

Table 3.1 The structure of CNN No.1 and loss graphs during the training process 56
Table 3.2 The structure of CNN No.2 and loss graphs during the training process 57
Table 3.3 The structure of CNN No.3 and loss graphs during the training process 58
Table 3.4 Classification accuracy of CNN No.3 tested with different SNR..... 58
Table 3.5 Differences between source and target structures in material properties..... 59
Table 3.6 Differences between source and target natural frequencies..... 59
Table 4.1 Parameters of the bridge 77
Table 4.2 Parameters of the vehicle 77
Table 4.3 Summary of the pretrained networks 80
Table 4.4 Optimised hyperparameters of the selected classifiers 81
Table 4.5 MCC comparison of damage classification 84
Table 4.6 classification metrics representing the effect of noise in acceleration data 86
Table 4.7 Classification metrics representing the effect of sensor location..... 88
Table 4.8 Classification metrics of models trained with different vehicle speeds..... 91
Table 4.9 Effect of vehicle speed on classification metrics 93
Table 4.10 The classification of road surface roughness 94
Table 4.11 Classification metrics of models trained with different surface roughness .. 95
Table 4.12 Classification metrics representing the effect of surface roughness class 97
Table 5.1 Definition of different target domains for VBI systems 114
Table 5.2 Model Hyperparameters Studies..... 121
Table 6.1 Three different AE architectures..... 132
Table 6.2 Performance comparison between the three different AE architectures..... 136
Table 6.3 Comparison between domain generalisation and other methods..... 140
Table 6.4 Parameter ranges to create the domain-generalised dataset..... 142
Table 6.5 Optuna search range and selected hyperparameters 146
Table 6.6 Dual-head AE model description..... 153

ABSTRACT

Bridges are critical civil infrastructures that often require reliable monitoring and condition assessments to prevent high-consequence failures. AI-driven bridge condition assessment methods that utilise structural dynamic responses, such as vehicle-bridge interaction (VBI) acceleration data, have demonstrated high performance in recent years. However, there are significant challenges that limit their implementation in the real world, including labelled data scarcity and the domain shifts due to the discrepancies between numerical models and the real structures.

This thesis proposes novel transfer learning (TL) techniques to enhance AI-based bridge condition assessment under moving vehicles. Advanced deep-learning models, including convolutional neural networks (CNNs) and deep autoencoders, are developed to optimise feature extraction from vehicle-bridge interaction (VBI) responses. The TL strategies include (i) fine-tuning for precise structural-parameter identification, (ii) leveraging pretrained networks fed with time-frequency features for damage localisation, (iii) a hybrid domain-adaptation approach for estimating prestress forces with only one label from the target domain, and (iv) domain-generalised autoencoders that identify crack shapes without any target-domain labels. Applied TL techniques transfer knowledge between different numerical datasets, mitigating label scarcity and domain shift issues.

The outcomes of this thesis exhibit advancements in robustness of condition assessment of bridges under various types of domain shifts, including modelling errors and sensor noise. Numerical validations of the proposed approaches confirm that reliable condition assessment is achievable even in the presence of high data scarcity. Finally, the proposed AI-driven methodologies bring current bridge condition assessment techniques one step closer to real-world implementation.

CHAPTER 1 - INTRODUCTION

1.1 Background

Bridges are an inseparable part of transport networks, whose failure will cause catastrophic economic and societal risks. Real-time monitoring of bridge behaviour can significantly contribute to their safety. The concept of structural health monitoring refers to the practice of structural maintenance by continuously observing its behaviour. Bridges face various operational loads and environmental conditions that can accelerate their structural deterioration. Factors like fatigue, corrosion, and others may lead to damage. Structural damage is defined as changes in structural geometry or material that impact the structure's performance, usability, dependability, and service life (Avci et al., 2021). Some of the damage could propagate and eventually cause the collapse of the structure. Therefore, continuous monitoring of the structure's health during its lifespan is essential for important bridges.

Vibration-based techniques for damage detection are commonly used to evaluate the health condition of structures. The structure's dynamic behaviour is linked to its structural health status. Variations in measured dynamic responses can be employed to identify damage to the structure. Typically, damage is characterised by a reduction in stiffness within a specific element. (Sarmadi et al., 2022).

Recent advancements in Artificial Intelligence (AI) and the growing demand to utilise AI for solving engineering problems have led to new developments in applying AI-driven methods for structural health monitoring of civil infrastructure, such as bridges. Among AI-driven methods, Machine learning (ML)-based methods use data as their input to train a statistical model for structural damage identification. Therefore, ML-based methods are fundamentally data-driven, relying on data to function. In these approaches, Finite Element models serve as the main tools for generating datasets of features linked to various damage scenarios. It is crucial that these features are sensitive enough to detect damage while remaining less affected by operational and environmental variables. ML-based methods are categorised into four main types: (i) unsupervised learning, (ii)

supervised learning, (iii) semi-supervised learning, and (iv) reinforcement learning (RL). Unsupervised learning requires data in the structure's intact state for training. The data from the current condition of the structure will be compared to the intact condition using the trained model during the monitoring phase in order to identify any deviations from the undamaged state (Fugate et al., 2000). For supervised damage detection, a labelled dataset that includes damage-sensitive features and the corresponding labels (e.g. location and severity) for a variety of structural health states is necessary for training (Figueiredo et al., 2011). Semi-supervised learning acts as a middle ground between supervised and unsupervised approaches by using both labelled and unlabeled data for training. This technique can improve model performance, particularly when labelled data is scarce or expensive to obtain. In contrast, reinforcement learning (RL) operates without labelled data or external supervision; instead, the model learns optimal actions through interactions with its environment, guided by a reward-based trial-and-error mechanism (Khazaeli et al., 2021).

Deep Learning (DL) shows more promising results compared to traditional ML algorithms, and it can learn patterns from structural vibrational response and relate them to structural health conditions. DL is an advanced branch of machine learning capable of automatically learning and extracting relevant features directly from raw data, often leading to superior performance in both classification and regression tasks. It employs multiple layers of interconnected processing units to progressively capture higher-level representations of the input data through hierarchical abstraction. Thus, the feature extraction process is no longer a distinct issue. As a result, DL algorithms are excellent for detecting structural damage using the raw signal data (Krizhevsky et al., 2017). Three of the main DL approaches are deep autoencoders, deep convolutional neural networks (CNNs) and recurrent neural networks (RNNs) such as long short-term memory networks (LSTMs) (Lin et al., 2017).

Damage in structures is typically a localised phenomenon, making global modal parameters such as natural frequencies and mode shapes insufficient for precise local damage detection. Incorporating the Vehicle-Bridge Interaction (VBI) model can help reduce uncertainty and enhance the accuracy of structural identification in bridge systems (Kim et al., 2011). A coupled VBI system can be characterised as a time-varying system. (Zhu & Law, 2015). The system's time-varying features are highly sensitive to local

damages, especially in the response when the vehicle passes over the damaged area (Bu et al., 2006).

In this study, time-dependent parameters derived from the Vehicle-Bridge Interaction (VBI) model are used to train a deep learning (DL) model. However, a discrepancy exists between the numerical model, used to generate synthetic training data, and the real-world structure, which produces actual dynamic responses. To address this mismatch, transfer learning (TL) is applied. TL is a machine learning approach that leverages knowledge learnt from the data in a source domain to elevate the performance of a task in a target domain by exploiting the underlying similarities between the two. (Pan & Yang, 2010). Parameter-based TL approaches focus on fine-tuning a selected portion of model parameters that were pre-trained on the source domain. Conversely, feature-based methods, such as Domain Adaptation and Domain Generalisation, aim to align and transfer the source domain features and the target domain features within a common representation space. In particular, domain Adaptation seeks to identify a latent feature space where damage-sensitive patterns from both domains are well aligned. Meanwhile, Domain Generalisation focuses on learning features that remain consistent and discriminative across multiple source domains, without relying on any prior knowledge of the target domain. (Poole et al., 2022).

1.2 Aim and Objectives

This project aims to develop a novel and practical condition assessment approach for bridge structures using the VBI data and to reduce errors resulting from deviations between the numerical model and the real bridge structure.

This research proposes to meet the following objectives:

1. To develop a data-driven condition assessment approach for bridges from the VBI data using DL.
2. To develop an approach for bridge condition assessment using TL technique.
3. Numerical and experimental verification of the proposed approach.

1.3 Research Significance

The current thesis includes several significant contributions to the field, including:

- Introducing data-driven DL frameworks utilising vehicle bridge interaction responses.
- Employing time-frequency features for optimal condition assessment of the bridges, including damage localisation and prestressed force identification.
- Proposing a hybrid domain adaptation approach that remains robust against modelling errors in VBI systems under conditions of data scarcity in the target domain.
- Developing a domain-generalised autoencoder that reconstructs bridge responses and crack signature function without requiring any target-domain labels.

As a result, this project will bring structural health monitoring closer to practical and efficient application.

1.4 Structure of the Thesis

This thesis comprises seven chapters, as delineated below:

Chapter 1 - Introduction presents the background, research motivations, aim and objectives, and an outline of the thesis structure.

Chapter 2 - Literature Review reviews the latest research on vibration-based bridge condition assessment, focusing on AI-driven methods, deep learning, and transfer learning, and identifies key research gaps.

Chapter 3 - Structural Parameter Identification using Fine-tuning Technique proposes a fine-tuning approach employing convolutional neural networks (CNNs) for structural parameter identification from log-normalised Frequency Response Function spectra of an FE bridge model.

Chapter 4 - Bridge Damage Detection Leveraging Time-Frequency Features applies pretrained CNNs and classification algorithms to continuous-wavelet-transform images of VBI accelerations for damage location classification. Evaluates the effects of noise, sensor position, vehicle speed, and surface roughness on damage localisation accuracy.

Chapter 5 - Hybrid Domain Adaptation for Estimation of Prestressed Forces introduces a hybrid domain adaptation approach for estimating prestressed forces (PF) in

PSC bridges under different simulated domain shifts, in conditions of label scarcity in the target domain.

Chapter 6 - Domain-Generalised Autoencoder for Damage Identification proposes a domain-generalised CNN+LSTM autoencoder architecture that utilises passing vehicle acceleration data to reconstruct bridge acceleration responses, as well as crack severity and shape functions, without requiring any target-domain labels.

Chapter 7 - Conclusions and Recommendations summarises key research outcomes and contributions, and outlines recommendations for future work.

A summary of the thesis flow is provided in the flowchart depicted in Figure 1.1.

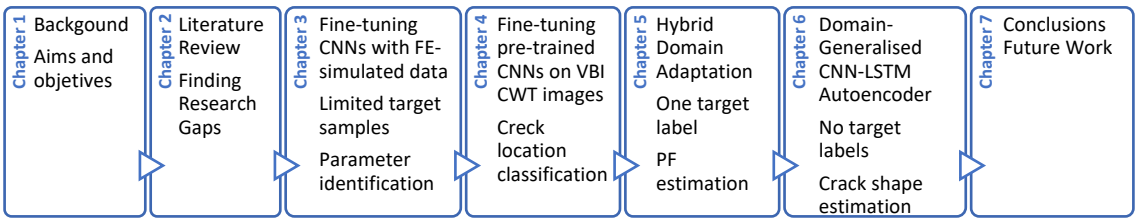


Figure 1.1 Flowchart of the thesis structure

CHAPTER 2 - LITERATURE REVIEW

2.1 Overview

This chapter explores the evolution of structural health monitoring (SHM) for bridges, tracing its development from traditional visual inspections to vibration-based approaches and, more recently, to methods powered by artificial intelligence (AI). Visual inspections, while widely used, are time-consuming, labour-intensive, and restricted to detecting only surface-level or visible damage. These limitations have driven the adoption of vibration-based techniques, which are capable of identifying internal or less obvious forms of deterioration. Changes in structural parameters such as stiffness and mass influence the dynamic response of the bridge, and these variations can be directly linked to its health condition.

Model-based approaches detect damage by comparing measured dynamic responses with a calibrated finite element (FE) model, typically requiring the solution of an inverse problem. However, the need for continuous model updating makes these methods impractical for large and complex bridge structures. In contrast, data-driven techniques rely on pattern recognition and extract damage-sensitive features directly from measured dynamic responses.

Data-driven damage detection can follow two main strategies: parametric methods, which utilise modal parameters (e.g., natural frequencies and mode shapes), and non-parametric methods, which operate directly on time, frequency, or time-frequency signals. Vehicle-bridge interaction (VBI) responses offer enhanced local sensitivity, as the moving vehicle simultaneously serves as both exciter and sensor. The time-varying nature of VBI signals increases their responsiveness to localised damage.

Machine learning tools can map the damage-sensitive features to structural condition labels. Deep learning eliminates the need for manual feature design: convolutional neural networks can be employed to analyse spectra or VBI accelerations, and recurrent neural networks can handle full-time series.

The serious challenge in DL approaches is label scarcity. Labelled damaged data is not available in real scenarios. Therefore, comprehensive labelled data sets can be created using simulations (e.g. FE models) to train the DL models and apply transfer learning (TL) to real structures. Fine-tuning uses pretrained network weights to customise models in situations where only a small amount of labelled data per target class is available. Conversely, domain adaptation focuses on aligning the source and target feature spaces, particularly when labelled data corresponding to the target domain is scarce or unavailable.

The review identifies a number of research gaps: use of global modal parameters instead of the acceleration responses, requirement for a large number of sensors installed on the bridge, domain shift between simulation and the real bridge, and lack of available target domain data corresponding to different damage scenarios. Later chapters address these issues with hybrid TL strategies, resulting in a reliable damage localisation and prestress force estimation of the bridges under moving vehicles.

2.2 Structural Health Monitoring

Structures age like humans. They may suffer damage during their service lifetime. This damage could be caused by fatigue, corrosion, etc. The damage may be propagated and even cause a global failure of the structure. Therefore, monitoring the structural condition throughout its service life is essential. Structural health monitoring (SHM) involves observing the structure's behaviour to assess and maintain its integrity over time. Structural damage is defined as changes in structural geometry or material in a way that affect the structure’s performance, serviceability, reliability and service life-time (Avci et al., 2021). Some damages, like corrosion or fatigue, affect the structure gradually. These kinds of damage accumulate over time and, therefore, degrade the structural performance and reliability. However, some damage appears rapidly and may suddenly cause the structure to fail. For example, an earthquake or fire might lead to the rapid failure of structures.

Visual inspection can be addressed as the earliest method for structural health monitoring, but it has important disadvantages. First, it is a local method that is time-consuming for the inspection of a large-scale structure. Second, it needs human experts who are highly trained for the task. Third, many structural members are not exposed, and damage in such elements cannot be visually detected.

Vibration-based structural damage detection has become a widely adopted approach in SHM. It has attracted significant attention over recent decades due to its sensitivity to even minor structural changes, which are often reflected in the system's dynamic response. In this method, the vibrational behaviour of a structure serves as an indicator of its health condition. The origins of vibration-based damage detection date back to the 18th century, when railway workers employed a rudimentary yet effective technique, tapping train wheels with a hammer and interpreting changes in the resulting sound as signs of potential damage (Turner & Pretlove, 1991).

2.3 Vibration-Based Structural Damage Detection

Vibration-based techniques in damage detection evaluate the structural condition by measuring and analysing the structure's dynamic responses. These techniques do not rely on knowing possible damage locations in advance, making them ideal for comprehensive monitoring. They often incorporate Non-Destructive Testing and Evaluation (NDTE) approaches, such as Ultrasonic Testing (UT) or electromechanical impedance-based techniques, to identify and characterise damage without causing harm or necessitating closure to the structure. A limited number of sensors placed on strategic points can be adequate for an acceptable damage detection result. Many algorithms are presented to correlate the measured response of the structure to a labelled situation, such as damage or no damage. These labels could correspond to damage in a particular element so that a damage localisation procedure would be feasible. In the next step, the severity of damages can be measured by going down the hierarchy. The damage detection problem can be addressed as a hierarchy of four levels (Yan et al., 2007).

Level 1 (Detection): Identifying the presence of damage in the structure.

Level 2 (Localisation): Identifying the location of damage among the structural members.

Level 3 (Assessment): Identifying the severity of damages.

Level 4 (Prediction): investigating the safety of the structure, including its residual life.

Vibration-based damage detection methods have attracted significant attention in recent decades due to their effectiveness in monitoring structural integrity. These methods involve both hardware and software components. The hardware aspect includes data acquisition systems equipped with sensors, such as accelerometers, that capture the dynamic responses of the structure. This sensor data is then processed and analysed through computational techniques to identify potential signs of damage. The gathered

data (usually available in the form of raw signals) will then be processed by the software part using signal-processing techniques and pattern recognition methods. The dynamic responses of the structure are finally translated into its health state. Machine Learning (ML) techniques are the most pioneering software for doing this translation. ML is a science that extracts knowledge from data. It enables computers to learn from examples, not from algorithms and programs. Just like a human, the computer will learn to interpret the data within a dataset and find useful patterns in it. These patterns will help the machine in decision-making. In a structural health monitoring problem, the acquired data from the hardware (e.g. accelerations) can be fed into a trained ML algorithm, and the health state of the structure can be obtained as a result. ML techniques perform great in solving complicated problems whose solution cannot be easily programmed. They also perform robustly in dealing with noise-polluted data and environmental and operational effects on data (e.g. temperature) (Avci et al., 2021). A wide range of ML techniques have been explored for structural damage detection. Among the most prominent are k-Nearest Neighbours (kNN), Random Forests (RF), Support Vector Machines (SVM), Artificial Neural Networks (ANN), and more recently, Convolutional Neural Networks (CNN), which have shown strong performance in handling complex structural response data.

2.3.1 Model-Based and Data-Driven Approaches

In practice, SHM can be implemented using two main approaches: model-based and data-driven.

Model-based SHM methods highly depend on numerical simulations to identify structural damage by comparing measured responses with those predicted by a calibrated model. In contrast, data-driven approaches utilise measured data directly to train models capable of detecting damage patterns without the need for an explicit physical model.

2.3.1.1 Model-based damage detection approaches

Model-based approaches treat damage detection as a broad category of inverse problems with uncertainties. They require building a very detailed model of the structure with original measurements from the field to update the model. The model is expected to reflect the dominant structural behaviour as in reality, with the least probable error. In-situ operational measurements from the structure can be compared to the model-generated data, and the differences between them can help detect and quantify damage. A damage detection algorithm can be employed to estimate the location, severity and properties of

damage by analysing the deviation of measured features from those extracted from responses, utilising techniques such as linear algebra or optimisation theory. (Sarmadi et al., 2022). Such an accurate numerical model is complex to build, and it needs to be updated using model-updating methods. Model-updating methods aim to regulate the physical parameters of the model (e.g. stiffness, mass, damping, etc.) to make the model outputs identical to the real structure's. These methods are not practical for large-scale bridges. Generating a detailed model with all the complexities of the members and connections of a real structure with the exact same behaviour may be achievable in small-scale structures or simple experimental setups, but surely is not applicable for most of the civil engineering structures. Sensitivity-based structural damage detection methods are great examples for model-based approaches where the changes in the responses and behaviour of the structure (e.g. static deformation or modal parameters) are related to the changes of the mass and stiffness using a straightforward equation (Shadan et al., 2016). The accuracy of this equation and the strength of the method strongly depend on the model's accuracy.

This approach to SHM is not the subject of this chapter and is only presented to avoid confusion between model-based approaches that rely on updated finite-element models to solve an inverse problem of damage detection and data-driven approaches that mainly employ simulated models to create a comprehensive database of damage scenarios for pattern recognition.

2.3.1.2 Data-driven damage detection approaches

Data-driven approaches define structural damage detection as a pattern recognition problem. As the name of this approach implies, they need data for training. Acquisition of useful data which are labelled and organised for training was a complicated and even impossible task a decade ago. With the rapid development of computer technologies, these data can be simulated with much less effort, and advances in AI and DL make it possible to use this huge amount of data and large datasets for data-driven methods. After training, the trained model can be used as a black-box to receive the data as input and return the state of the desired structure as output (Sen & Nagarajaiah, 2018).

Compared to the model-based approach, in a data-driven approach, the data are used to form a model. This model does not need to replicate the system's physical parameters while it aims to find the best patterns among the data and correlate these patterns to the true behaviour and conditions of the structure. In damage detection problems, damage-

related patterns will be utilised for this purpose. In other words, data-driven damage detection methods rely on feature vectors in the data corresponding to the health states of the structure. The source of data can be acquired from the real structure (mostly the data corresponding to the intact state of the structure), from small-scale prototypes (which can simulate multiple damage scenarios), or finite element (FE) models (which can generate a big dataset of different damage scenarios). FE-models are utilised here to make a dataset of features correlated to different damage scenarios. These features should be resilient to damage and less sensitive to environmental uncertainties, operational conditions, and modelling errors. Here, model updating is not the goal, and a good, but not necessarily identical, model should be able to generate the desired dataset. However, the more accurate and detailed the model is, the better performance can be expected for damage detection. The described dataset is fed into an AI algorithm for training. It can be noted that Transfer Learning (TL) can be utilised to fine-tune an AI model trained with features extracted from the FE model with the features extracted from the real structure. Interestingly, TL enables the use of a trained AI algorithm for a specific problem to improve the solution of a totally different problem.

2.3.2 Damage-Sensitive Parameters

Vibration-based structural health monitoring relies on changes in structural parameters that occur due to damage. Modal parameters are particularly informative, as they encapsulate key characteristics of the structure's dynamic behaviour. Alterations in structural components, boundary conditions, or connections can lead to measurable changes in these parameters. Damage detection methods that utilise modal parameters are referred to as parametric algorithms.

Parametric vibration-based damage detection algorithms use modal features (e.g., natural frequencies) as input. These features are assumed to be sensitive to structural damage, making them suitable indicators for assessing the structural condition. furthermore, the occurrence of damage within the structure will result in alterations to the system's dynamic characteristics. (Avci et al., 2021). These features are manually extracted from the structural response to the excitation. For example, the peak picking method aims to find the peaks of a frequency response function (FRF), which are meant to occur at natural frequencies. This section reviews key research studies on structural damage detection that use supervised machine learning with parametric features.

Mehrjoo et al. (2008) introduced a damage detection method based on Artificial Neural Networks (ANNs), using the first few bending modes as input features. A multi-layer perceptron with one hidden layer comprising fifty neurons was trained to localise damage. The method was tested and validated on both a simple truss bridge and a finite element model of the Louisville Bridge. The damage scenarios were simulated by introducing a reduction in the stiffness of the selected structural members. A dataset is generated for all the desired damage scenarios, and by selecting a division of the training dataset among all the generated data, the training process of networks is done after 75000 epochs using the BP algorithm. This number of epochs is claimed to be the optimal number, while selecting a larger number of epochs, the generality of the method will decrease due to over-training. As a result, the unknown damages were identified with good accuracy in location and extent. Gonzalez and Zapico (2008) proposed the employment of an ensemble ANN classifier containing two different ANNs for a better damage detection efficiency. The method was verified by a finite element model of a steel moment-frame as a representation of a 5-story building. Modal parameters of the first 3 modes were selected as damage-sensitive features, and the model output was the modal stiffness and mass of the system members after an earthquake. However, the method performed with 95% confidence in identification of mass and stiffness changes in structural members, it is noted that it was also sensitive to input parameter errors (e.g. the natural frequencies should be measured with less than 0.1% error and the mode shapes needs to have less than 0.02% error). Betti et al. (2015) utilised ANNs and genetic algorithms together for damage detection of a benchmark SHM structure. Training data was generated from an FE model. Damage is simulated by a cut in column flanges. An ANN with two hidden layers, along with a genetic algorithm (GA), was trained for damage classification. The input of the model was modal characteristics of the structure (frequency-dependent indexes referring to the first four modes). The modal assurance criterion (MAC) is utilised to define the fitness function for GA. The proposed method's performance revealed an acceptable accuracy in damage detection of the benchmark structure. Cury and Cremona (2012) compared the performance of three popular classifiers (Artificial Neural Networks, support vector machine, and Bayesian Decision Trees (it should be noted that BDT is not an ML approach)) in their research. They introduced and used symbolic data analysis (SDA) to transform raw acceleration data and modal parameters into a new data representation called symbolic data. Several numerical and experimental models were utilised to verify the method, including a FE model of a simply supported beam, and a

real-scale structure of a rail bridge. It was reported that both SVM and ANN classifiers were 100% successful, and the BDT method was only 90% successful. Talaei et al. (2018) developed a damage detection approach that utilises natural frequencies and mode shapes as damage-sensitive features, combined with a twin Gaussian process (TGP) classifier. A dataset was generated by simulating various damage scenarios on a reinforced concrete bridge modelled in SAP2000, from which the first six natural frequencies and corresponding mode shapes were extracted. Thirteen measurement points were strategically placed along the bridge deck to capture bending mode shapes with high spatial resolution. The sensitivity of these features to different damage cases was analysed, and the Modal Assurance Criterion (MAC) was employed to assess the impact of damage on mode shapes. The proposed method demonstrated reliable performance in identifying both the location and severity of damage, even when the measured data contained 5% white Gaussian noise.

On the other hand, "non-parametric algorithms" utilise signal-based features rather than parametric features (e.g., modal parameters). A feature extraction method needs to be employed to extract damage-sensitive features using signal processing methods. Signal-based features can be categorised into time-domain, frequency-domain, and time-frequency domain features.

Time domain features are the features derived from the time history of the structure's dynamic response, such as Auto-Regressive (AR) and Cumulative Absolute Velocity (CAV) models. Figueiredo et al. (2011) presented an AR modelling method for damage detection of a three-story laboratory frame structure. The structure was excited randomly by a shaker. Seventeen different damage scenarios, each containing changes in mass and stiffness, were introduced as breathing cracks on the structural elements. The coefficients of AR models were extracted and used as damage-sensitive features for classification. These features were employed to train an Auto-Associative Neural Network (AANN) consisting of three hidden layers—namely, the bottleneck, mapping, and de-mapping layers. For comparative analysis, three additional classification techniques—Singular Value Decomposition (SVD), Feature Analysis (FA), and Mahalanobis Squared Distance (MSD)—were also applied. All four methods demonstrated the capability to identify structural damage under varying operational and environmental conditions. However, the AANN approach outperformed the others, as it more effectively captured the nonlinear relationships among the extracted features, leading to improved classification accuracy.

Frequency domain features are derived from structural response data that has been transformed into the frequency domain through preprocessing. Commonly used frequency domain representations include Frequency Response Functions (FRFs), Transmissibility Functions (TFs), and Power Spectral Density (PSD). The FRF matrix, demarcated as the ratio of the structural response to the applied excitation force in the frequency domain, provides valuable insights into the system's dynamic behaviour, capturing natural frequencies, mode shapes, and other modal characteristics. In this matrix, each row corresponds to a measurement degree of freedom, while each column represents an excitation degree of freedom. Transmissibility Functions (TFs), which are derived from FRFs, also carry rich dynamic information but have the advantage of being independent of the excitation force. (Liu et al., 2020).

Time-frequency domain features are the representation of the data in the time-frequency domain, which can be extracted using a variety of methods such as Short-Time Fourier Transform and Wavelet Transform (Boonyakitanont et al., 2020). These kinds of features are more damage-sensitive and provide more information from the signal data compared to frequency domain or time domain features. The Hilbert-Huang Transform (HHT) is a time-frequency signal decomposition that uses empirical mode decomposition (EMD) and Hilbert spectral (Mandic et al., 2013) for feature extraction. The Teager-Huang Transform (THT) is another time-frequency feature extraction method by Cexus et al. (2010). This approach employs empirical mode decomposition. (EMD) and the Teager energy operator. Pan et al. (2018) employed these two methods in conjunction with the wavelet transform for feature extraction for damage detection of a simulated model cable-stayed bridge.

A dimension reduction process could be applied to these features to reduce the number of variables and improve the efficiency of pattern recognition. A pioneer linear dimension reduction method is principal component analysis (PCA), which helps to reduce the sensitivity of the extracted features to operational and measurement errors, along with reducing the dimension of the features.

2.4 AI-Driven Structural Damage Detection

Artificial intelligence (AI) is an outstanding topic in computer science that aims to enable machines to solve problems and perform tasks like human intelligence. In fact, there are a variety of problems that can be defined by rules for the computers and are expected to be solved by means of human-designed algorithms using if and else statements (Avci et al., 2021). However, not all the tasks can be coded into simplified statements. These tasks are the more challenging ones, such as recognising the faces, the voices, or tasks that are done by humans without much effort. These tasks can be learned by machines using Machine Learning (ML) algorithms based on analysing the observed big data in an efficient manner (Sun et al., 2020). Figure 2.1 shows the specialisation of AI algorithms, including ML. ML is an application of artificial intelligence that enables the machine to perform a task based on experience and not explicit programming. For example, imagine that the health state of a structure needs to be determined. The machine can assess the informative data measured from the structure using a model, which is trained on a dataset of different structural health conditions. This dataset consists of response data features, each referring to a particular health state (e.g. damage in a particular member with a specific extent). To be fed into the ML algorithm, the damage-sensitive features should be extracted from the raw data. It may not be feasible to manually find an explicit correlation between these features and the damage information due to the complex nature of the problem. During training, the machine automatically learns the relationship between input features and outputs, and then it can be utilised for the diagnosis of any new condition of the structure based on the obtained data.

ML tasks can be divided into four main groups:

1. Classification: This task aims to identify the category to which the inputs belong, such as whether any particular member of the structure shows detectable damage.
2. Regression: The output of this task is a numerical estimation, such as the prestressed force in a PSC bridge.
3. Prediction: The objective of a prediction task is to forecast the next step in a time series, such as predicting the value of Bitcoin over the next week based on historical price data.
4. Clustering: Aims to divide the data among separate groups that have similar properties.

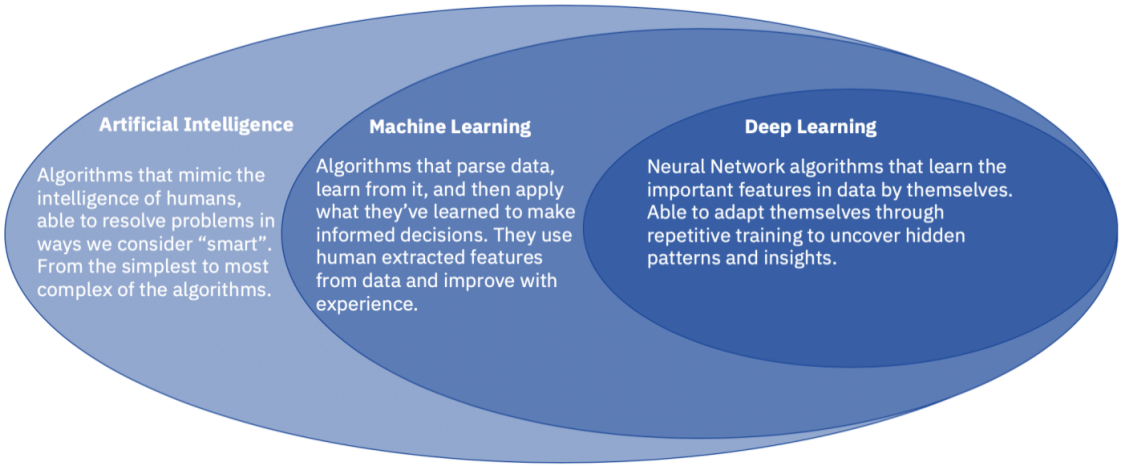


Figure 2.1 Specialisation of AI algorithms[1]

It can be noted that the clustering task has a main difference from the first three. It does not need any labelled data for learning and merely uses the unlabelled dataset to find the similarities between the features and divide them into separate clusters. This reveals the term "unsupervised learning", which is the opposite of "supervised learning". These two concepts will be discussed later in this chapter**.** Data-driven SHM approaches are based on statistical learning models. These learning models could be divided into two main subgroups: "supervised" or "unsupervised".

2.4.1 Unsupervised Structural Damage Detection

Unsupervised learning merely requires data from the intact state of the structure for its training process. During the training process, a statistical model based on the damage-sensitive features will be generated. During the monitoring process, the acquired data from the structure will be compared with this model, and any deviation from the undamaged state can be detected (Fugate et al., 2000). The learning methods for unsupervised learning can be categorised into novelty detection methods and clustering methods.

2.4.1.1 Novelty Detection Methods

Novelty detection methods are defined to compare the data from the unknown state of the structure to the model of the intact structure. Some of the most popular techniques for novelty detection are: Mahalanobis squared distance-based outlier analysis (Worden et al., 2000), and auto-associative neural networks (AANN) (Sohn et al., 2002). Self-

[1] Figure from: https://www.ibm.com/blogs/systems/ai-machine-learning-and-deep-learning-whats-the-difference/

organising neural networks (Yeung & Smith, 2005), and deep restricted Boltzmann machine (Rafiei & Adeli, 2018). Worden and Manson (2003) have evaluated the three popular approaches for unsupervised learning (MSD-based outlier detection, KDE and AANN) to perform a novelty detection task on an aluminium aircraft wing-box. The employed damage-sensitive features were extracted from the spectral lines of the transmissibility frequency response functions (FRFs) near the picks. PCA was applied as a dimension reduction tool. Ultimately, a ten-dimensional feature set derived through principal component analysis (PCA) was presented as damage-sensitive features. In the context of outlier analysis, the damage threshold was established via a Monte Carlo procedure. Regarding the neural network, white Gaussian noise was incorporated into the training data to evaluate the network's generalisation capabilities. The authors indicate that all three methods are effective in novelty detection. However, the outlier analysis combined with ANN demonstrated the highest sensitivity, detecting damage from a 75mm saw-cut.

2.4.1.2 Clustering Algorithms

Clustering algorithms aim to categorise the data into different clusters. Each cluster could be referring to a different damage state in the structure. Some of the most popular techniques for clustering are: k-means clustering method (Sohn et al., 2008), Gaussian mixture models (GMM) (Santos et al., 2017), and density peaks clustering algorithm (Cha & Wang, 2018).

Silva et al. (2016) introduced a clustering method using genetic algorithms. A hypersphere algorithm is employed to find the optimum number of clusters. Two different bridge structures (The Z-24 Bridge and the Tamar Bridge) were utilised for validation. The method was also compared to MSD-based outlier investigation and proved its superior performance in clustering.

2.4.2 Supervised Structural Damage Detection

Supervised learning algorithms require data from both intact and damaged conditions of the structure for training (Figueiredo et al., 2011). The data corresponding to different damage conditions should be labelled properly. A labelled dataset containing damage-sensitive features and corresponding labels (location and severity) is mandatory for supervised damage detection. The machine will learn to correlate between the damage-sensitive features and the health state of the structure through a training process.

The training data can be extracted from the structure but there are some limitations while a little amount of damage scenarios can be simulated on a real structure (for example by adding lumped masses on different locations). A comprehensive dataset can be acquired by means of numerical simulations. However, an accurate yet simple model would be more than enough to generate the labelled dataset.

A variety of classification methods have been proposed for ML-based supervised damage detection, some of the most popular classifiers are artificial neural networks (de Lautour & Omenzetter, 2009), support vector machines (Bornn et al., 2009; Chong et al., 2014; HoThu & Mita, 2013), decision trees (Mariniello et al., 2021), random forest (Zhou et al., 2013), Gaussian process (Talaei et al., 2018), Sparse representation (Chen et al., 2018) and k-nearest neighbour classifier (Vitola et al., 2017).

2.4.2.1 Artificial Neural Network (ANN)

An ANN is a parallel data processing approach that is formed based on biological neurons. An ANN consists of multiple artificial neurons that are placed in fully connected layers. An ANN is composed of an input layer, one or multiple hidden layers, and an output layer. The Input of each layer is the output of the previous layer. Each neuron performs by means of its weight, bias, and activation function. Activation functions are the core concept of ANNs, which make it possible to simulate nonlinear behaviour (Adhikary & Mutsuyoshi, 2004). An ANN without activation functions is nothing more than a combination of linear regressions and thus a linear regression itself. The multilayer perceptron (MLP), trained using the back propagation (BP) algorithm, is the most widely used ANN model among all existing models. (Crivelli et al., 2014). BP algorithm is a method applied for the training process, in which the deviation of the ANN output from the desired known output is calculated using a loss function and The weights and biases of the neurons are modified through gradient descent to minimise the system's loss. system.

2.4.2.2 Support Vector Machine (SVM)

The Support Vector Machine (SVM) was introduced by Cortes and Vapnik (1995). SVM performs based on the statistical learning theory. This classifier is a multivariate statistical approach, which can be applied to two or more different classes by seeking the boundaries (also called hyper-planes) with the maximum margin (the maximum difference between the nearest data). In practice, SVM projects the data (support vectors) in a higher-dimensional space using a nonlinear transform (by means of kernel functions), a linear

boundary which could classify the data in the higher-dimensional space is a nonlinear boundary in the original space, as illustrated in Figure 2.2. This method enables SVM to classify the data in a nonlinear manner and gives it a stronger generalisation capability.

The Kernel function should be selected based on the application. This manual selection of the optimal kernel function is the most notable limitation of SVM in practice. Some of the most popular kernel functions are: Linear, Polynomial, Radial, and Sigmoid Functions (Patle et al., 2013).

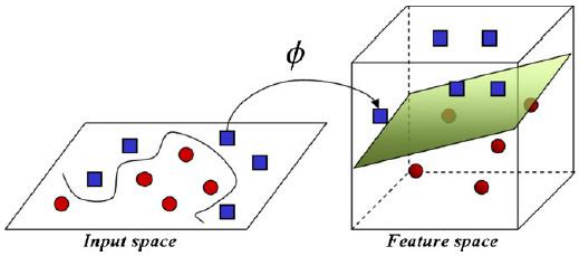


Figure 2.2 An illustration of the hyperplane for separation in the non-linear case [2]

2.4.2.3 K-Nearest Neighbours (KNN)

The kNN approach is one of the simplest methods for classification, which often yields competitive results. It has been used in statistical applications (Noi & Kappas, 2018; Weinberger & Saul, 2009). The basic theory behind kNN is that it searches among the dataset to find a group of k samples that are nearest to unknown samples (the distance can be defined by many metrics, such as Euclidean or Manhattan distance). The label of the unknown sample will be determined by the class attribute of the majority label of its k nearest neighbours (Akbulut et al., 2017; Wei et al., 2017). Figure 2.3 shows a typical example of k-NN algorithms. In the case where k=1, the method is also called nearest neighbour or lookup table (Khoshnoudian & Talaei, 2017).

[2] Figure is from: https://towardsdatascience.com/the-kernel-trick/

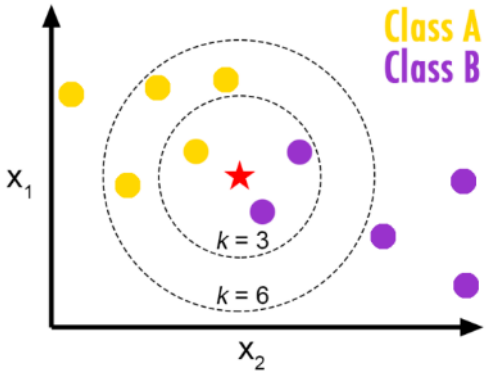


Figure 2.3 A schematic illustration of kNN algorithm with k = 3 and k = 6 [3]

2.4.2.4 Decision Trees and Random Forest

A decision tree serves as a decision support instrument, similar to a flowchart, wherein leaf nodes denote attribute assessments. The topmost node in the tree is called the root node. Decision trees are applicable for both classification and regression; however, their utilisation in classification tasks is highlighted more. The Random Forest methodology functions as an ensemble approach, providing outcomes based on the mean or majority voting across multiple decision trees. (Salkhordeh et al., 2021).

2.4.2.5 Gaussian Process (GP)

A Gaussian process is a stochastic method that maps data within a domain where any subset of the range exhibits a multivariate Gaussian distribution; thus, any linear combination of the projected data has a Gaussian distribution as well (Rasmussen, 2004). Gaussian process is described by its data mean and covariance function. Gaussian PP is an effective approach for modelling the non-linear relationship between inputs and outputs. (Mojaddady et al., 2011).

Twin Gaussian Process (TGP) represents an enhanced form of Gaussian Process methodology, applicable across diverse time series values. The fundamental principle of TGP is that it investigates not only the correlations among inputs but also those between outputs. Indeed, both inputs and outputs are modelled by distinct Gaussian Processes, each employing different kernel functions. Originally introduced by Bo et al. for the reconstruction of monocular three-dimensional human body poses, TGP demonstrated a substantial improvement, thereby establishing itself as a state-of-the-art approach. (Bo & Sminchisescu, 2010).

[3] Figure is from https://helloacm.com/a-short-introduction-to-k-nearest-neighbors-algorithm/

2.4.2.6 Sparse Representation Classification (SRC)

The concept of sparsity refers to the assumption that a data point can be expressed in a low-dimensional space utilising only a limited number of active elements (also called atoms). A sparse representation (SR) of a signal is a linear representation obtained by extracting a limited number of atoms. These atoms are derived from an overcomplete dictionary that encapsulates the vital information of the signal. SRC algorithms confer several advantages for classification purposes. Firstly, each signal is represented by a sparse representation, which enforces sparsity in the expansion coefficients. Secondly, sparse representation yields a robust depiction of signals in noisy environments. Thirdly, it provides generative components across a broad spectrum, thereby enabling an efficient and versatile signal representation. (Gu et al., 2021; Wang et al., 2018). Zolfaghari et al. (2014) proposed a couple of sparse coding (CSC) techniques for estimation of the human body pose in images, utilising a sparse coding algorithm. They determined that the CSC offers a more accurate representation in comparison to the basic SR.

2.4.2.7 Ensemble Methods

Wolpert (2001) They introduced the promising concept of 'no free lunch,' suggesting that finding an efficient classifier is not always possible in complex systems. He argued that combining multiple classifiers can overcome the limitations of using just one. An ensemble system typically includes several classifiers; all receive the input features, and their outputs are combined to make the optimal decision. The most common method for building an ensemble is majority voting (Hansen & Salamon, 1990), weighted averaging (Perrone & Cooper, 1992) and simple averaging approach (Opitz & Shavlik, 1996).

2.5 Deep Learning-Based Structural Damage Detection

It was previously discussed that conventional ML techniques utilise the extracted features from the raw data. In a damage detection problem, these features must be damage-sensitive and robust to operational and environmental uncertainties. Many feature extraction methods were discussed earlier, each with its own pros and cons. A weak feature extraction will lead to sub-optimal features, which can affect the classification performance. The most prevailing advantage of Deep Learning (DL) methods is that they can learn to extract optimal features automatically from raw data, leading to the best classification results. A DL algorithm consists of multiple processing layers to learn the representation of the input data during a multi-level abstraction. Thus, the feature extraction procedure is no longer a separate issue. It makes the DL algorithm suitable for

damage detection of structures based on the raw signal data (Krizhevsky et al., 2017). Two of the main DL approaches are deep Autoencoders (simply called Autoencoders) and deep convolutional neural networks (DCNNs).

2.5.1 Deep Autoencoders

Autoencoders were first introduced by Hinton and Salakhutdinov (2006). An autoencoder is a type of DNNs that contains multiple hidden layers. The activation function is usually selected to be ReLU as the activation function to address the gradient descent issues that can occur during training multiple layers. An Autoencoder is generally trained to reproduce the input at its output, learning to reconstruct the original data. The Encoder segment reduces the data's dimensionality through multiple layers, transforming it into a set of latent space features. The Decoder is subsequently employed to reconstruct the initial input based on these features. Autoencoders, at their core, are nonlinear tools for feature extraction, capturing important information from input data layer by layer through their hidden layers. They can be employed for feature extraction along with the classification task. Autoencoders retain the nonlinearity of data compared to traditional methods like PCA. In fact, PCA can be viewed as a special case of an Autoencoder with a single hidden layer, a linear activation function, and a basic mean squared error loss.

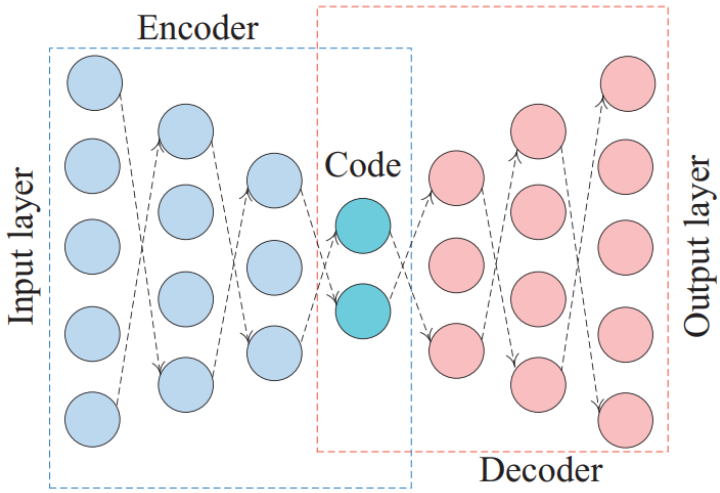


Figure 2.4 Deep Auto Encoders (Jiang et al., 2021)

Pathirage et al. (2018) introduced a structural damage detection method based on Autoencoders. However, they used modal parameters (natural frequencies and mode shapes) as inputs to the Autoencoder. This is not necessary and is far from the main purpose of an Autoencoder, which aims to extract the damage-sensitive parameters from raw data automatically. That's why the authors mentioned using FRF data as inputs in

their future work. A pre-training procedure is conducted for training the hidden layers of the Autoencoder, then a general optimisation of the whole network is done by fine-tuning. A numerical model of a steel frame structure, along with an experimental setup, was utilised for verification of the proposed method. The authors indicated that further study is needed, considering operational and environmental uncertainties. Jiang et al. (2021) proposed an unsupervised damage detection method based on extracting features from the signal data using Deep Autoencoders, as shown in Figure 2.4 The method has some limitations in detecting damage severities and can only reflect the deviation between the structure's current and healthy state.

2.5.2 Deep Convolutional Neural Network (DCNN)

Convolutional Neural Networks (CNNs), also referred to as Deep Convolutional Neural Networks (DCNNs), are among the most powerful tools in modern machine learning. A CNN is usually made up of several convolutional layers placed in a series order. Each convolutional layer is succeeded by a pooling layer. The primary goal of these layers is to extract task-relevant features from the inputs while reducing their dimensionality. Convolutional layers apply learnable filters (kernels) that slide over the input matrix to detect local patterns and features. Pooling layers, commonly max pooling, are then used to downsample the feature maps by summarising the information in local regions, which helps in reducing computational complexity and controlling overfitting. As a result, CNNs are capable of autonomously learning both low-level and high-level features from unprocessed input data (Shin et al., 2016). Fully connected layers are subsequently used to perform a classification or regression task on the extracted features. The final output of the CNN is produced through a Softmax activation function in the output layer. Yu et al. (2019) proposed a DCNN for damage detection in building structures equipped with Magnetorheological Elastomer (MRE) isolators under ambient conditions. Their approach directly utilises raw vibration signals from the structures, eliminating the need for manual feature extraction or signal denoising.

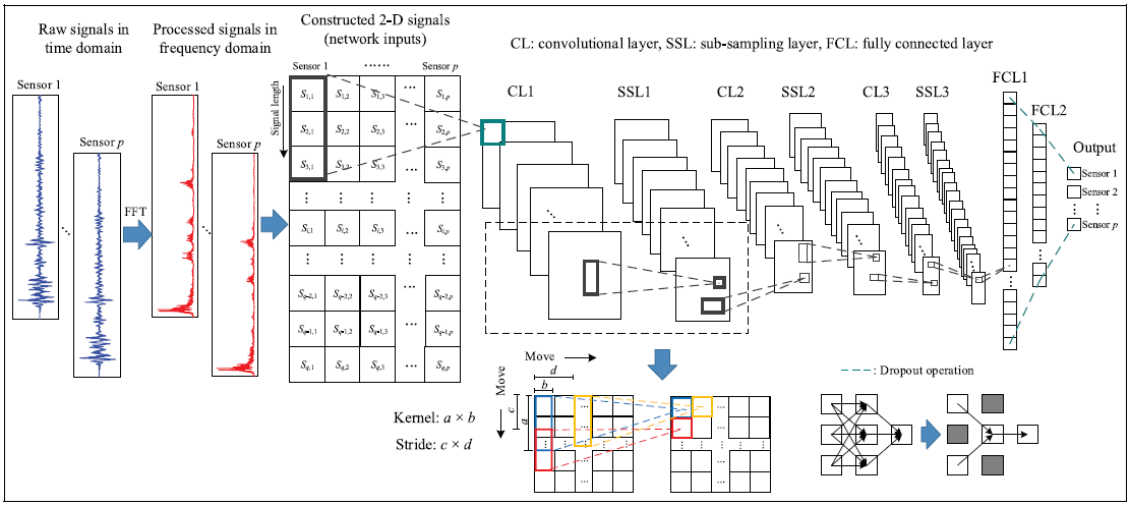


Figure 2.5 Structure of Wang et al. proposed DCNN-based DI model (Yu et al., 2019)

Figure 2.5 illustrates the structure of the DCNN model (Yu et al., 2019). The DCNN is used to process the signals acquired from the structure for damage detection purposes. Fourier transform (FFT) is deployed on raw signals as a data preprocessing procedure so that the input of DCNN would be in the frequency domain. The dimension of raw data will be reduced during this process. A sensor is installed at each story of the building; thus, the input matrix is an n×m matrix where m>>n (n is the number of accelerometers installed on different storie, and m denotes the signal length). For the first convolutional layer, a relatively big kernel is applied. Selecting a small kernel size is not suitable for the first convolutional layer. A small kernel will avoid noise effects and keep the valuable information of the signals. Next, a max-pooling kernel is used to lower the dimension of convoluted matrices by selecting the maximum number in each kernel while the kernel strides over the convoluted data. The kernels of the next convolutional layers are selected to be smaller. The proposed model comprises three convolutional layers, each succeeded by a max-pooling layer, and is finalised with two fully connected (FC) layers. The FC layers classify features extracted by the convolutional stages, with the final output produced via a Softmax activation function. To enhance training stability and reduce overfitting, batch normalisation and dropout are employed. The Leaky ReLU activation function is used, and the model is trained using a cross-entropy loss function.

The input excitation combines El Centro earthquake records (N-S component) with pseudo-random binary currents serving as control inputs to a magnetorheological elastomer (MRE) isolator. Nineteen damage scenarios, simulated through varying degrees of stiffness reduction, are used to train the deep convolutional neural network

(DCNN). The model's robustness is evaluated under three noise levels to account for uncertainties. Its performance is compared with General Regression Neural Networks (GRNNs) using multiple evaluation metrics, with results indicating superior accuracy.

2.5.3 Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is a specialised neural architecture tailored for handling sequential and time-series data. In contrast to standard feedforward networks, RNNs feature internal feedback loops, where the output at one time step is passed back into the network as input for the next. This structure enables the network to preserve information across time steps, making it well-suited for capturing temporal patterns and dependencies. One of the key strengths of RNNs lies in their ability to process input sequences of varying lengths, making them highly flexible for a range of time-dependent tasks (Hermans & Schrauwen, 2013). Lee et al. (2021) trained a DNN for damage detection of a floating structure. The damage scenarios were introduced on mooring lines, considering environmental underwater conditions. They trained the model using 740 damage scenarios. The model was trained and tested using numerical simulations without measurement noise. When using statistical parameters of the time series (e.g., mean and standard deviation) as input, the model's accuracy did not exceed 79%. However, by using the full time series data to train a RNN, the accuracy improved significantly, reaching up to 99%.

2.6 Transfer Learning

Transfer Learning (TL) is an artificial intelligence technique that facilitates the reclaim of knowledge gained from one task to improve learning on another, related task. It involves utilising information from a well-resourced source domain to support model training in a target domain, especially in cases where labelled data in the target domain is scarce. The concept is inspired by human learning; for instance, mastering how to ride a bicycle can facilitate learning to ride a motorcycle, due to shared underlying principles. TL is especially valuable when the source and target domains exhibit certain similarities in structure or representation (Pan & Yang, 2010). This concept becomes more interesting when it can be applied to structural damage detection problems. For example, an ML model could be trained for damage detection of an FE model. A positive transfer could make it possible to use the knowledge of this training process for damage detection of a real structure with limited available data. For better decision-making about using TL, different types of TL are explained below (Pan & Yang, 2010):

Inductive Transfer Learning:

Inductive TL is applicable when the tasks in the source and target domains differ, regardless of the similarity between the domains themselves. In this approach, a small number of labelled data is required from the target domain for training. The knowledge from the source domain helps the model learn the target task more efficiently. If labelled data is also available in the source domain, the transfer becomes more effective and informative.

Transductive Transfer Learning:

Transductive TL is used when the tasks remain the same across the two domains, but the domains themselves differ. In this scenario, the source domain must provide a rich set of labelled data, while the target domain has only a limited amount of unlabelled data. When the feature spaces are shared but the marginal distributions differ between domains, domain adaptation techniques are typically employed to align the two domains and enable successful knowledge transfer.

Unsupervised Transfer Learning:

Unsupervised TL applies when the source and target tasks are different but related, and no labelled data is available in either domain. In such situations, unsupervised learning

methods—such as clustering or dimensionality reduction—can be used to extract meaningful representations and facilitate knowledge transfer between tasks.

To conduct a successful TL, we need to answer three main questions: When, What and How to transfer.

2.6.1 When to Transfer?

It is important to know the exact problem we are dealing with and the related domains and tasks. If the source and target domain are irrelevant, then a TL will not help and even could reduce the performance of domain task which is called “negative transfer”. We should be completely aware of the source and target domains and tasks, and the similarities between them, before deciding to use TL.

2.6.2 What to Transfer?

For TL, there is another important question that needs to be answered: what to transfer? To answer this question, TL solutions are categorised into four main categories: instance-based, feature-based, parameter-based, and relational-based approaches, which will be described below.

2.6.2.1 Instance-Based TL Approaches

Assuming that the feature spaces of the two domains are the same but the marginal class distribution is different, instance-based methods can provide an effective solution. These techniques involve assigning weights to instances from the source domain data based on their similarity to those in the target domain. This process begins by evaluating the similarity between data samples across the two domains. Based on these similarities, instance weights are computed to adjust the source domain distribution, allowing it to better align with the target domain and thus improving model generalisation (Zhuang et al., 2021).

2.6.2.2 Relational-Based TL Approaches

These methods leverage the logical rules and structural relationships present in the source domain. These rules and relationships are then transferred to the target domain to support learning in the absence of extensive training data (Zhuang et al., 2021). Statistical relational learning methods can be employed to identify relationships between data within a domain and transfer this knowledge to another domain. (Mihalkova & Mooney, 2008)

2.6.2.3 Parameter-Based TL Approaches

Parameter-based transfer learning approaches utilise specific parameters from a machine learning model that is already trained on the source domain to enhance learning in target domain. For example, in DNNs, parameters such as weights and biases learned from the source task can serve as a valuable starting point when training a model for a different, yet related, task. This strategy helps accelerate convergence and improve performance, particularly when the target domain has limited labelled data (Pan & Yang, 2010).

2.6.2.4 Feature-Based TL Approaches

Feature-based transfer learning approaches aim to transform the features of the source and target domains so that they align effectively within a shared representation space. Domain adaptation techniques are commonly employed to achieve this alignment. These approaches can be broadly classified into two categories: asymmetric methods, which map features from the source domain to the target domain; and symmetric methods, which project features from both domains into a common latent space where their distributions are more closely matched (Tan et al., 2018).

2.6.3 How to Transfer?

This question should be answered based on the specific problem we are dealing with and the amount of labelled and unlabelled data in each domain. For instance, if we want to train a DNN but we do not have sufficient data. Parameter-based TL approaches, such as fine-tuning, can be used. If we want to transfer the knowledge between two domains without any labelled data in the target domain, domain adaptation techniques might be the solution. However, the exact definition of the problem and domains and awareness of the limitations of each TL method is a must. Here, two of the most popular TL methods that have been used in structural damage detection problems will be described:

2.6.3.1 Fine-Tuning

Fine-tuning is an inductive TL that aims to transfer the parameters of an ML model from a source domain to a target domain. Most ML algorithms require initial values for their training parameters (like the weights in DNNs). These values are usually selected randomly. Choosing the initial state of an ML algorithm makes the training process more efficient, requiring much fewer training data, and also prevents further issues such as anchoring in local minima (Willard et al., 2020).

In CNNs, a common TL strategy is pre-training the model utilising the source domain data and subsequently adjusting (fine-tuning) it using target domain data. During fine-tuning, the parameters of certain convolutional layers—typically the earlier ones—are frozen, as these layers capture general feature representations that are often transferable across tasks. This allows the model to retain useful feature extraction capabilities while adapting the later layers to the specific characteristics of the target domain. (Zhuang et al., 2021). An example of fine-tuning for vision-based structural damage detection is shown in Figure 2.6.

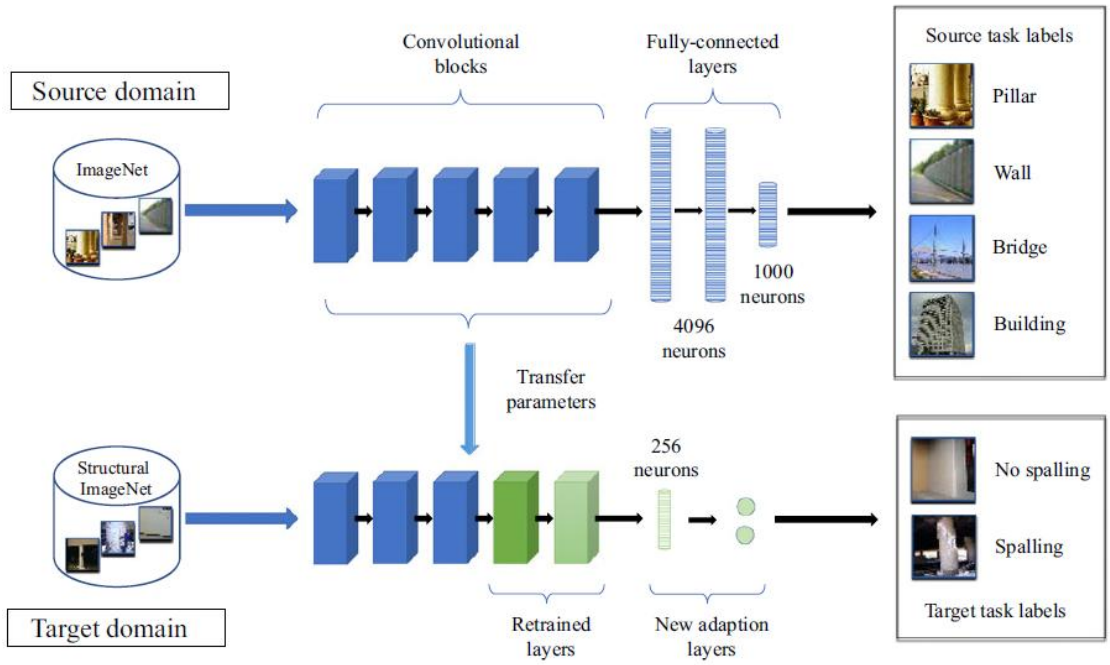


Figure 2.6 An example of fine-tuning for vision-based structural damage recognition (Gao & Mosalam, 2018)

As an example of a fine-tuning approach in vibration-based structural health monitoring, Azimi and Pekcan (2020) proposed a CNN-based method for damage detection in a benchmark building structure. Their model uses extremely compressed acceleration data—specifically, the mean, variance, and scale factor of power spectral density histograms—as input features. A CNN was first pre-trained on compressed data represented by discrete power density histograms and subsequently fine-tuned using the extremely compressed data. Given the similarity between the source and target domain data, all convolutional layers were frozen during fine-tuning, while only the fully connected layers were retrained. In essence, the flattened output from the convolutional layers served as the input to a newly trained FC network. The method was numerically validated using the IASC-ASCE four-story steel frame benchmark structure. Eight

damage scenarios were defined by removing the braces in different stories of the structure (with 2,4, and 8 removed braces). The excitations were defined as wind (white Gaussian noise) and a shaker (for one of the cases). A 10% N/S noise was added to the structural response. 16 sensors were distributed on the structure to capture its response, and 1000 data samples were generated for each damage scenario. A comparison between the accuracy of the models in pre-training and fine-tuning phases shows that the TL is performing great, and the three statistical parameters of the histogram data were more than enough to lead to a good prediction of structural health state, except in one of the damage scenarios. The method was also validated using experimental acceleration data from two benchmark structures, and TL proves to be robust and has an impressive performance with an accuracy of 91%. *However, it should be highlighted that the main drawback of using the fine-tuning approach in damage detection problems is its need for a labelled target for training the new FC layers. At least one labelled data point for each damage class is needed, while labelled data in the target domain is scarce and expensive in practice.*

2.6.3.2 Domain Adaptation

Domain Adaptation (DA) is a transductive transfer learning technique designed to transfer knowledge of feature representations between domains, as illustrated in Figure 2.7. The primary goal of DA is to align the source and target data within a common feature space by minimising the distributional differences between them, while maintaining the structural integrity of the original data. By achieving this alignment, a classifier or regressor that has proven to work effectively in the source domain can then be trained and applied in the adapted feature space. This enables the transfer of classification or regression knowledge to the target domain, allowing unlabelled target data to infer their labels based on the learned structure (Guan & Liu, 2021).

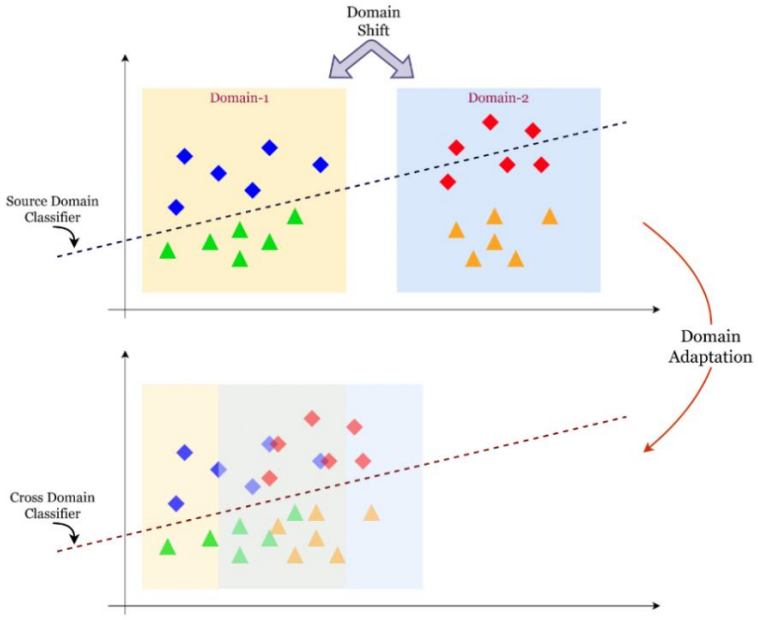


Figure 2.7 Domain Adaptation[4]

Domain adaptation (DA) techniques can be classified into three main categories according to the available labelled data (B. Li et al., 2021). 1) **Unsupervised DA:** The source domain is completely labelled, and the target domain is completely unlabelled. 2) **Semi-supervised DA:** The source domain is completely labelled, and the target domain includes both labelled and unlabelled data. 3) **Supervised DA:** Both source and target domains have fully labelled datasets.

Additionally, DA methods are also categorised based on the shared class labels in the domains (You et al., 2019), as illustrated in Figure 2.8:

1. Closed-set DA: Same set of classes in both domains, though their feature distributions differ.
2. Partial DA: The source domain includes all elements of the target domain, plus additional ones;
3. Open-set DA: There are one or some classes in the target which are absent in the source domain, posing additional challenges for classification.
4. Universal DA: The two domains may share some classes but also include private, domain-specific classes not present in the other domain.

[4] Image from: https://www.v7labs.com/blog/domain-adaptation-guide

These classifications provide a framework for selecting appropriate DA techniques depending on the characteristics and constraints of the application scenario.

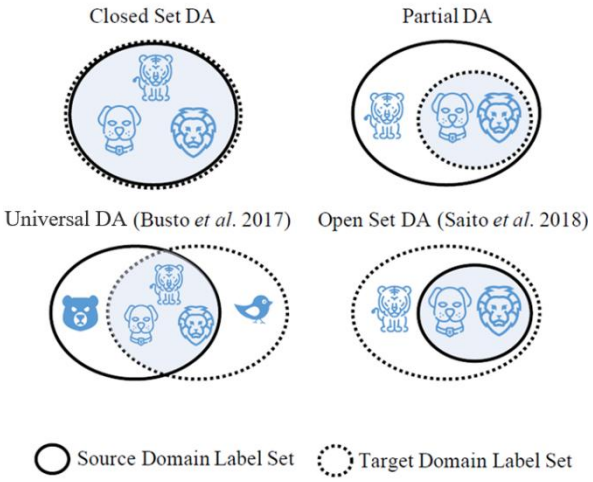


Figure 2.8 Domain adaptation categorisation based on the shared classes between domains (You et al., 2019)

2.6.3.3 Deep Domain Adaptation

The revolution of DL has a great influence on ML approaches since they can extract the features from raw data. But they need a massive labelled dataset to be trained. Thus, deep domain adaptation (DDA) approaches are introduced to tackle this problem by combining powerful DL methods and domain adaptation approaches. DDA techniques can be divided into three main categories:

Discrepancy-based: This strategy utilises a Deep Adaptation Network (DAN) to extract features that are transferable across domains, referred to as domain-invariant features. The approach aims to minimise the discrepancy between the source and target domains by reducing both marginal and conditional distribution differences. This is achieved through the optimisation of Maximum Mean Discrepancy (MMD) objective functions, which promote alignment of feature representations in a shared latent space.

Reconstruction-based: In this approach, an autoencoder is employed, as shown in Figure 2.9. The encoder is trained to extract representative features from the source domain data, while the decoder is designed to reconstruct samples from the target domain using these extracted features. This process encourages the model to learn a shared representation that captures commonalities between the two domains, thereby reducing discrepancies and promoting effective knowledge transfer.

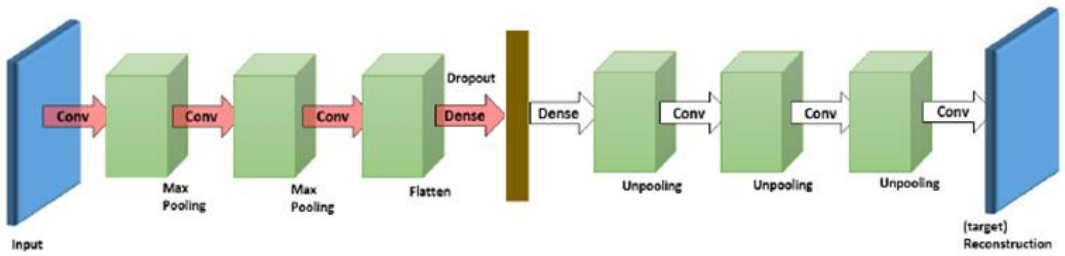


Figure 2.9 Deep Reconstruction-based Domain Adaptation (Ghifary et al., 2016)

Adversarial-based: The Domain-Adversarial Neural Network (DANN) similar in objective to discrepancy-based methods, is used to extract features with the lowest sensitivity to the domain shift, also called domain-invariant features. However, the cross-domain discrepancy will be minimised using an adversarial objective function (Ganin & Lempitsky, 2015), as shown in Figure 2.10. The original idea was inspired by Goodfellow et al. (2014), introducing Generative Adversarial Networks (GAN). This method uses a domain discriminator and a feature extractor simultaneously. The domain discriminator is aimed at distinguishing the domain of the input data, and a domain confusion loss is introduced to guide this process. Simultaneously, the feature extractor aims to learn discriminative features for the primary classification task while also deceiving the domain discriminator. In doing so, the model encourages the extracted features to be indistinguishable across domains, thereby achieving domain invariance. In other words, the feature extractor trains to minimise classification loss for labelled source samples and maximise the domain confusion loss for all of the samples at the same time (Jingyao Li et al., 2021).

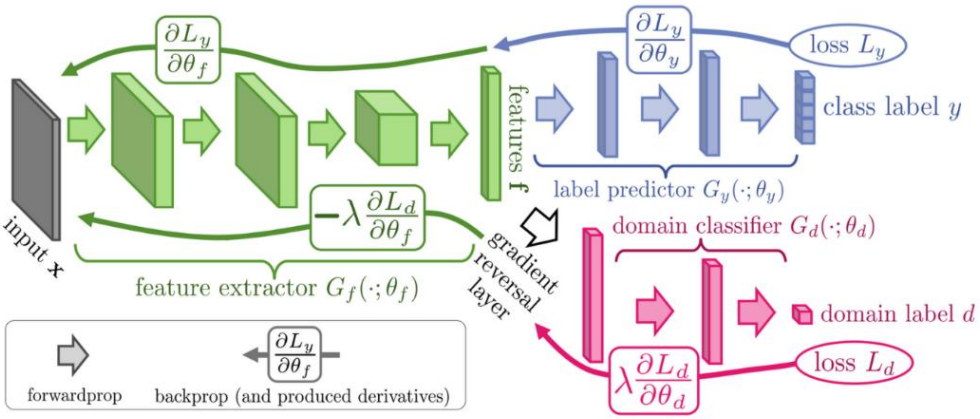


Figure 2.10 Adversarial deep domain adaptation (Ganin & Lempitsky, 2015)

2.6.4 Domain Adaptation-Based Structural Damage Detection

A good example of using the domain adaptation approach for damage detection is presented in the population-based structural health monitoring terminology (Zhuang et al., 2021). Population-based SHM approach is presented across a series of articles. (Bull, Gardner, Gosliga, et al., 2021; Gardner et al., 2021; Gosliga et al., 2021; Tsialiamanis et al., 2021)This approach assumes similar structures as members of a population and suggests that knowledge could be transferred between them. The method's importance becomes more highlighted when the data available in the target domain are unlabelled.

A **homogenous population** consists of nominally identical structures (with the same geometry, topology, and materials) that might have minor variations in the population, e.g., minor imperfections due to manufacturing processes, transport of the components, or differing operational environments.

A **heterogeneous population** consists of non-identical members with dissimilarities. It should be noted that if any two members of a population are heterogeneous, the whole population will be regarded as heterogeneous. This similarity could be defined as structural equivalence (which means two structures are topologically equivalent with the same boundary conditions). The three main sources of difference in the structures are listed as geometry, material and topology (Bull, Gardner, Gosliga, et al., 2021).

For a homogeneous population, if a particular useful knowledge for damage detection is available for a subset of members, this knowledge can be generalised to the whole population by defining the concept of form. One of the most important benefits of defining a form is its application in novelty detection. For example, in the test FRF data, the individual members of a turbine farm can be compared to the form of the population, and a novelty could be detected by defining a threshold for deviations. Feature-based TL has been employed in the third article of the series of articles of Foundations of population-based SHM for both homogeneous and heterogeneous populations. TL makes it possible to make general use of the knowledge of a particular member of the population. For example, existing damages on a turbine located in a wind farm can be used for damage detection of the other turbines. A single ML classification method can be trained for all the members. The proposed transfer learning (TL) approach is grounded in the concept of domain adaptation. It assumes that, by leveraging the labelled source domain data, the unlabelled target samples can be classified. This is achieved by mapping both domains into a shared latent space, where data points corresponding to the same class are aligned,

thereby facilitating knowledge transfer across domains (Figure 2.11). The unlabelled data can be labelled the same as the matched labelled data. This method can be used for heterogeneous populations having inconsistent features and even inconsistent label spaces. However, its efficiency depends on the similarity of the geometry of the data (Gardner et al., 2021).

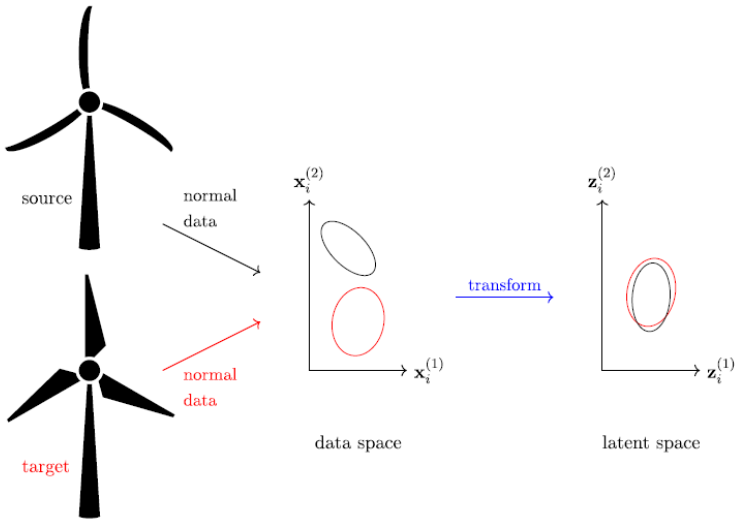


Figure 2.11 Domain adaptation between source and target (Bull, Gardner, Dervilis, et al., 2021)

Domain adaptation can still be effective even when the label spaces of the source and target domains differ. However, identifying domains with the same label space is typically sufficient to enable positive transfer and improve performance. (e.g. the damage labels of a 3-story building can be assumed to be the same as the damage labels of the first three levels of a 4-story building). However, some classes will remain unmatched and therefore unlabelled. In these cases, it is crucial to cluster the data within each domain prior to domain adaptation. Clusters referring to extra unknown labels should be removed from the mapping procedure. Finding the data referring to the extra labels is critical since removing the wrong data will lead to negative transfer.

In this context, transfer learning through feature mapping can be interpreted as a form of feature extraction. Joint Domain Adaptation (JDA) is employed to project features from both domains into a shared latent space, typically a Reproducing Kernel Hilbert Space (RKHS). JDA works by minimising the distance between corresponding clusters in the source and target domains, effectively aligning them within a common k-dimensional feature space. The resulting mapped features are considered to be damage-sensitive and representative of patterns shared across all domain instances. A classifier can then be

applied to this transformed space to assign labels to each cluster, enabling accurate classification in the target domain (Gardner et al., 2021).

2.6.5 Transfer the Knowledge from FE Models to Real Structures

Transferring knowledge from a finite element (FE) model to a real structure represents a large contribution to the field of SHM. In this context, the damage-sensitive features extracted from experimental data or real-world structures constitute the target domain, while features derived from the FE model form the source domain. However, due to the scarcity or even lack of labelled data in the real structure (target domain), traditional fine-tuning approaches are not feasible, making them less suitable for practical transfer learning applications in real-world scenarios. Gardner et al. (2021) introduced a joint domain adaptation for transferring the knowledge between the FE model and the experimental setup of a three-story building, considering that these two form a heterogeneous population. Five damage scenarios were simulated in the experimental setup by a cut in a single beam in the first story with different depths. The cut was modelled in the FE model as an open crack. The first three natural frequencies were used as the damage-sensitive features, as shown in Figure 2.12. However, the natural frequencies extracted from the FE model appeared to be half of those of the experimental setup. Feature mapping was applied using JDA, and the result showed the great performance of domain adaptation in mapping the data from the same clusters on top of each other, which is a sign of positive transfer.

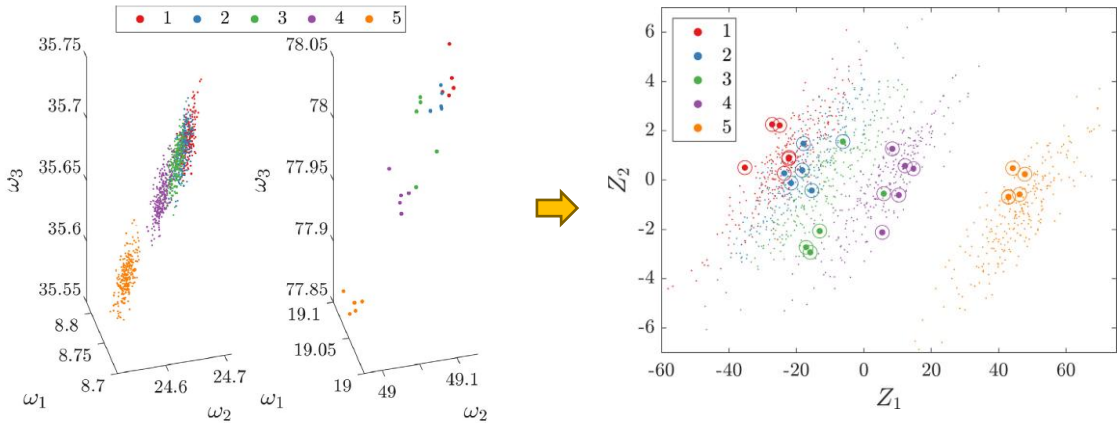


Figure 2.12 Domain adaptation between numerical model and real structure (Gardner et al., 2021)

They concluded that, based on their approach, a FE model can be used in labelling the unknown damage states in a real structure, and the latent space which maps the labelled FE-model data and unlabelled experimental data truly on top of each other can be generalised for mapping the features of an unknown damage state and finding its true label. *However, in this study, it is assumed that five unlabelled classes of data are extracted from the experimental setup, and the exact same damage classes are available and labelled in the FE model. This is not the kind of data we have in practice. In practice, we do not have a bunch of unlabelled data in the target domain corresponding to all labelled damage scenarios.*

Poole et al. (2022) introduced another domain adaptation framework for class imbalance. They proposed partial domain adaptation using static alignment in which target domain data contributes to fewer classes compared to the source domain data, as shown in Figure 2.13. This assumption is much more realistic since we have the unlabelled features of a damaged scenario and not all the damage scenarios. We also have the features of the healthy state of the structure, and maybe a couple of pre-defined damaged scenarios. In this study, it is assumed that all the features from the target domain (real structure) are unlabelled, and partial domain adaptation is used to find the unknown labels, including the label of the unknown damage scenario. *But it still has a major drawback. The partial domain adaptation technique needs to be conducted as a pre-processing procedure every time a new condition of the structure appears. It is time-consuming and does not allow for real-time monitoring of the structure.*

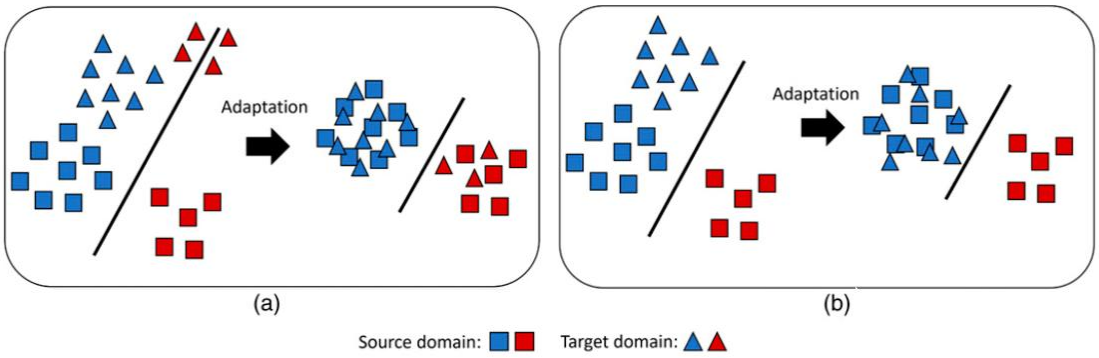


Figure 2.13 (a) closed set domain adaptation in (b) partial domain adaptation for damage detection (Poole et al., 2022)

2.7 Summary

This chapter has reviewed the evolution of AI-driven condition-assessment methods for bridge structures, from modal-parameter-based techniques to data-driven approaches that operate directly on raw time, frequency, or time-frequency responses. Recent studies show that non-parametric features are able to detect local damages more reliably than natural-frequency shifts or mode-shape curvatures. DL models such as CNNs can now learn these rich patterns without hand-crafted feature extraction.

One of the most important drawbacks of data-driven damage detection is the lack of available data from the real structure. Therefore, the training dataset is generated by an FE model. However, training the AI model with FE model data and testing it with real structural data is like training with apples and testing with oranges. Transfer learning (TL) techniques can be employed to eliminate the domain shift between finite-element (FE) simulations and real structures when there is limited measured, labelled data. Parameter-based TL (e.g., fine-tuning) is effective when a few target labels exist, whereas feature-based TL (e.g., domain adaptation) can align unlabelled target data with labelled source domains.

Despite the aforementioned advancements in the field, the following research gaps have been identified through a systematic survey of peer-reviewed SHM studies published in recent years:

- Real bridges do not offer labelled data for multiple damage scenarios; therefore, supervised data-driven approaches are not feasible in the real world. The real challenge is to transfer knowledge from one or more labelled source domains to a target domain that lacks any data (labelled or unlabelled) covering its damage scenarios. In practice, the only realistic data available from the target bridge typically corresponds to its healthy state.
- Even after model updating, FE outputs do not always match the real bridge responses exactly, and the domain shift still exists.
- Most existing TL works still rely on global modal parameters instead of the raw acceleration responses, which contain extensive information related to the bridge's health state.
- Most studies require a large number of sensors to be installed on the bridge to achieve reliable damage localisation or parameter estimation.

- Current TL methods cannot transfer knowledge effectively when the target bridge lacks labelled damage data or is completely unlabeled and unseen.

This thesis addresses the identified research gaps as follows:

- Develops a fine-tuning method that utilises FE-simulated source domain data with limited labelled data from the target domain for added-mass detection (Chapter 3), and proposes a hybrid domain-adaptation framework that only requires one label from the target domain (healthy state of the structure) to estimate the remaining prestressed forces (Chapter 5).
- Addresses the domain shift issue by employing transfer learning strategies including fine-tuning (Chapter 3), domain adaptation (Chapter 5), and domain generalisation (Chapter 6).
- Replaces conventional modal-based inputs with log-normalised Frequency Response Function (FRF) spectra for parameter identification (Chapter 3), Continuous Wavelet Transform (CWT) images derived from VBI accelerations (Chapters 4 and 5), and raw VBI acceleration data (Chapter 6).
- Demonstrates accurate damage localisation using just a single accelerometer, robustly accounting for realistic noise conditions, varying vehicle speeds, and surface roughness (Chapter 4).
- Develops a domain-generalised CNN+LSTM autoencoder for reconstruction of bridge responses and crack severity, location and shape estimation without requiring any labelled data from the target domain. (Chapter 6).

CHAPTER 3 - STRUCTURAL PARAMETER IDENTIFICATION FOR BRIDGE STRUCTURES USING FINE-TUNING TECHNIQUE AND FREQUENCY RESPONSE FUNCTIONS

3.1 Overview

This chapter outlines a transfer-learning methodology for structural parameter identification in reinforced concrete (RC) bridges using convolutional neural networks (CNNs). A CNN initially trained on a baseline finite-element model (FEM) is fine-tuned with a small set of target-domain data so that it can identify structural parameter changes even when only limited target-domain measurements are available. Adding or removing lumped mass provides a direct, measurable variation of the system matrices. Detecting such variations is critical for structural-health monitoring (SHM), digital-twin calibration, and informed asset management.

This chapter first summarises the CNN-based SHM workflow and the transfer-learning strategy, and then specifies the network architecture and fine-tuning procedure. An FE bridge model is employed to generate frequency-response-function (FRF) data that populate the source-domain training set. The FRF data encompasses both modal and non-modal dynamic characteristics of the structure, allowing for clear differentiation of changes in the structural parameters.

The performance of multiple CNN configurations is compared. Robustness is evaluated by introducing white noise into the vibration signals. The domain-shift effects are investigated by adapting the source-trained network to a target FE model with slightly altered material properties, representing discrepancies between simulation and reality.

Results confirm that the fine-tuned CNN reliably detects added-mass scenarios across various noise levels and maintains high classification accuracy, even with limited target-domain data. Despite the limitations in the proposed approach, the fine-tuned CNN offers a data-efficient tool for parameter identification in bridge structures.

3.2 Proposed Methodology

The overall methodology used in this chapter is explained in this section with a focus on Deep Learning Framework, Convolutional Neural Networks (CNNs), their components and structure, and inductive TL (i.e., Fine-tuning) and its application in transferring the knowledge between DL models.

3.2.1 Deep Learning Framework for Bridge Condition Assessment

A DL framework will be established for added mass detection of the bridge. The inputs of the DL model would be the dynamic responses of the bridge at different sensor locations, and the output of this model will be the location of the added mass on the bridge slab. The framework is described as follows:

1. A FEM of the bridge structure is established based on the VBI system.
2. Different added mass scenarios are introduced to the finite element model, and the vibrational responses of the bridge corresponding to each scenario are calculated.
3. The acceleration response data in the time domain are converted into the frequency domain through the application of a Fast Fourier Transform (FFT).
4. To enhance the model's robustness, the FRF data are augmented by adding different percentages of white noise to the acceleration data.
5. A labelled dataset of each added mass scenario and corresponding responses is generated.
6. A DL model (e.g. CNN) is trained using the labelled dataset to extract the sensitive features from the vibrational data and to make a correlation between the extracted features and the label of the added mass scenario.

The framework is illustrated in Figure 3.1.

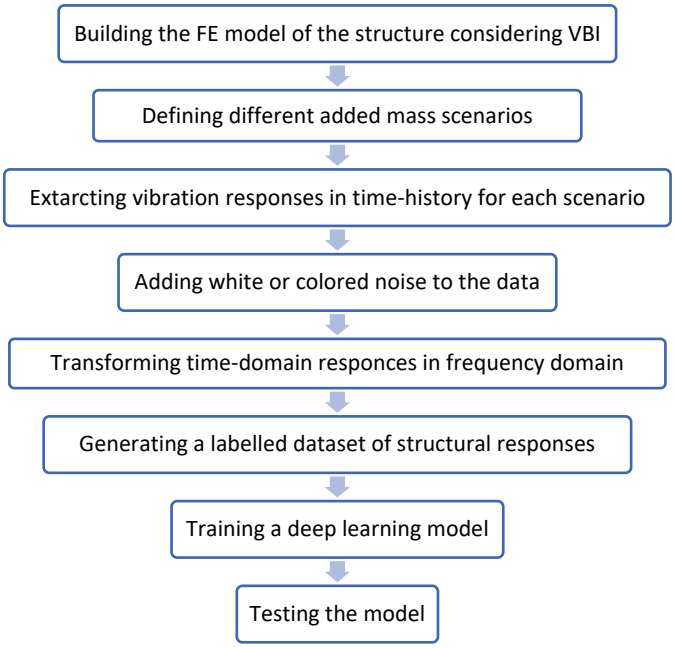


Figure 3.1 Deep Learning Framework

3.2.2 Convolutional Neural Networks (CNNs)

CNNs stand out as one of the most promising approaches in DL, which are genuinely designed for image data. They can also handle tabular data as input. Their inputs can be one-dimensional, such as time series data; two-dimensional, such as a combination of time series or a black-and-white image; or even three-dimensional, such as colored images. (Yu et al., 2019).

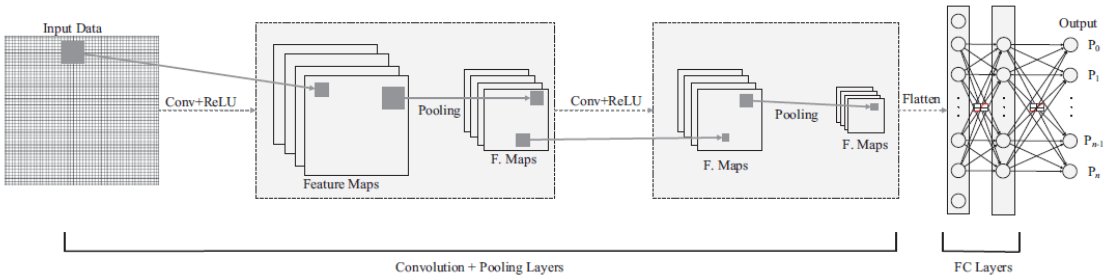


Figure 3.2 A typical convolutional neural network (Azimi & Pekcan, 2020)

The CNN, with the structure shown in Figure 3.2 is made up of a feature extraction component, including convolutional layers, concluded by FC layers. Typically, the output layer of CNNs is configured as Softmax. To enhance the performance of CNN, batch normalisation and dropout layers are also used in many applications (Zhang et al., 2019). Details of each layer and a few other concepts will be described in the following.

3.2.2.1 Convolutional layer

A convolutional layer is made of a number of windows named kernels that can slide all over the input data, conducting a convolution function by multiplying the kernel elements by the data elements and summing the results. The illustration of the convolutional kernel is presented in Figure 3.3, while the formulation can be expressed as:

$$f(i) = \sum_{n=1}^{v_k} S(i+n)K(n) \qquad \text{Eq 3.1}$$

A number of kernels will be applied to the data on each step. The number of kernels applied in a convolutional layer determines its depth, which is a hyperparameter selected by the user (Yu et al., 2019). The size of the kernels and the stride, which is the step length of the scrolling kernel, are hyperparameters in convolutional layers (Zhang et al., 2019). An activation function will be applied to the output of each convolution layer.

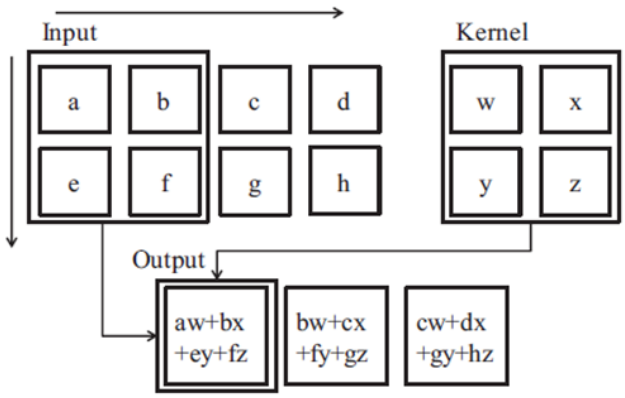


Figure 3.3 An example of a convolution kernel (Lin et al., 2017)

3.2.2.2 Activation functions

The activation function plays a fundamental role in a CNN, which converts the data from a linear space to a nonlinear and more complex one through a mapping function, resulting in more abstract features. Among a variety of activation functions, the sigmoid function (Figure 3.4(a)) is one of the most popular ones, with the following formula:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad \text{Eq 3.2}$$

The simplicity in calculation and the excellent property of the sigmoid function have made it one of the popular activations for many years. However, it has its own drawbacks: Firstly, it kills gradients when the output gets close to 0 or 1, also called vanishing gradient. Secondly, the gradient always leans toward zero, which prevents parameters

from moving properly in the backpropagation mechanism. This makes it hard to train the first hidden layers, especially when the neural network becomes deeper. Since 2017, the RELU function (Figure 3.4(b)) has gained much attention (Lin et al., 2017). A ReLU function is formulated as:

$$f(x) = max(0, x) \quad \text{Eq 3.3}$$

The ReLU function has two advantages. Firstly, it preserves the gradients at the right side of the function, instead of the vanishing procedure, which improves the training speed compared to the sigmoid function. Secondly, it has a lower computational cost. However, the drawback is that the ReLU function may die in the training procedure, which is probable as weights will be updated with a large gradient, so the activation will be zero, and it will not change again because the ReLU left part is always zero. A variant of ReLU called "leaky RELU" (Figure 3.4(c)) has been developed to overcome the "dying ReLU" problem, which is as follows:

$$f(x) = \begin{cases} x, & x \geqslant 0 \\ \lambda x, & x < 0 \end{cases} \quad \text{Eq 3.4}$$

where λ is a constant value (e.g. 0.01). This formula makes it possible for a nonzero gradient to escape from dying ReLU condition where a unit is not active (Lin et al., 2017).

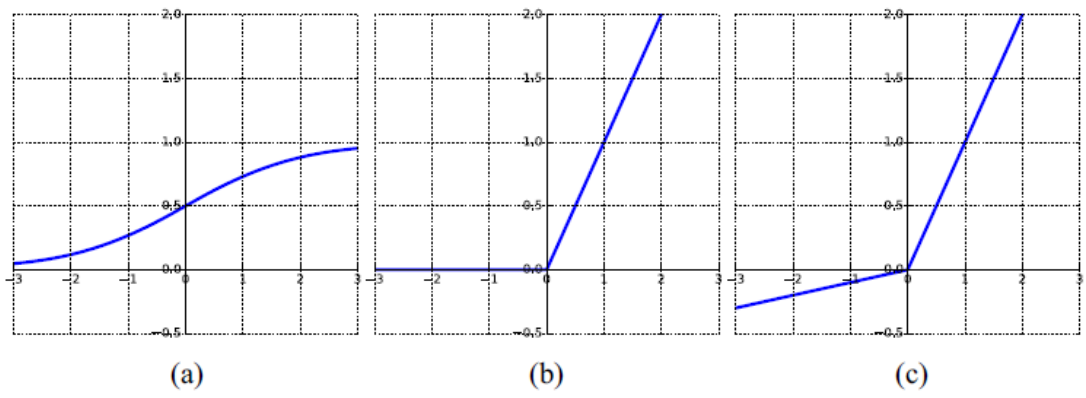


Figure 3.4 Common activation functions: (a)Sigmoid, (b) ReLU and (c)leaky ReLU (Lin et al., 2017)

3.2.2.3 Batch normalisation

The batch size is a hyperparameter and is defined as the number of training samples used to estimate the error gradient in each iteration. To solve the internal covariate shift problem in training the network, the batch normalisation (BN) layer can be used. By changing the network weight during the training phase, the distribution of internal

activations will change, which makes the distribution of the network unstable. The learning algorithm then tries to fit this instability, which results in a low convergence rate (Lin et al., 2017).The BN calculates the mean and variance of each batch of training samples and shifts the data to a zero mean and variance =1. The normalised data is then assigned a weight γ and bias β to improve the expressive power of the data.

To reduce the problem of updating coordinates in the neural network, the reparameterisation procedure can be performed in the batch normalization algorithm (Zhang et al., 2019).

$$\mu_D = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{Eq 3.5}$$

$$\sigma_D^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_D)^2 \quad \text{Eq 3.6}$$

$$\hat{x}_i = \frac{x_i - \mu_D}{\sqrt{\sigma_D^2 + \epsilon}} \quad \text{Eq 3.7}$$

$$y_i = \gamma \hat{x}_i + \beta \quad \text{Eq 3.8}$$

The batch normalisation has gained much attention recently and widely used in configuration of CNN due to a many reasons: a) it reduces the number of iterations the network needs, so the training procedure becomes faster and deeper network can be trained, b) possibility to have higher learning rate that was challenging with gradient descent, c) more activation functions can be used because they are normalised with BN, d) the overall accuracy is improved through BN (Azimi & Pekcan, 2020).

3.2.2.4 Sub-sampling layer (Max-pooling layer)

In CNNS, the max pooling layer follows the convolutional layer, decreasing the input data's dimensions and improving the quality of the features extracted. A demonstration of 1-D max pooling is shown in the Figure 3.5. The max pooling operator reduces the size of the feature map by selecting the maximum value from each input region. (Yu et al., 2019).

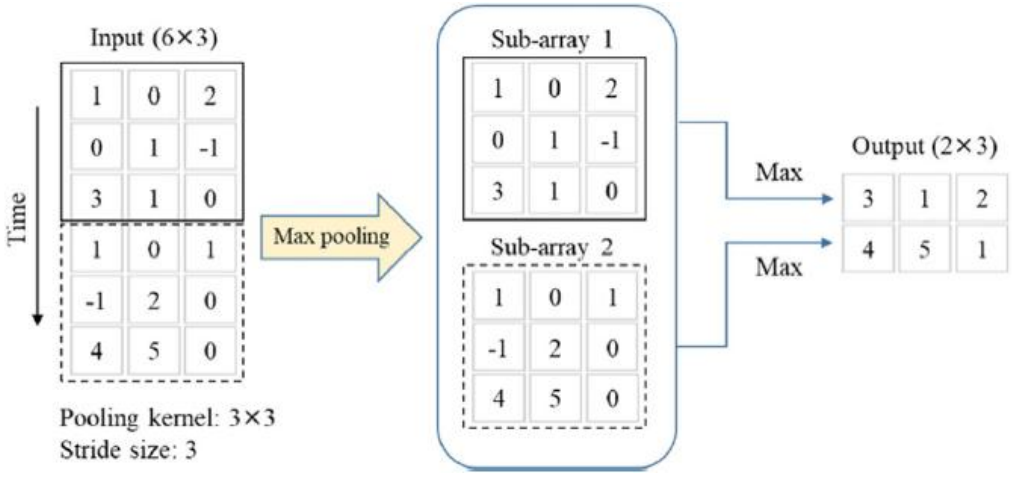


Figure 3.5 Demonstration of a 1-D max pooling (Zhang et al., 2019)

3.2.2.5 Dropout

To solve the overfitting problem, the dropout operation can be applied after flattening, which deactivates some neurons during the training procedure as shown in Figure 3.6 and reactivate them again in validation. This procedure improves the performance of the network with a low computational cost (Zhang et al., 2019).

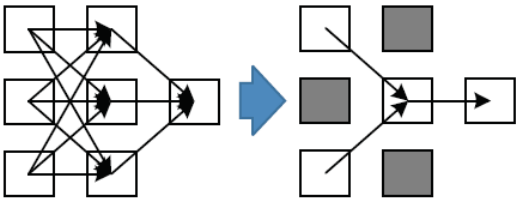


Figure 3.6 Dropout operation (Yu et al., 2019)

3.2.2.6 Fully Connected (FC) layer

An FC layer constitutes the regressor or classifier. The neurons within this layer are linked to all the ones in the preceding layer in a fully connected manner. Consequently, the inputs are multiplied by the weight w and then summed. A bias value b is then added to the latter. The result is an input of activation function f that computes the output. w and b can be trained in the network. (Zhang et al., 2019).

$$y = f(\sum u \times w + b) \qquad \text{Eq 3.9}$$

3.2.2.7 Softmax output layer

The Softmax in the last layer of the CNN network, which identifies the outputs as a structural state. In this layer, the probabilities of all predictive candidates are computed,

and then the output/final candidate will be selected based on the highest calculated probabilities (Zhang et al., 2019).

$$y_i = \frac{\exp(u_i)}{\sum_{i=1}^{n} \exp(u_i)} \quad \text{Eq 3.10}$$

Softmax layer converts the input to a vector in the range [0,1]. Each element is considered as a probability that was estimated by the CNN model (Lin et al., 2017).

3.2.2.8 Cross-Entropy Loss Function

The network is updated in every batch based on the difference between the estimated values derived from the CNN and the actual location of local structural state change (i.e. target values). The categorical cross-entropy serves as the loss function for the convolutional neural network and can be mathematically represented as follows:

$$E = -\frac{1}{N}\sum_{x} [y\ln a + (1-y)\ln(1-a)] \quad \text{Eq 3.11}$$

Where E is the loss function, N represents the total number of samples within the training dataset, x and y denote the samples and targets, respectively, and a is the predicted value generated by the CNN. (Zhang et al., 2019).

3.2.3 Transfer Learning Technique

A transfer learning approach is used to carry over knowledge acquired in the source domain to the target domain. In this research, the source is the bridge finite element model. There is no limitation in simulating damage scenarios using this model and a comprehensive labelled dataset can be generated. The target domain is a similar but not identical structure that comes with only a limited amount of data that can be either labelled or unlabelled. This structure can be a real bridge or an experimental model. An easy approach to verify the proposed TL method is to build another finite element model with a slight change in its geometry and material properties. A limited number of data points should be generated by this model since it is representing the target domain.

There are two primary categories of TL: inductive and transductive.

Inductive TL assumes the target task has some labelled data available; The model is initially trained on a source task and subsequently fine-tuned on the target task to enhance its performance.

Transductive TL can be helpful where the target domain data are unlabelled; the model must adapt to the new domain's distribution without direct supervised guidance on that domain.

Assuming the target domain contains at least one labelled data point belonging to each damage scenario, an inductive TL approach like "fine-tuning" could be a straightforward method for TL. The feature extraction model trained using the source data is used to extract the features from the target data. Then, a new classifier will be trained on the extracted features. The unlabelled data can be used for testing the model (Figure 3.7 (a)).

However, if the target domain contained only unlabelled data for damage scenarios, a transductive TL approach such as "domain adaptation" strategy could be employed to transfer the knowledge between the source and target domains (Figure 3.7 (b)). .

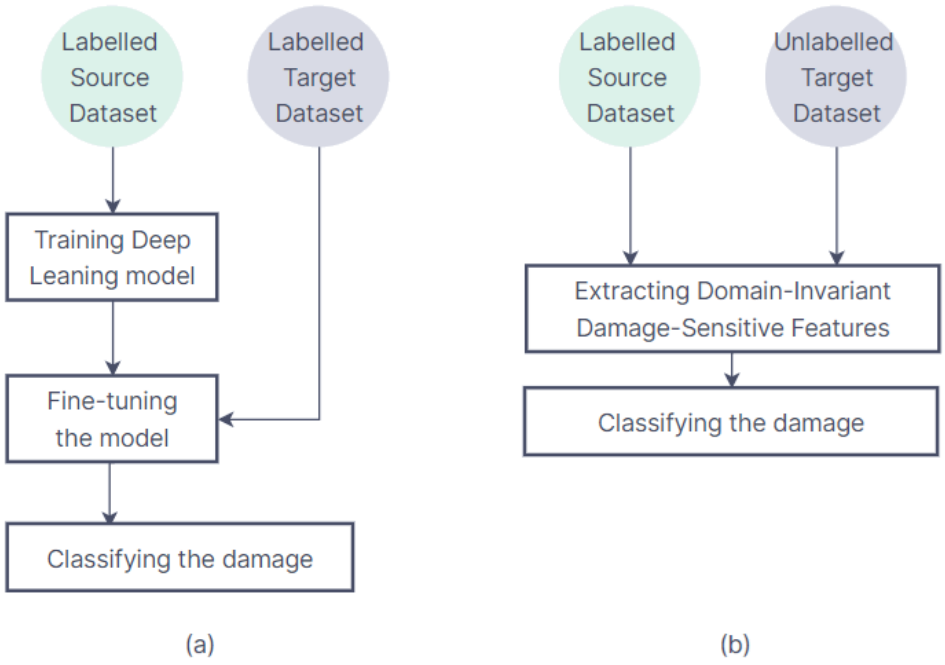


Figure 3.7 (a)Inductive and (b)Transductive TL approaches

In CNNs, the general concept is to pre-train a model using the source data. The hyperparameters of a number of convolutional layers are frozen throughout the fine-tuning process. This is done in consideration of the fact that these layers include information that is useful for feature extraction from the target domain data (Niu et al., 2020). Different fine-tuning strategies might be used, depending on the degree to which the data in the source domain and the data in the target domain are similar (Figure 3.8). The more similar the two domains are, the more convolutional layers can be frozen. When

the domains are very similar, it is possible to freeze all the convolutional layers which perform as the feature extractors of the CNN.

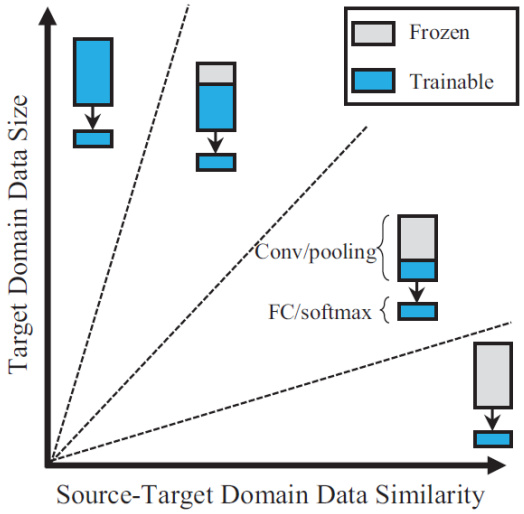


Figure 3.8 Different fine-tuning tactics based on similarity of the two domains (Azimi & Pekcan, 2020)

In this study, a CNN model will be trained using a large number of labelled datasets generated by a FE model. These datasets will be made of the vibrational response of the bridge for each damage scenario, along with its label. Since the source and target domains are pretty similar, we can freeze all the convolutional layers. It means that the feature extraction part of the trained network will be used to extract the damage-sensitive features of the target domain. New fully connected layers will be trained using the target domain data. Since all the classes in target domain need to have at least one labelled representative, this study can only be conducted using numerical models. Thus, a numerical FE will be generated with a few differences from the original one. These differences can be in geometry or the properties of the structural materials.

3.3 Case Study

In this section, a TL fine-tuning method is employed to transfer knowledge from a composite bridge finite element model to another finite element model.

3.3.1 Bridge Finite Element (FE) Model

The bridge deck FE modelling is performed using ANSYS 2021 R2 software. Two 6-meter main steel girders with a cross section of 150UC 30 and a 6x1m concrete slab are modelled. The structural elements are defined as shell elements, and the mesh can be observed in Figure 3.9. Fixed boundary conditions are defined on both ends of the steel girders.

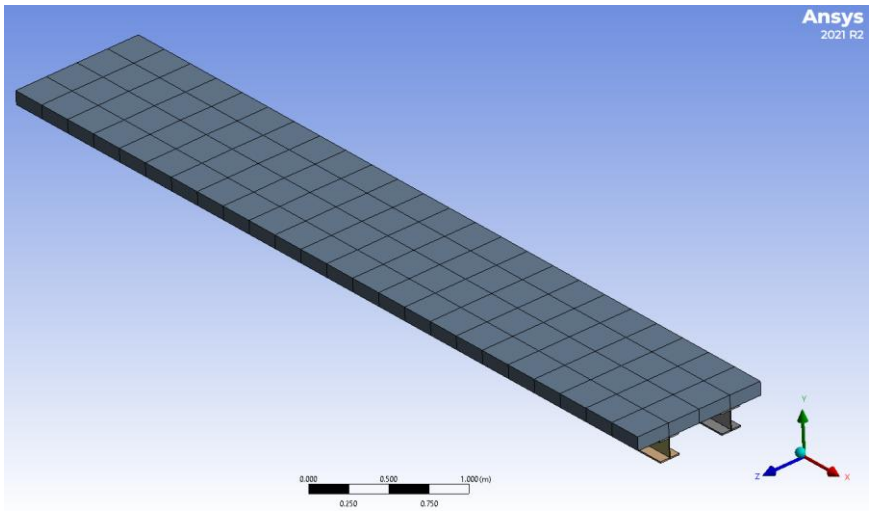
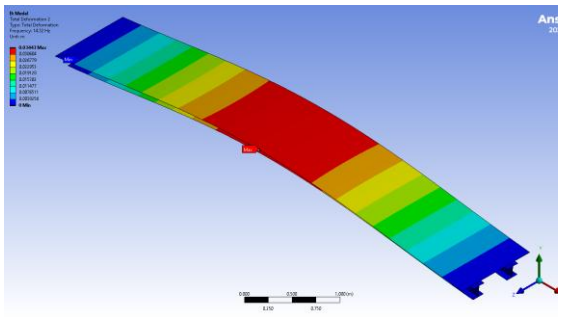


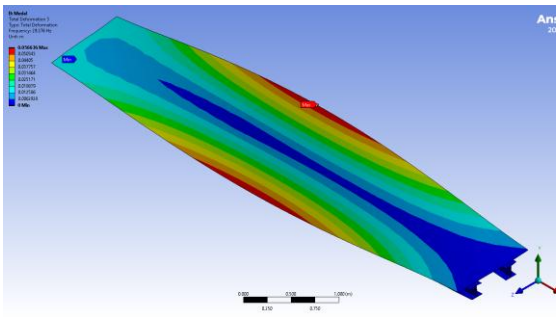
Figure 3.9 Finite element model of the composite bridge in ANSYS 2021

The bridge underwent modal analysis, revealing its bending and torsional mode shapes and their respective natural frequencies, as illustrated in the Figure 3.10.

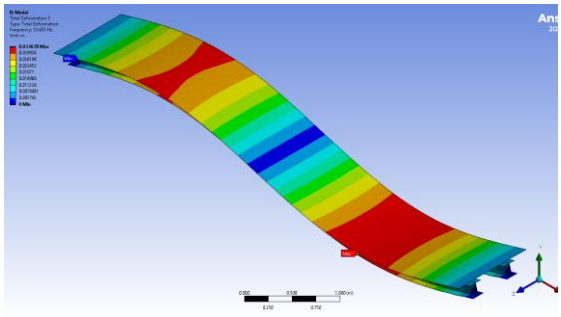
Eighteen accelerometers are positioned on the perimeters of the concrete slab to record the vertical vibrational response of the bridge. The locations of accelerometers are presented in Figure 3.11 a.



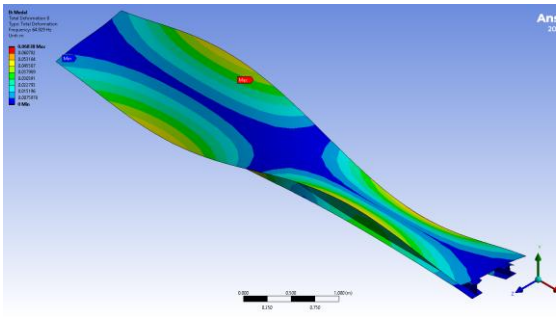
1st Bending mode - F=18.661 Hz



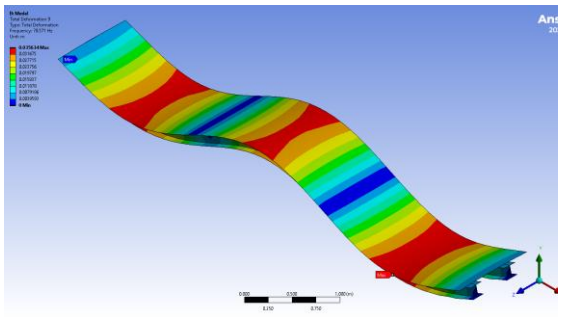
1st torsional mode - F=35.669 Hz



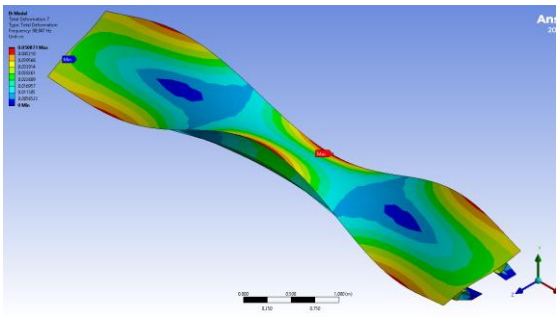
2nd Bending mode - F=44.162 Hz



2nd torsional mode - F=78.345 Hz



3rd Bending mode - F=103.47 Hz



3rd torsional mode - F=130.28 Hz

Figure 3.10 First three bending and torsional modeshapes and natural frequencies

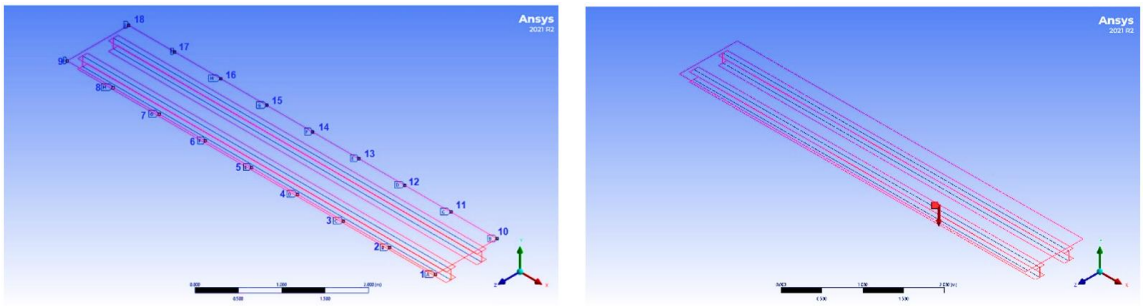


Figure 3.11 Location of a) accelerometers, and b) impact force on the bridge concrete deck

The excitation force is defined as a 100N impact force in a certain location on the bridge deck (1.75 m from the short edge and 0.25m from the long edge of the concrete slab) as is shown in 3.11 b . Time history data of the impact force is used for structural transient analysis 3.12 a. The total time is set to be 10 sec and the time step is defined as 0.005 sec. The vibrational response of the bridge is then recorded by the accelerometers in the time domain. A sample of structural responses captured from Sensor No.13 is shown in Figure 3.12 b.

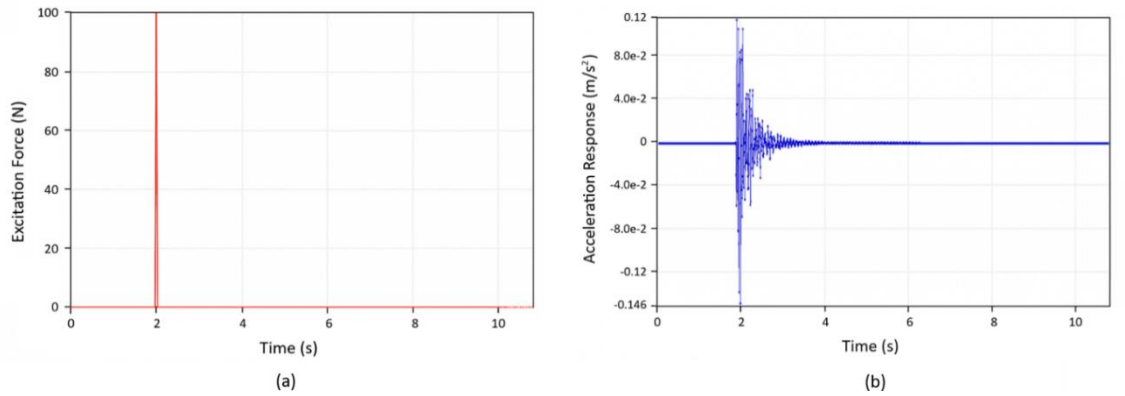


Figure 3.12 Time history of a) the impact force and b) bridge acceleration response in Y direction captured by sensor 13

3.3.2 Vehicle Location Estimation from Bridge Responses Using Deep Learning

Condition assessment of a bridge using an added mass is conducted in this study, utilising convolutional neural networks. The added mass ranges from 90 to 180 kg, which is 5% to 10% of the structure's mass (1800 kg). Thirteen different scenarios are defined based on the distance of the added mass from the short edge of the concrete deck (x), with 0.5 m intervals (Figure 3.13).

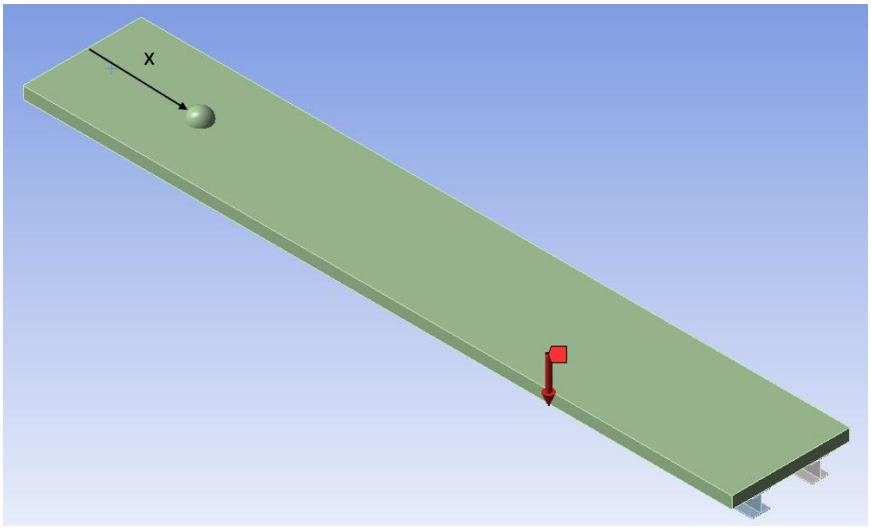


Figure 3.13 Added mass on the bridge deck

The acceleration response of the bridge is recorded by an array of 18 uniaxial accelerometers mounted along the deck edges. For each sensor, vertical acceleration $a_k(t)$, a frequency-response function (FRF) is formed with respect to the impact excitation $f(t)$ via the discrete Fourier transform:

$$H_k(\omega) = \frac{A_k(\omega)}{F(\omega)} = \frac{\mathcal{F}\{a_k(t)\}}{\{f(t)\}}, k = 1, \dots, 18 \qquad \text{Eq 3.12}$$

where $A_k(\omega) = \mathcal{F}\{a_k(t)\}$ and $F(\omega) = \mathcal{F}\{f(t)\}$ are the complex spectra of the response and excitation, respectively. The FRF $H_k(\omega)$ shows how a unit harmonic load is transmitted to the sensor k over the measured bandwidth.

To create a compact, real-valued input for the CNN, the magnitude of the FRF is first converted to the natural logarithmic scale and then normalised by its maximum value:

$$X_k(\omega) = \frac{\log\left(|H_k(\omega)|\right)}{\max_{\omega}[\log\left(|H_k(\omega)|\right)]} \qquad \text{Eq 3.13}$$

where $|H_k(\omega)|$ is the modulus of the FRF. This log-magnitude normalisation compresses the large dynamic range, amplifies resonance peaks and yields a dimensionless spectrum $X_k(\omega) \in [0,1]$ that helps CNN training.

Data augmentation is conducted by adding 2 % white Gaussian noise to the raw acceleration data before the FRF calculation. As a result, each sample in the training dataset consists of 18 normalised log-magnitude spectra. A number of samples of the CNN input dataset are illustrated in Figure 3.14.

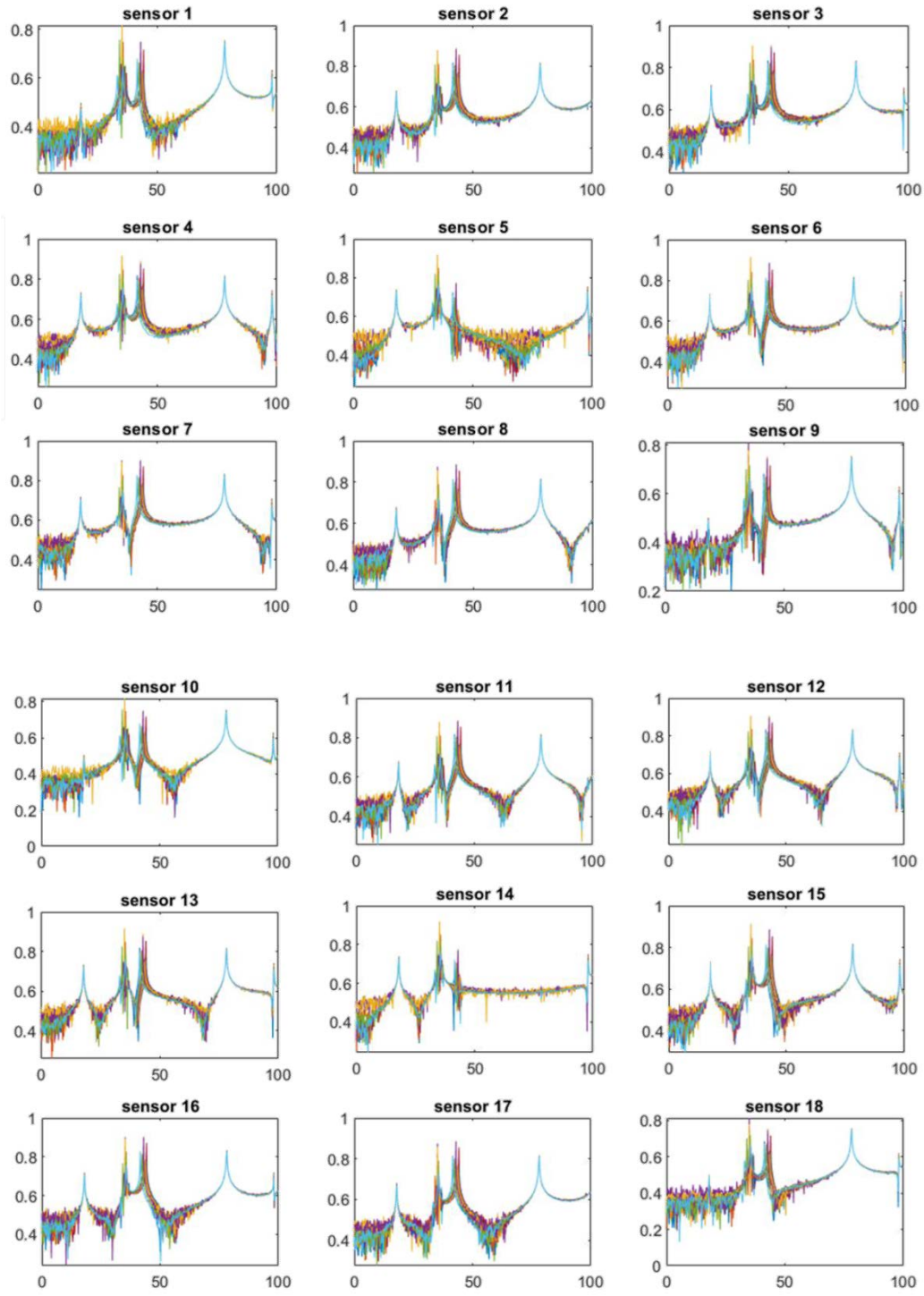


Figure 3.14 Samples of the CNN input dataset

A convolutional neural network is then trained using 80% of the presented data in the dataset and tested with the remaining 20%. The input data is a 500x18 matrix corresponding to FRF data captured from 18 sensors with a frequency resolution of 0.2 Hz from 0 to 100 Hz. This matrix can be reshaped to a 100x90 matrix for a more

convenient feature extraction process. The output layer contains 13 neurons corresponding to 13 scenarios. Convolutional layers are supposed to do the feature extraction and fully connected layers are employed for classification.

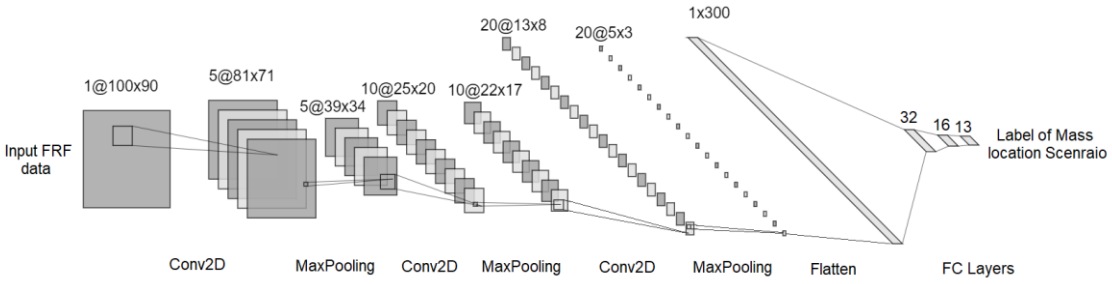


Figure 3.15 a sample of the CNN structure

The structure of CNN can be optimised on a trial-and-error basis. The learning rate in all the models is set as 10e-6, and the batch size is set as 32. These values are also selected by trial and error. Regarding the CNN structure, It is important to note that improved performance does not always result from adding more convolutional layers while It's crucial to prevent overfitting in which the network will learn the correlation between input data and output labels in the training phase with a high accuracy but cannot generalise the learnt model to unseen data in the testing phase. Three different structures and the training accuracy and loss functions are presented below.

Table 3.1 The structure of CNN No.1 and loss/accuracy graphs during the training process

	No. of kernels/ Neurons	Kernel Size	Training Accuracy and Loss
Input	1	100×90	
Conv2D	5	20×20	
Batch Normalization			
ReLU			
Maxpooling		4×4	
Conv2D	10	15×15	
Batch Normalization			
ReLU			
Maxpooling		4×4	
Conv2D	20	10×10	
Batch Normalization			
ReLU			
Maxpooling		4×4	
Flatten			
Dence	32		
Batch Normalization			
ReLU			
Dence	16		
Batch Normalization			
ReLU			
Dropout (0.5)			
Dence (Output)	13		

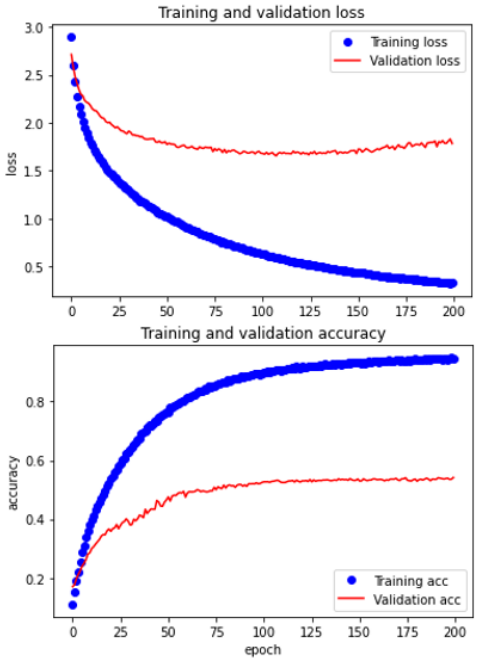


Table 3.2 The structure of CNN No.2 and loss/accuracy graphs during the training process

	No. of kernels/ Neurons	Kernel Size	Training Accuracy and Loss
Input	1	500×18	
Conv2D	5	483×1	
Batch Normalization			
ReLU			
Maxpooling		2×2	
Conv2D	10	10×10	
Batch Normalization			
ReLU			
Maxpooling		2×2	
Conv2D	15	5×5	
Batch Normalization			
ReLU			
Maxpooling		2×2	
Flatten			
Dence	32		
Batch Normalization			
ReLU			
Dence	16		
Batch Normalization			
ReLU			
Dropout (0.5)			
Dence (Output)	13		

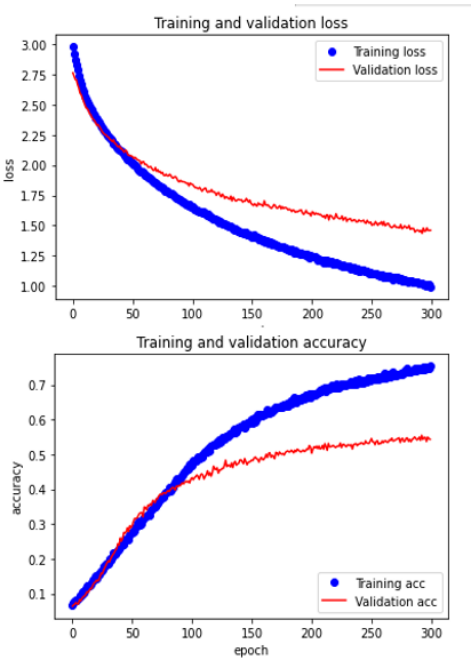
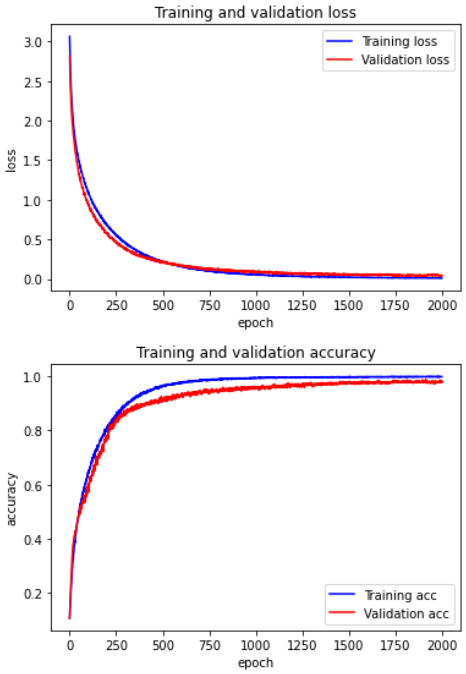


Table 3.3 The structure of CNN No.3 and loss/accuracy graphs during the training process

	No. of kernels/ Neurons	Kernel Size	Training Accuracy and Loss
Input	1	100×90	
Conv2D Batch Normalization ReLU	3	50×50	
Conv2D Batch Normalization ReLU	5	30×30	
Conv2D Batch Normalization ReLU	7	10×10	
Flatten			
Dence Batch Normalization ReLU	64		
Dropout (0.5)			
Dence (Output)	13		



The classification accuracy of the trained network is tested using datasets with different noise levels added to the acceleration signals. The outcomes are tabulated in Table 3.4.

Table 3.4 Classification accuracy of CNN No.3

Noise-to-Signal Ratio	2%	5%	7%	10%
Classification Accuracy (%)	100	99.51	97.57	89.96

3.3.3 Fine-Tuning

Although the trained CNN's accuracy in classifying source domain labels is acceptable, this knowledge can be transferred to a target domain. A second FE model with the same geometry but different material properties, representing the target domain, is shown in the Table 3.5.

Table 3.5 Differences between source and target structures in material properties

		Source	Target
Steel	Young's Modulus (Pa)	2.00e11	2.1e11
	Density (kg/m^3)	7850	7845
Concrete	Young's Modulus (Pa)	1.94e10	1.5e10
	Density (kg/m^3)	2392	2000

Table 3.6 compares the natural frequencies and Figure 3.16 depicts the frequency response functions of the source and the target domains. The difference between natural frequencies is not discernible. However, the domain shift is evident in the FRF data that serve as input to the CNN.

Table 3.6 Differences between source and target natural frequencies

# Mode	Natural Frequencies (Hz)	
	Source	Target
1	18.661	18.885
2	19.362	21.059
3	35.669	35.167
4	37.162	40.402
5	44.162	46.501
6	78.345	77.730
7	94.149	96.072
8	98.373	101.41
9	103.470	107.260
10	107.370	109.710

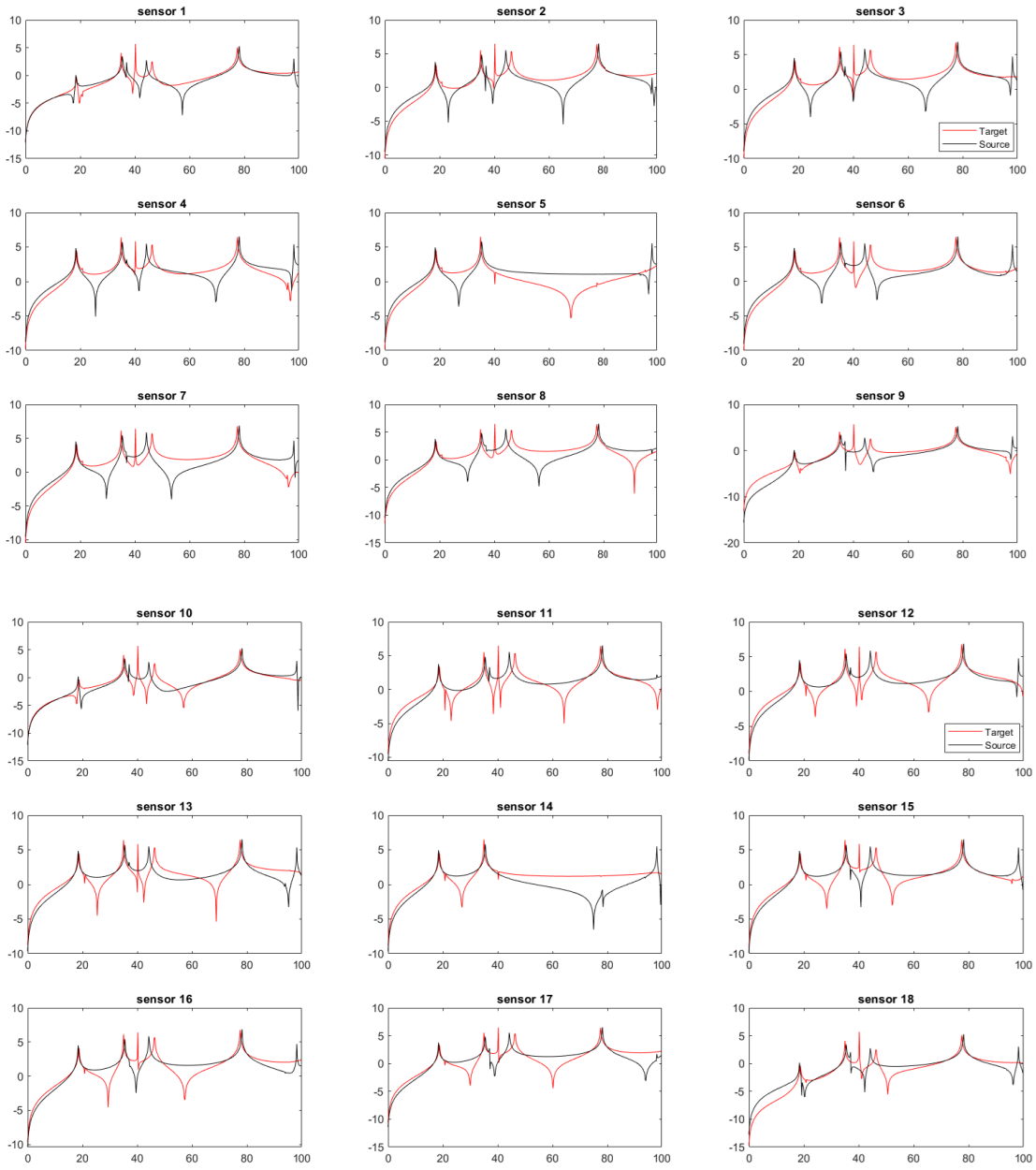


Figure 3.16 Differences between source and target domains

As observed, the feature spaces for source and target domains are identical, and both tasks share the same labels. However, their marginal and conditional distributions differ.

Because the source and target domains are similar, all convolutional layers are frozen during fine-tuning. This allows the pre-trained CNN's feature extraction component to be used for capturing damage-sensitive features from the target domain data.

A classifier will then be trained to correlate the extracted features to the output label. Damage-sensitive features in the source and target domains are visualised in Figure 3.17 using t-SNE, a statistical technique for visualising high-dimensional data.

It can be observed that the damage-sensitive features in both domains are distinctly categorised into 13 separate clusters, each corresponding to a specific mass location scenario or class.



Figure 3.17 t-SNE visualisation of source and target domain damage-sensitive features

In the target domain, the amount of added mass is assigned to be 110,120,130,140,150, and 160 kg. the FRF data corresponding to 140 kg added mass are assumed to be labelled and are employed for training the classifier. The rest of data are assumed to be unlabelled and are utilised for testing the fine-tuning method. The vibrational responses in target domain are polluted with 2% of white Gaussian noise.

A fully connected deep neural network (DNN) comprising 128 neurons, utilising the Relu activation function, along with a dropout layer, is employed independently for classification purposes. Additionally, a random forest algorithm with 500 estimators is utilised for classification. The classification outcomes are presented in Figure 3.18 and Figure 3.19, respectively. The accuracy of a fully connected DNN classifier is 79.56% while the accuracy of a random forests classifier is 83.56%, and an ensemble of the two classifiers will lead to 92.1% accuracy on the target domain data.

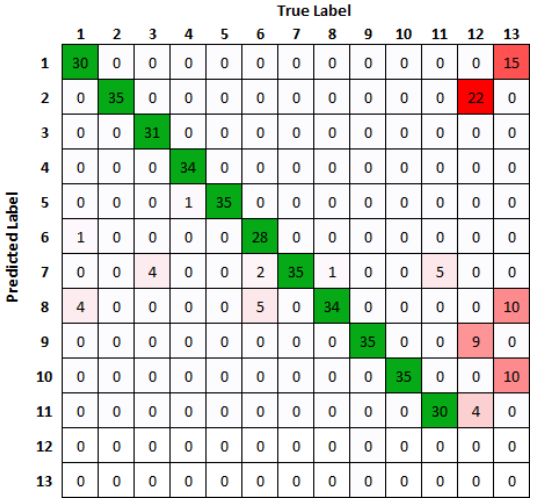


Figure 3.18 Confusion matrix of the target classification after fine-tuning using DNN FC classifier

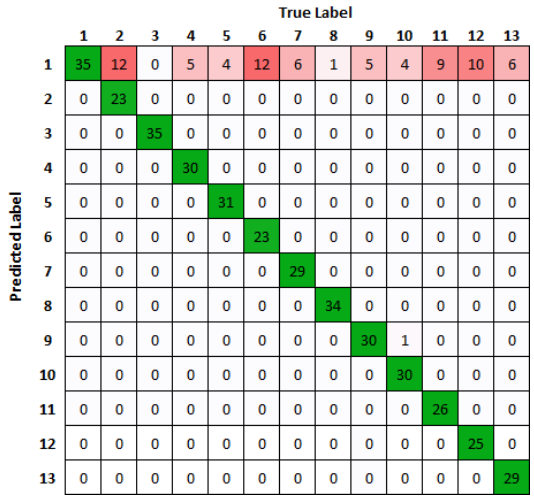


Figure 3.19 Confusion matrix of the target classification after fine-tuning using a random forest classifier

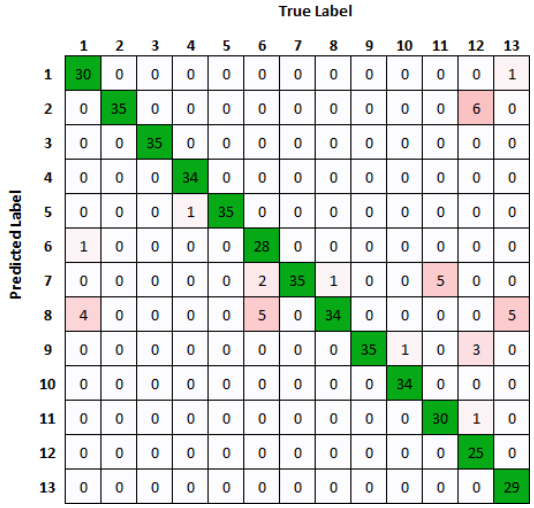


Figure 3.20 Confusion matrix of the target classification after fine-tuning using an ensemble of classifiers

3.4 Summary

This chapter presented a methodology for structural parameter identification of bridge structures using FRFs and fine-tuning techniques applied to CNNs. The approach focused on addressing challenges associated with identifying the changes in structural mass due to damage or modifications. CNNs were utilised due to their superior capability in capturing complex patterns from frequency-domain data. The effectiveness of fine-tuning pretrained CNN models is evaluated on simulated datasets derived from FE bridge models under discrepancies between two shifted domains.

The key findings of this chapter are summarised as follows:

- CNN models demonstrated high accuracy (100%) in classifying structural changes using FRFs with a 2% noise-to-signal ratio. The accuracy remained robust at 99.51% and 97.57% when noise-to-signal ratios were increased to 5% and 7%, respectively, and dropped to 89.96% at a 10% noise-to-signal ratio.
- The customised CNN architecture, with layers designed to capture patterns in log-normalised FRF spectra, demonstrated its capability of extracting damage-sensitive features from the input data.
- The fine-tuning method proved to be applicable for transferring knowledge when both domains have an identical feature space but the target data alone are insufficient to train a CNN from scratch; in such scenarios, a CNN can be trained on abundant, labelled source data and subsequently fine-tuned with limited, labelled target data.
- The final classification results show that using an ensemble of multiple classifiers can lead to more robust damage detection and significantly improve classification accuracy to 92.1% on the target domain data.
- The proposed approach has inherent limitations: this study concentrated on a specific damage mechanism (lumped mass addition) and required a minimum of several labelled target samples to enable fine-tuning, so its effectiveness for other damage types or completely unlabelled target scenarios remains uncovered.
- It is acknowledged that the extraction of FRFs in practice requires knowledge of the excitation location and loading time history. Modal parameters such as natural frequencies and mode shapes could also be explored as alternative inputs to overcome this issue.

CHAPTER 4 - TRANSFER LEARNING BASED BRIDGE DAMAGE DETECTION: LEVERAGING TIME-FREQUENCY FEATURES

4.1 Overview

Building on the CNN models trained and fine-tuned in Chapter 3, this chapter fine-tunes pre-trained, well-established CNN models to advance the proposed framework toward a more efficient damage classification by applying transfer learning-based methods for bridge damage detection using time-frequency representations of Vehicle-Bridge Interaction (VBI) responses. Traditional vibration-based damage detection methods often have limitations in accurately detecting localised damage because global modal parameters, such as natural frequencies and mode shapes, are generally insensitive to local structural changes. VBI-based approaches have an improved sensitivity to local damages since they utilise the time-varying characteristics of the dynamic bridge responses, subjected to moving vehicle loads.

Time-frequency representations, such as wavelet transforms, are chosen to capture rich dynamic information by characterising the variability of the signal's frequency components over time. This leads to the extraction of features that are sensitive to localised damage.

As an extension of the work in Chapter 3, which primarily focused on fine-tuning the trained CNNs, this chapter examines the fine-tuning of globally recognised pretrained CNNs. The time-frequency features extracted from VBI acceleration responses are used to fine-tune the pre-trained CNNs for classifying bridge damage locations. The efficacy of the proposed method is assessed through a comparative analysis with various pre-trained networks and optimised classification algorithms.

The effects of the noise level, vehicle speed, and sensor location on the predicted results are also examined. The numerical results demonstrate that the proposed method can accurately identify damage on concrete bridges using solely a single sensor positioned on

the bridge deck. This method possesses significant potential for practical implementation in localising structural damage in bridges, with further refinement possible through the utilisation of field measurement data.

4.2 Related Works

Bridges are designed to sustain different cases of load combinations, including dead, live and environmental loads. The majority of the applied live load on bridge structures results from moving vehicles (Shokravi et al., 2020). When a vehicle passes the bridge damage location, it will induce the local dynamic response (Law and Zhu, 2004). The damage detection approaches that utilise interactive responses of the bridge with moving vehicles have a high sensitivity in detecting local damage and therefore have attracted the research interest of many researchers and engineers (Zhu & Law, 2015). Considering the traffic excitation is often unknown in practice, Zhu and Law (2006) presented a method to identify the structural damage and the moving vehicular loads simultaneously. Li et al. (2013) proposed a method for the detection of structural damage in bridges using vehicle-bridge interaction responses without requiring any prior knowledge about the moving vehicle. Zhan et al. (2021) presented a method to identify the damage of prestressed concrete box-beam bridges with the correction function of response spectra between two adjacent beams is used as the indicator of the damage.

Utilising the bridge's acceleration response to moving vehicles for the purpose of structural damage detection provides numerous advantages.

1) The structural damage detection is conducted under operational conditions, and the identified results are much more reliable and accurate (Adhikary & Mutsuyoshi, 2004).

2) In comparison with the traditional long-term bridge monitoring system, the number of required sensors can be significantly reduced. Sensors can be installed on the bridge and/or the moving vehicle.

Consequently, the use of vehicle-bridge interaction (VBI) dynamics improves structural damage identification performance for the bridge structures (Kim et al., 2011) while the vehicle can function as a moving exciter. However, these methods need an accurate numerical model of the VBI system. Uncertainties, such as the modelling errors, the bridge road surface roughness, measurement noise, and the vehicle-bridge contact model can significantly affect the damage identification results.

Data-driven methods could provide a potential solution for these limitations as they are able to evaluate the bridge conditions by utilising a large amount of high-dimensional collected data through various machine learning (ML) algorithms. Traditional machine learning methods depend significantly on feature extraction. Poor feature extraction can result in subpar features, which can negatively impact classification accuracy (Fallahian et al., 2018). However, deep learning (DL) methods introduce a paradigm-shifting framework capable of autonomously extracting optimal features from raw data, resulting in enhanced classification and regression performances (Pathirage et al., 2018). The feature extraction procedure is embedded in the DL networks, making these algorithms well-suited for structural damage detection (Krizhevsky et al., 2017). However, a considerable amount of high-quality labelled data is required to effectively train the DL networks (Sun et al., 2020). This constraint limits the practical application of DL in bridge damage detection as it is exceedingly challenging or even impossible to obtain a large amount of damaged state data of a bridge.

Several CNNs have been trained on over one million ImageNet images to categorise images into a thousand classes. Using the knowledge from these strong pretrained networks to train a DL algorithm for a specific task can be effectively achieved through Transfer Learning (TL) methods. Applying this concept to address the challenges of structural damage detection makes it even more captivating. Parameter-based TL techniques use a subset of parameters from a source-trained model to develop a model for the target domain. In CNNs, the goal is to fine-tune the pretrained model using training data from the target domain. For fine-tuning, the hyperparameters of a group of convolutional layers (or all of them) will be frozen, considering that these layers contain feature-extraction-relevant data (Zhuang et al., 2021). Recently, Tang et al. (2022) proposed a transfer learning-based load identification method based on the power spectral density (PSD) of VBI response. the process of a vehicle crossing the bridge varies over time, and the time-dependent features of the VBI system may serve as indicators of structural damage detection. (Zhu & Law, 2015). These time-varying parameters show a high sensitivity to local damages, especially at the time that the vehicle passes by the local damage location (Bu et al., 2006). A time-frequency analysis method can be used to reveal the damage-sensitive feature from acceleration responses of the VBI system (Li et al., 2020). Singh and Sadhu (2021) presented a hybrid approach to monitor the bridge's changing modal parameters by combining time-varying filtering-based empirical mode

decomposition with synchro-extracting transform. The time-varying characteristics of the nonstationary VBI response can also be captured using Continuous Wavelet Transform (CWT), which represents the signal evolution in both time and frequency (Taha et al., 2006). Tan et al. (2020) integrated the Shannon entropy with CWT to detect the damage of the bridge by depicting the signal discontinuities of structural dynamic responses. Silik et al. (2021) conducted a comparative study on using wavelet-based time-frequency features in structural health monitoring, showing its effectiveness in extracting damage-sensitive features.

In this study, the acceleration responses of a vehicle-bridge interaction system are represented in the time-frequency domain using the CWT, and the time-frequency 2D representation of the signals is utilised to build a labelled training dataset of images. Fine-tuning is then employed to leverage different well-established pre-trained networks for the purpose of locating damage. Fine-tuning in this study entails replacing the pre-trained network's last classification block with a new classification algorithm. It implies that the inputs of classification algorithms are the features extracted from pre-trained CNNs. VGG-16, GoogLeNet, ResNet-50 and AlexNet are the four pre-trained convolutional networks that are investigated in this study, and k-Nearest Neighbour (KNN), Support Vector Machine (SVM), Fully Connected (FC) layers, Ensemble Subspace Discriminant (ESD) and Ensemble Decision Trees are the classification algorithms employed for fine-tuning. An optimisation process is also applied to each classifier hyperparameter to achieve its optimal performance. As a result, the comparison of damage classification performance is made among the best results of each classifier. The top-performing combination of the pretrained network and the optimised classifier is selected. The performance of the fine-tuned network is investigated through four distinct studies, including investigations on the effects of noisy acceleration data, passing vehicle speed, sensor location and road surface roughness on the performance of the suggested method.

4.3 Proposed Methodology

The basic theory underpinning the proposed approach is presented in this section. First, the VBI system is described, which is utilised to generate the datasets for the proposed transfer learning based structural damage detection. Next, the damage model is then described for simulating different damage scenarios. The technique of extracting time-varying characteristics from VBI acceleration responses is then outlined, followed by a description of the TL algorithm used for structural damage classification.

4.3.1 Vehicle-Bridge Interaction (VBI) Analysis

For the VBI system, the passing vehicle can be represented by the moving load, moving mass, and moving mass-spring system. To fully account for the interaction between the vehicle and bridge, a quarter-vehicle model is shown in Figure 4.1. The vehicle travels at a constant speed v across a simply supported bridge with a uniform cross-section, maintaining continuous contact with the bridge surface throughout its passage. At any time t, the vertical displacement of the vehicle is influenced by three components: the vertical displacement of the bridge at the contact point, the relative displacement between the vehicle mass and the contact point, and the roughness of the road surface.

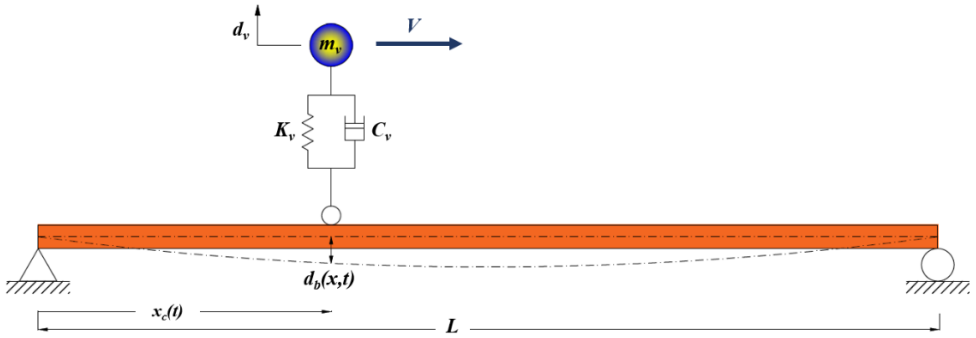


Figure 4.1: The VBI model (Li et al., 2020)

The bridge is discretised into N beam elements. The equation of motion for the vehicle and bridge subsystems can be expressed as:

Bridge: $$\boldsymbol{M_b}\ddot{\boldsymbol{d}}_{\boldsymbol{b}} + \boldsymbol{C_b}\dot{\boldsymbol{d}}_b + \boldsymbol{K}_b\boldsymbol{d}_b = \boldsymbol{\phi}(t)\boldsymbol{P}_{bint}(t) \quad \text{Eq 4.1}$$

Vehicle: $$m_v\ddot{d}_v + c_v\dot{d}_v + k_v d_v = P_{vint}(t) \quad \text{Eq 4.2}$$

where $\boldsymbol{d}_b, \dot{\boldsymbol{d}}_b, \ddot{\boldsymbol{d}}_{\boldsymbol{b}}$ are the nodal vertical displacement, velocity, and acceleration vectors of the bridge, respectively. $\boldsymbol{M}_b$, $\boldsymbol{C}_b$ and $\boldsymbol{K}_b$ denote the mass, damping and stiffness matrices of the bridge respectively. Similarly, m_v, c_v and k_v represent the mass, damping and stiffness of the moving vehicle, and the displacement, velocity and acceleration of the vehicle are $d_v, \dot{d}_v, \ddot{d}_v$, respectively. $\boldsymbol{P}_{bint}(t)$ corresponds to the interacting force vector between the two sub-systems on the bridge system. $P_{vint}(t)$ stands for the interaction force on the vehicle system. It should be noted that the number of interaction forces needs to be set as the number of vehicles. Given the number of vehicles is N_p and the corresponding moving interaction force vector $\boldsymbol{P}(t) = \left\{P_1(t), P_2(t), \dots, P_l(t), \dots, P_{N_p}(t)\right\}^T$. $\boldsymbol{\phi}(t) = \left\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 \dots \boldsymbol{\phi}_l \dots \boldsymbol{\phi}_{N_P}\right\}$ is defined as a $2(N+1)\times N_p$ matrix, while $\boldsymbol{\phi}_l = \{0\ 0\ 0\ \dots\ \boldsymbol{\psi_i}\ \dots\ 0\ 0\ 0\}^T$

and ψ_i is the vector of the shape functions evaluated for the force $P_l(t)$, located on $\hat{x}_l(t)$ (Law & Zhu, 2011):

$$\boldsymbol{\psi}_i = \begin{Bmatrix} 1-3\chi^2+2\chi^3 \\ \chi\, l_e(\chi-1)^2 \\ 3\chi^2-2\chi^3 \\ \chi^2\, l_e(\chi-1) \end{Bmatrix}^T, \chi = \left[\frac{\hat{x}_l(t)-(i-1)l_e}{l_e}\right], (i-1)l_e \leqslant \hat{x}_l(t) < (i)l_e \qquad \text{Eq 4.3}$$

Figure 4.2 depicts the *i*th element of the bridge with a length of l_e which is loaded by a moving force $P_l(t)$.

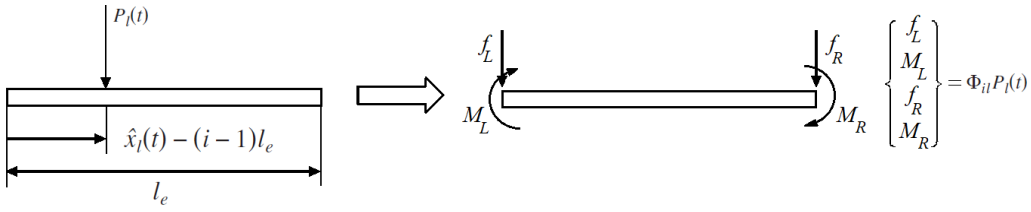


Figure 4.2 Equivalent nodal loads for a beam element loaded by the bridge-vehicle interaction force

It is evident that $\boldsymbol{\phi}(t)$ is a time-dependent factor that covers the time-varying characteristics of the moving interaction force. By combining Eq 4.1 and Eq 4.2, the equation of motion for the VBI system is obtained as:

$$\begin{bmatrix} \boldsymbol{M_b} & m_v\boldsymbol{\phi} \\ 0 & m_v \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{d}}_b \\ \ddot{d}_v \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_b & 0 \\ -c_v\boldsymbol{\phi}^T & c_v \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{d}}_b \\ \dot{d}_v \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_b & 0 \\ -k_v\boldsymbol{\phi}^T - c_v\dot{\boldsymbol{\phi}}^T & k_v \end{bmatrix}\begin{bmatrix} \boldsymbol{d}_b \\ d_v \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi} m_v g \\ k_v r(x) + c_v r'(x) V \end{bmatrix} \qquad \text{Eq 4.4}$$

Road roughness *r(x),* which represents the irregularities in the pavement surface, can be addressed as a Gaussian random process with zero mean. ISO 8608 standard defines road roughness using the power spectral density function S_{rr} as below:

$$\begin{cases} S_{rr}(\omega) = S_{rr}(\omega_0)(\omega/\omega_0)^{-2} (\omega \leqslant \omega_0) \\ S_{rr}(\omega) = S_{rr}(\omega_0)(\omega/\omega_0)^{-1.5} (\omega > \omega_0) \end{cases} \qquad \text{Eq 4.5}$$

where ω is the frequency; ω_0 stands for the standard frequency equal to $0.16\times V$ Hz and V is the speed of the moving vehicle; $S_{rr}(\omega_0)$ corresponds to the amplitude of power spectral density function at the standard frequency, which is defined as the roughness coefficient (Tang et al., 2022).

4.3.1.1 VBI Model validation

The VBI model is validated using the experimental results from Law and Zhu (2004). The test beam is a five-meter-long Tee-section reinforced concrete beam and a vehicle is

pulled along the beam by an electric motor at an approximate speed of 0.5 m/s. The masses of the light and heavy vehicles are 1081.6 and 1536.7kg, respectively. The stiffness and damping for each wheel are 6.23 × 10^6 N/(m/s) and 8.94 × 10^3 N/(m/s) respectively. The bending stiffness (EI) is calculated from the cross-section of the beam as 5.59 × 10^7 N.m^2. The detailed experimental setup can be found in the reference (Law and Zhu, 2004).

Figure 4.3 shows the comparison of the numerical results with the experimental measurements of the displacements at 3/8 of the span subjected to the light and heavy vehicles. The results show that the numerical algorithm is reliable and accurate to calculate the responses of the bridge subjected to moving vehicles. In this study, the VBI responses are calculated using the same validated algorithm.

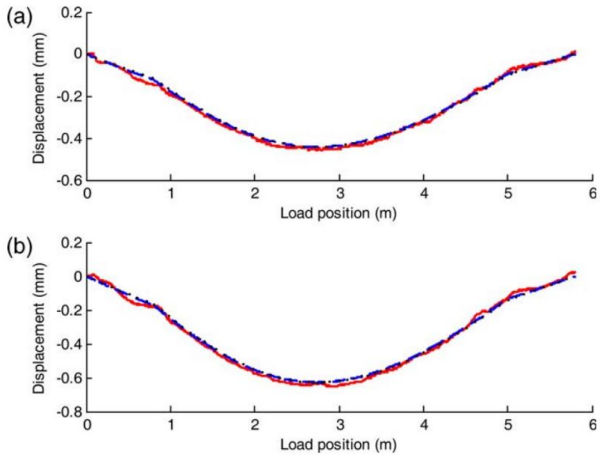


Figure 4.3 Deflections at 3/8 span of the beam subjected to: (a) Light moving loads; (b) heavy moving loads (___measured; ... VBI model) (Law & Zhu, 2004)

4.4 Damage Simulation

The damage model developed by Wahab et al. (1999) is used to simulate the crack zone of the reinforced concrete beam in this study. The stiffness reduction in a damaged zone due to crack is defined as the following function:

$$EI(x) = E_0 I\left\{1 - \gamma \cos^2\left[\frac{\pi}{2}\left(\frac{|x-l_c|}{\beta L/2}\right)^n\right]\right\} \qquad (l_c - \beta L/2 < x < l_c + \beta L/2) \qquad \text{Eq 4.6}$$

where E_0 and I are Young's modulus and cross-sectional moment of inertia of the intact beam. l_c denoted the midpoint location of the damage zone on the beam and L is the length of the beam. γ, β, and n are the damage parameters. The parameter β which is defined between 0 and 1 is corresponding to the length of the damaged zone, i.e., if the damage is in the midspan and β is 1, the damage will cover all the beam. The parameter γ which

is also defined between 0.0 and 1.0 reflects the damage magnitude, e.g., zero means no damage and 1.0 means no bending stiffness in the midpoint of the damaged zone. *n* reflects the variation of Young's modulus over the damage zone.

4.5 Time-Frequency Features

The acceleration responses of the VBI model, which were obtained by solving the coupled equation of motion Eq. (4), contain rich information on the vibrational characteristics of the VBI system. These characteristics are the main source for building a dataset for a data-driven based damage detection study. However, different data representations can highlight various aspects of the raw data. Representing data in the time domain does not accentuate the frequency content of the response, especially when the data is noise-polluted and representing data in the frequency domain leads to disregarding time-varying characteristics of the acceleration signal. Consequently, time-frequency analysis is utilised to present the time-varying frequency content of the signals, providing comprehensive features of the acceleration response in both the time and frequency domains.

For time-frequency analysis, the continuous wavelet transform (CWT) is employed. The CWT, utilised inner products to evaluate the homology of a signal and an evaluation function. But unlike STFT, which relies on a fixed set of sinusoid functions with angular frequency, the CWT employs a wavelet function that is scaled and translated in time to match the signal at each time point. Wavelet functions are intended to capture local features in signals (Taha et al., 2006). Also, CWT's variable time-frequency resolution analysis better suits non-stationary signals with varying frequency content and can offer a comprehensive representation of the time-frequency content of signals. The output of CWT can be stored as an image containing time-varying characteristics of the VBI acceleration response. A dataset of these images can be utilised to train image-based ML algorithms. Each point in this image represents the energy of the signal at a certain time and frequency. The CWT of a signal *x(t)* using a wavelet function $\psi(t)$ is:

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int x(t) \psi * \frac{(t-b)}{a} \, dt \qquad \text{Eq 4.7}$$

where *a* is the scale parameter, which determines the size of the wavelet, and *b* is the translation parameter (Balafas & Kiremidjian, 2015). ψ is the wavelet function which is selected to be "Morlet" (aka. Gabor) function in this study and is defined as:

$$\psi(t) = \pi^{-1/4}\, e^{i\omega_0 t}\, e^{-t^2/2} \qquad \text{Eq 4.8}$$

where ω_0 is a parameter that determines the centre frequency of the wavelet and is set to be 6 in this study. The Morlet wavelet is a zero-mean function with equal variance in time and frequency, which is capable to provide a trade-off between the resolution in time and frequency while having a straightforward analytical form, which boosts computing efficiency (Silik et al., 2021). The effect of the wavelet function and its parameter can be further investigated in future studies. Additionally, if boundary effects occur in T-F representations, zero-padding can help mitigate the issue.

An example of time-frequency representation of the VBI acceleration response is presented in Figure 4.4 along with its time and frequency domain representations. The local time-varying features of the CWT representation are clearly shown, especially when the vehicle enters and leaves the bridge. The highlighted colours in this figure highlight the natural frequencies corresponding to the coupled VBI system.

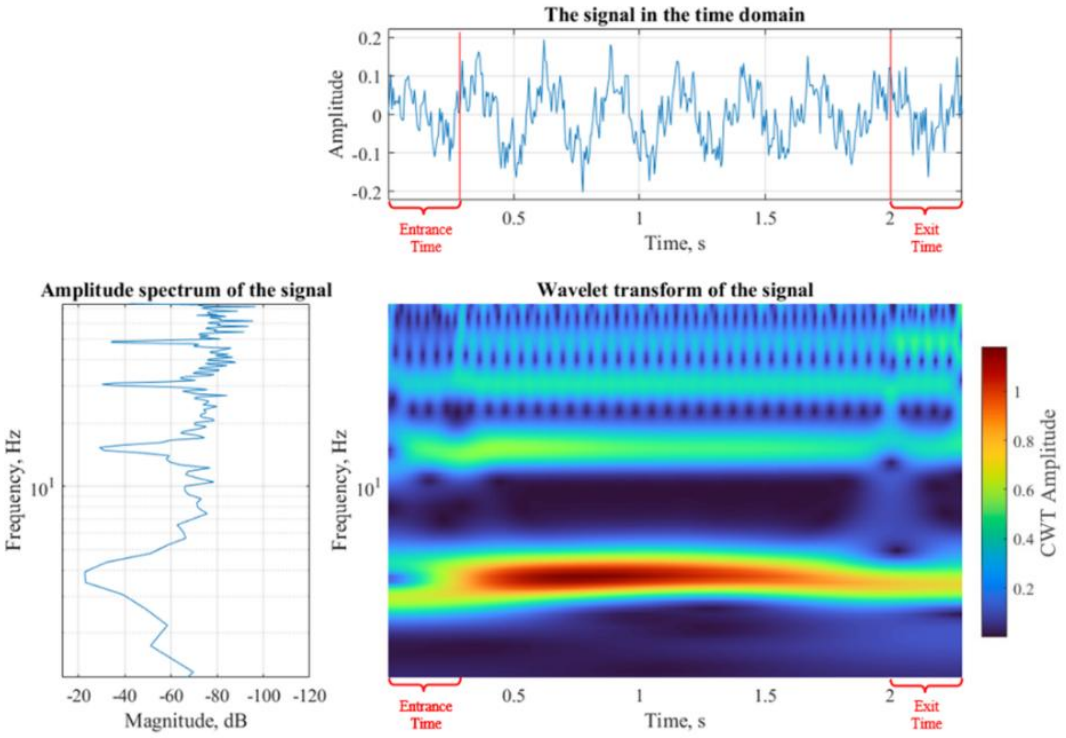


Figure 4.4 Representation of VBI acceleration response of the bridge in time, frequency, and time-frequency domains

4.6 Transfer Learning-Based Structural Damage Detection Method

Transfer Learning (TL) is an ML method introduced to transfer knowledge between two tasks. To be more specific, it aims to use the information from a source domain to perform a distinct but similar task in the target domain. The idea comes from noticing that mastering one task (source) can help in performing a similar task (target) when the source and target domains share certain similarities. (Pan & Yang, 2010). This concept can be effectively applied to structural damage detection problems. For example, when an ML model is trained for image classification, a positive transfer can enable leveraging the modelling knowledge for detecting damage in a bridge.

Fine-tuning is a parameter-based approach for inductive TL. Most ML algorithms require initial values for their training parameters (e.g. the weights and biases in DNNs). These values are usually selected at random. While choosing a reasonable initial state of an ML algorithm can enhance the training process efficiency, requiring much less training data and also preventing further issues such as anchoring in local minima (Willard et al., 2020). Fine-tuning of CNNs consists of pre-training a model utilising the source domain data and then fine-tune it using the target domain data. For fine-tuning, the hyperparameters of a set of convolutional layers are frozen, taking into account that these layers contain beneficial feature extraction information (Tan et al., 2018).

In this work, a fine-tuning strategy is utilised to leverage the knowledge of well-established pretrained CNNs for bridge damage classification. To extract features from time-frequency representations of VBI acceleration responses acquired from a single sensor situated on the bridge span, VGG-16, GoogLeNet, ResNet50, and AlexNet are investigated. The features are then classified into damage location classes using five distinguished classification algorithms, including k-Nearest Neighbour (KNN), Support Vector Machine (SVM), Fully Connected (FC) layers, Ensemble Subspace Discriminant (ESD), and Ensemble Decision Trees. The hyperparameters for each of the aforementioned classifiers are optimised for the top-performing classification. Figure 4.5 depicts a schematic representation of the proposed framework.

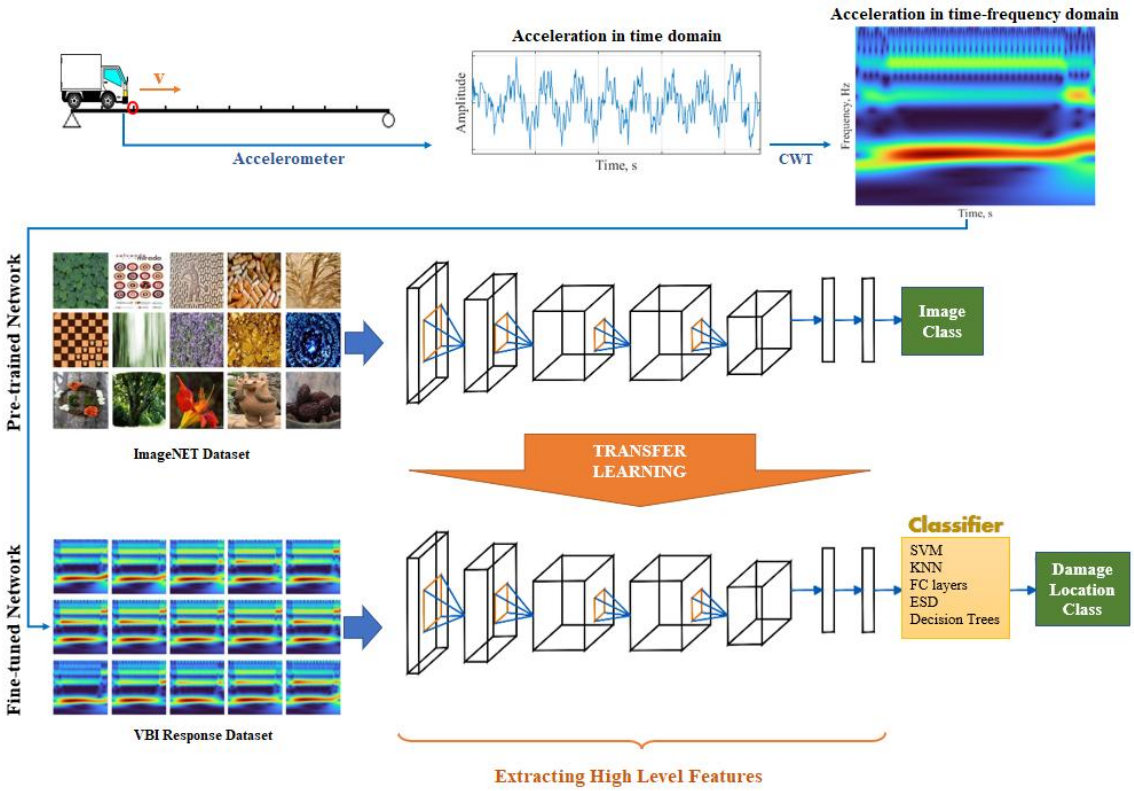


Figure 4.5 Structure of the proposed method

4.6.1 Pretrained Convolutional Neural Networks

AlexNet, developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012, was one of the first successful implementations of deep convolutional neural networks. It introduced key innovations such as the use of ReLU activation functions and dropout for regularisation. AlexNet was also the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), marking a major breakthrough in computer vision (Krizhevsky et al., 2017).

VGG16, introduced in 2014 by the Visual Geometry Group at the University of Oxford, VGG16 is a deep convolutional neural network consisting of 16 layers and trained on the ImageNet dataset. It is characterised by its use of small 3×3 convolutional kernels, which enhance the model's ability to capture detailed patterns. The architecture's considerable depth enables it to extract and represent complex features from input images. (Simonyan & Zisserman, 2014).

ResNet-50 established by Microsoft Research in 2015 is much deeper (50 layers) than VGG16. One of the key features of ResNet 50 is its use of residual connections, which

allow for the reuse of earlier feature representations in later layers and solve the vanishing gradient problem by adding shortcuts to the network (He et al., 2016).

GoogLeNet, also known as Inception-v1, was developed by Google in 2014. It was the winning model in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). GoogleNet utilises inception modules for learning more complex features (Alom et al., 2018).

4.6.2 Classification Algorithms

One common strategy for fine-tuning CNNs is to replace the last layer of the FC layers of the pre-trained model with a custom classifier to adjust the representations learned from the source domain to the target domain. This approach is applied in this study with a number of classification algorithms including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Subspace Discriminant Analysis (SDA), Fully Connected (FC) layers and Ensemble Decision Trees.

K-Nearest Neighbours (KNN) The KNN approach is one of the simplest methods for classification, which often yields competitive results. It has been used in statistical applications (Noi & Kappas, 2018; Weinberger & Saul, 2009). The basic theory behind KNN is that it searches among the dataset to find a group of k samples that are nearest to unknown samples (the distance can be defined by many metrics, such as Euclidean or Manhattan distance). The label of an unknown sample will be determined by the class attribute of the majority label of its k nearest neighbour (Akbulut et al., 2017; Wei et al., 2017). In the case where $k=1$, the method is also called nearest neighbour or lookup-table (Khoshnoudian & Talaei, 2017). A larger value of k will result in a smoother decision boundary but may also make the classifier more susceptible to noise in the data.

Support Vector Machines (SVMs) are founded on the concept of identifying the hyperplane within a high-dimensional space that optimally separates distinct classes. The SVM was developed by Cortes and Vapnik (1995). This classifier is a multivariate statistical approach, which can be applied to two or more different classes by seeking the boundaries (also called hyper-planes) with the maximum margin (the maximum difference between the nearest data). In practice, using a nonlinear transform (by means of kernel functions), SVM projects the data (support vectors) in a higher-dimensional space, where it becomes easier to find a linear boundary that can classify the data accurately. The selection of the Kernel function should be made based on the application.

This manual selection of the optimal kernel function is the most notable limitation of SVM in practice (Patle & Chouhan, 2013). Some popular Kernel functions include Gaussian, Linear, Quadratic, and Cubic.

Ensemble Subspace Discriminant (ESD) is an ML technique that combines multiple subspace discriminant analysis algorithms for improved classification performance. It was introduced as an ensemble approach for dimensionality reduction and feature extraction in high-dimensional datasets. The fundamental concept of ESD involves dividing data into multiple subspaces, conducting subspace discriminant analysis on each subspace, and then merging the results for a final decision. ESD is computationally efficient and has been employed in various classification tasks, including face recognition, text classification, and speech recognition. It provides an outstanding classification performance by employing multiple subspace discriminant analysis algorithms. Furthermore, it is capable of managing high-dimensional data by projecting it into a lower-dimensional subspace (Radhika & Varadarajan, 2018).

Fully connected classifier, also known as a multi-layer perceptron (MLP) or fully connected neural network, is a type of supervised learning model commonly used for both classification and regression tasks. It is a feedforward network consisting of multiple layers of neurons, where each node in one layer is connected to every node in the next. The model is typically trained using the backpropagation algorithm, which iteratively updates the network's weights and biases to minimise the discrepancy between predicted and actual outputs. Key hyperparameters, such as the number of hidden layers and the number of neurons per layer, play a critical role in the model's performance and must be carefully tuned. Inadequate tuning can lead to underfitting or overfitting, limiting the model's ability to generalise to new data. (Kislov & Gravirov, 2017).

Ensemble Decision Tree model aggregates the outputs of multiple individual decision trees to generate a final prediction. Each decision tree functions as a flowchart-like structure, where internal nodes represent attribute-based decisions, and leaf nodes correspond to outcomes. The topmost node is referred to as the root node. While decision trees can be applied to both classification and regression tasks (Salkhordeh et al., 2021), they are particularly effective in classification problems. Ensemble techniques such as Bagging, AdaBoost, and RUSBoost are commonly employed to enhance the accuracy and robustness of decision tree classifiers.

4.7 Case Study

In this section, the vibrational response of a simply supported concrete beam excited by a moving vehicle is presented. A FE model is set up, and coupled VBI analysis is used to generate the VBI responses. The properties of the bridge are presented in Table 4.1.

Table 4.1 Parameters of the bridge

Length of the bridge (m)	30
Flexural rigidity = EI (N·m²)	2.50e+10
mass for unit length (kg/m)	5000

A two-axle vehicle passing the bridge is shown in Figure 4.6 and the two-axle vehicle model in Law and Zhu (2004) is adopted to analyse the vehicle-bridge interaction system. The properties of the vehicle are presented in Table 4.2. The natural frequencies of the vehicle are 1.63, 2.30, 10.35, and 15.10 Hz. The weight ratio between the vehicle and the bridge is 0.135.

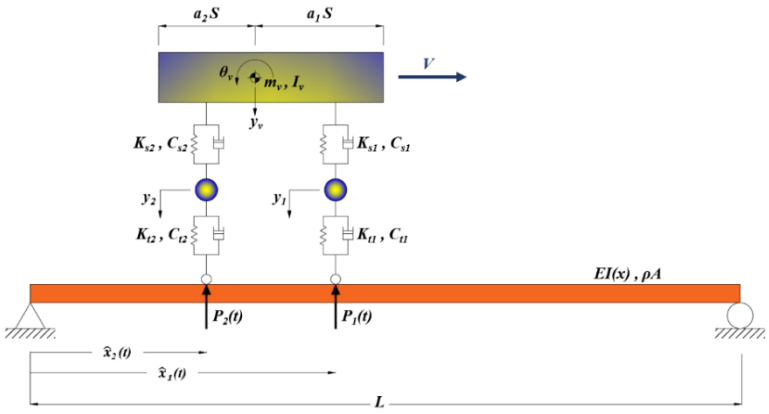


Figure 4.6 Two-axle moving vehicle-bridge system

Table 4.2 Parameters of the vehicle

Weight of the vehicle, front wheel, back wheel (kg)	m_0=17735; m_1=1500; m_2=1000
Pitching inertia of the vehicle	I_0=1.47e+5
Axle spacing (m)	S=4.27; a_1=0.519; a_2=0.481
Stiffness of the suspension (N/m)	K_{s1}=2.47e+6; K_{s2}=4.23e+6
Stiffness of the wheels (N/m)	K_{t1}=3.74e+6; K_{t2}=4.60e+6
Damping coefficient of the suspension system	C_{s1}=3.00e+4; C_{s2}=4.00e+4
Damping coefficient of the wheel	C_{t1}=3.90e+3; C_{t2}=4.30e+3

Class A road surface roughness with a range of $S_{rr}(\omega_0)$ in the range of $2 \sim 8\times10^{-6}$ m^2/(cycle/s) is selected to simulate the smooth surface of the concrete bridge.

A single transducer is assumed to be installed on $0.2L$, and the vertical acceleration of the bridge deck is used to generate damage-sensitive features. The vehicle speed is set to 15m/s which passes the 30m bridge in two seconds. Thus, the time-span of the time-frequency features is 0 to 2s with a sampling rate of 0.005s. The frequency range of the time-varying features is selected between 0 to 75 Hz, covering the first four natural frequencies of the bridge, which are 3.90, 15.61, 35.13 and 62.45 Hz.

In the VBI simulations, the vehicle is assumed to enter directly on the span without an approaching section, and no separate initial excitation is applied. This simplification isolates the vehicle-induced response.

Wahab et al. (1999) model is applied to simulate damage scenarios. This model is a stiffness reduction model that models the crack zone as azone for stiffness reduction and can simulate the crack location, depth, length and shape The parameter β of the model corresponding to the length of the damaged zone is set to 0.2, and n is set to 2. The parameter γ, which reflects the damage magnitude, covers a range of 0.1 to 0.55 with a step of 0.05 to simulate different damage severities. Parameter l_c corresponding to the midpoint location of the damage zone is used to generate eleven damage location classes $C \in \{1, 2, \ldots, 11\}$ corresponding to $l_c \in \{0, 0.1L, 0.2L, 0.3L, \ldots, 0.9L, L\}$. Figure 4.7 and Figure 4.8 show the VBI acceleration responses in time domain and time-frequency domain, respectively, for the intact structure and two damage classes.

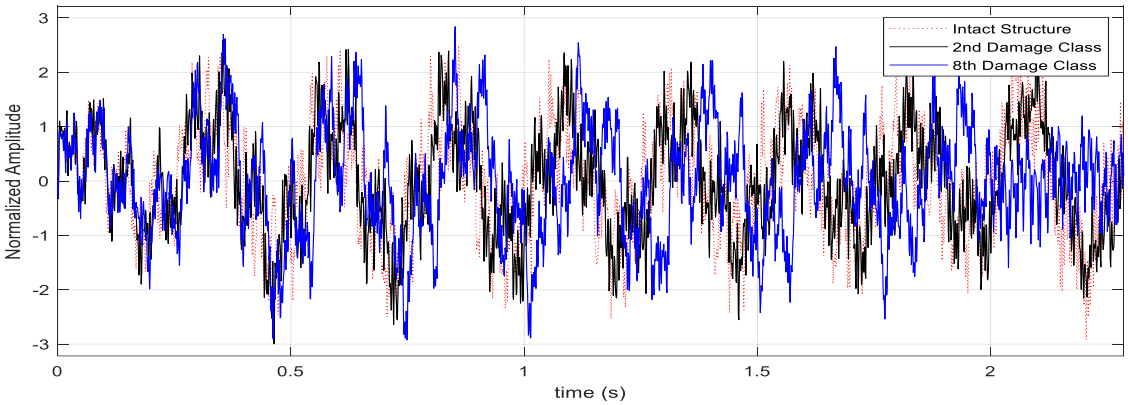
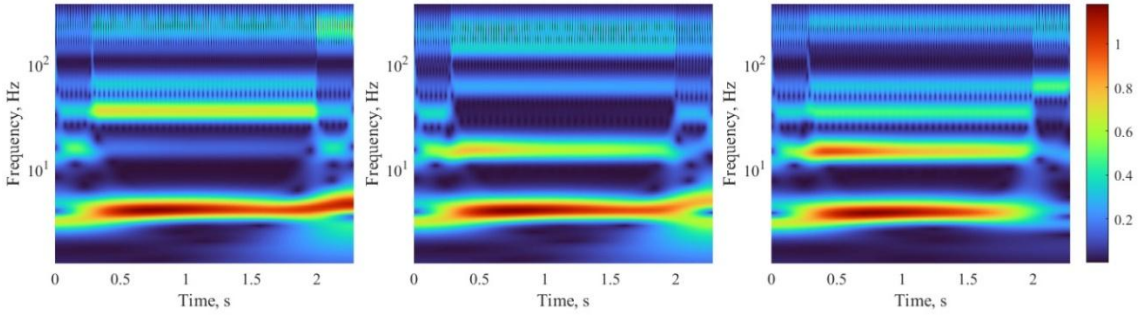


Figure 4.7 Samples of VBI acceleration response in time-domain for the Intact structure, 2nd, and 8th damage classes



(a) Intact structure (b) 2nd damage class (c) 8th damage class

Figure 4.8 Time-frequency representation of VBI acceleration responses for different damage scenarios

A dataset of colour images of CWT time-frequency representation of the VBI acceleration response corresponding to each damage location class is generated. For each class, there are 200 images in the training dataset, including 10 damage magnitudes ranging from 10% to 55%. A data augmentation by adding Gaussian noise with signal-to-noise ratio SNR=55dB to the acceleration response is applied to increase the number of samples in the training dataset. In this, the performance of different pretrained networks and classification methods is under investigation, and the effect of noise pollution will be investigated in the next step; thus, the selection of SNR=55dB can be justified for data augmentation. For generating the test dataset 100 colour images of CWT time-frequency representation of the VBI acceleration response are used, covering the damage magnitude of 14% to 50% with a step of 1.5% and augmented by adding the same amount of white Gaussian noise to the signal.

4.7.1 Damage Classification Using Transfer Learning

The training image dataset using the time-frequency representation of the VBI acceleration response corresponding to 11 damage location classes is resized to become compatible with the input sizes of the four chosen pretrained networks. After feeding the data into each pretrained network, the feature vectors extracted from their feature extraction parts are obtained. These feature vectors are the same as the input vectors of the pretrained model's final fully connected layer. The size of the feature vector differs among pretrained networks. Table 4.3 summarizes the input sizes, structure sizes, and extracted feature sizes of the investigated pretrained networks.

Table 4.3 Summary of the pretrained networks

Model	Convolutional Layers	FC Layers	Input Size	Extracted Feature Size
AlexNet	5 convolutional blocks	3	227×227×3	4096
ResNet-50	50 residual blocks	1	224×224×3	2048
VGG-16	16 convolutional blocks	3	224×224×3	4096
GoogLeNet	9 Inception Modules	1	224×224×3	1024

The extracted feature vectors are then fed as inputs to different classifiers. Each classifier is then trained to correlate the feature vectors to the corresponding damage location class and tested using the defined test dataset. Since the hyperparameters of the classifiers have a substantial influence on their performance, the MATLAB Classification Toolbox is used to optimise the hyperparameters of each classifier to achieve their best performance and obtain a fair comparison. Table 4.4 shows the search range for optimising the classifier hyperparameters and the optimal value for each.

For each of the pretrained networks, the performance of fine-tuning using different classifiers is compared by the following metrics:

Accuracy quantifies the ratio of correct predictions relative to the total number of predictions produced by a model, calculated as *(TP+TN)/Total*. While it is a straightforward and commonly used evaluation metric, it may provide a misleading impression of model performance when dealing with imbalanced class distributions.

Table 4.4 Optimised hyperparameters of the selected classifiers

Classifier	Hyperparameters	Search Range	Optimised Hyperparameter Value			
			ResNet-50	AlexNet	VGG-16	GoogLeNet
Support vector machines (SVM)	Kernel function		Linear	Linear	Linear	Linear
	Kernel scale	0.001-1000	1	1	1	1
	Box constraint level	0.001-1000	999.81	996.52	1.27	287.19
	Multiclass method		One-vs-All	One vs One	One vs One	One-vs-All
	Standardise data	Y/N	N	N	N	N
K-Nearest Neighbours (KNN)	No. of neighbors	1-1100	12	1	16	3
	Distance metrics		Cosine	Correlation	Cosine	City block
	Distance weight		Squared inverse	Squared inverse	Squared inverse	Squared inverse
	Standardise data	Y/N	N	N	N	Y
Fully Connected (FC) layers	No. of FC layers	1-3	2	2	2	2
	Activation function		ReLU	None	Tanh	None
	Lambda	5e-9 - 50	2.67e-.05	1.11E-06	1.10E-04	4.58E-09
	1st layer size	1-300	12	27	143	14
	2nd layer size	1-300	285	280	4	12
	Standardise data	Y/N	Y	Y	N	Y
Ensemble Subspace Discrimination (ESD)	No. of learners	30	30	30	30	30
	Subspace dimension	512/1024/2048	1024	2048	2048	512
Ensemble Decision Trees	Ensemble method		AdaBoost	AdaBoost	AdaBoost	AdaBoost
	Number of learners	10-500	496	95	420	441
	Learning rate	0.001-1	0.95786	8.03E-01	9.79E-01	9.88E-01
	Maximum No. of splits	1-2199	108	77	135	62

Precision quantifies the ratio of correctly predicted positive cases relative to the total number of positive predictions, given by *TP/(TP+FP)*. It reflects the model's ability to avoid false positives when identifying positive cases.

Recall, also called the true positive rate, measures how accurately the model detects actual positive cases accurately. It is calculated as *TP/(TP+FN)*, indicating the model's effectiveness in capturing all relevant positive instances.

F1 Score combines the two metrics of precision and recall into one metric using their harmonic mean (i.e., *2×Precidion×Recall/(Precision+Recall)*).

The Matthews correlation coefficient (MCC) is an all-encompassing metric for binary classification that accounts for all four elements of the confusion matrix. It is especially robust in cases of imbalanced datasets. MCC is defined as:

$$MCC = \frac{(TP * TN - FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \quad \text{Eq 4.9}$$

Although MCC was initially designed for binary classification, it can be adapted to multiclass problems by evaluating each class individually as the positive class, while treating all others as negative. The resulting scores can be averaged using macro or micro averaging techniques to obtain an overall performance measure.



Figure 4.9 comparison of classification metrics of different classifiers utilised to fine-tune each pretrained network

Figure 4.9 presents the radar graphs for comparison of classification metrics of different classification methods used to fine-tune each pretrained network. Table 4.5 and Figure 4.10 compare the MCC for damage classification using the studied pretrained networks and classifiers. The purpose of the comparison is to give a thorough assessment of each model's performance on the task and to assist in choosing the most suitable model for future applications. The findings of this comparison show that, regardless of the classifier used for fine-tuning, AlexNet's feature extraction leads to more accurate damage location categorisation, whereas GoogLeNet yields the least accurate results. It also shows that ESD pioneers all the other classifiers, whilst KNN performance rates last in the comparison.

Table 4.5 MCC comparison of damage classification

MCC	ResNet-50	AlexNet	Vgg-16	GoogLeNet
SVM	0.930	0.954	0.937	0.907
ESD	0.938	0.961	0.951	0.951
KNN	0.890	0.899	0.876	0.873
FC layers	0.933	0.948	0.928	0.888
Decision Trees	0.937	0.937	0.922	0.918

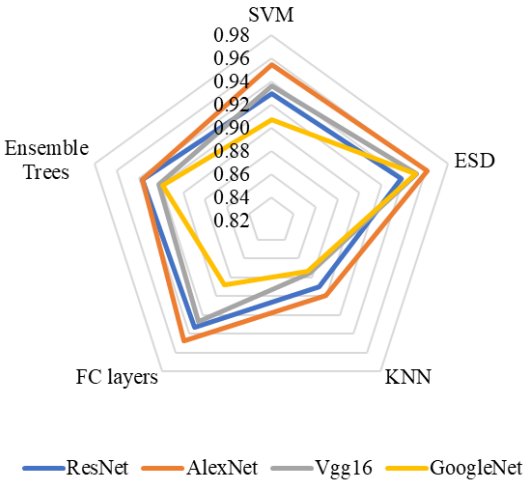


Figure 4.10 MCC comparison of damage classification

In this research, AlexNet was shown to be the best pretrained network for extracting high-level features for damage classification. Figure 4.11 depicts the ROC curves of the five studies classifiers used to identify damage location based on AlexNet generated high-level features and corresponding Areas Under the Curve (AUC), underlining the better performance of ESD as the classifier compared to other studied ones. As a result, a combination of AlexNet and ESD is selected for further studies. Using the features extracted from AlexNet to train an ESD classifier results in an MCC of 0.961. The confusion matrix of the test results of this fine-tuned network is presented in Figure 4.12.

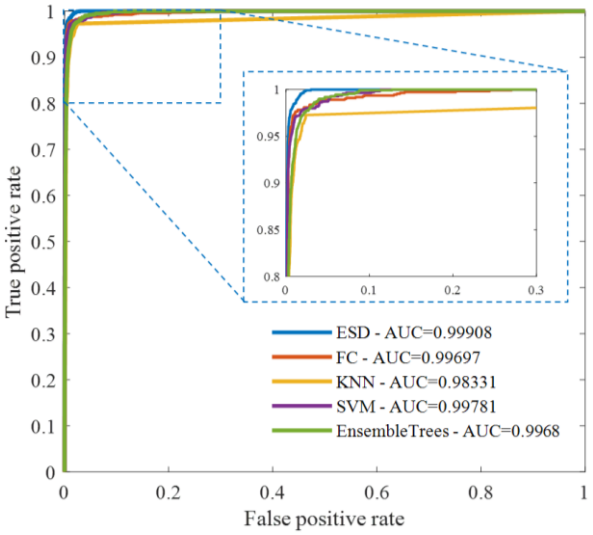


Figure 4.11 ROC curve comparison of damage classification utilising the features from AlexNet

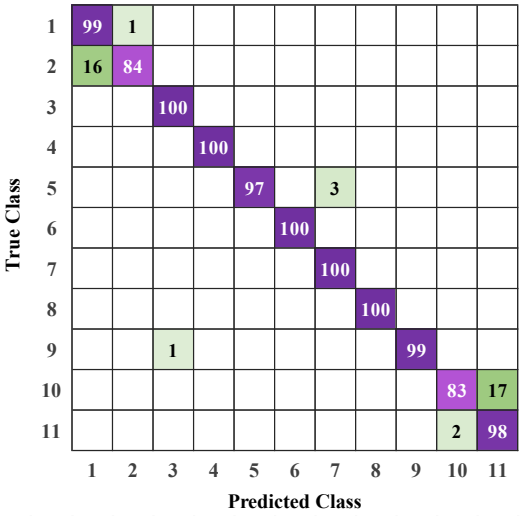


Figure 4.12 Confusion Matrix of the selected pretrained network (AlexNet+ESD)

4.7.2 Investigating the Effect of Noise in Acceleration Responses on Classification Performance

Measurement noise is unavoidable in experiments and disturbs the damage-sensitive features used for damage classification leading to misclassifying the damage, reducing the effectiveness of damage detection approach. This part is dedicated to studying the impact of noise pollution in acceleration data, as the performance of fine-tuning various pre-trained networks with different classifiers was tested independent of the noise effect. To examine the robustness of the winning combination of the pretrained network and the classifier (AlexNet and ESD), a dataset of noise polluted data with SNR of 55, 45, 40, 37 and 35dB is generated with 2200 samples for each SNR resulting a labelled dataset of 11000 images of time-frequency representation of acceleration data. Testing data include 1100 samples for each SNR value. The effect of noise pollution on classification performance is presented in Table 4.6 and Figure 4.13 The receiver operating characteristic (ROC) curve of the classification test results is also illustrated in Figure 4.14.

Figure 4.15 shows the confusion matrices of the test samples corresponding to each SNR level. It can be inferred that noise pollution with SNR of more than 40 will considerably reduce most of the classification metrics. However, the accuracy will be reduced to 0.963 in SNR 35 compared to 0.989 in SNR 55dB. A deeper examination of confusion matrices shows that a high level of noise pollution will mainly cause misclassification damages in symmetrical elements (e.g. Elements 1 and 11), and damage detection in the middle element (i.e. Element 6) will not be affected by noise pollution.

The proposed method is sensitive to the measurement noises with a SNR of lower than 40dB and the wavelet transform could be used to denoise the acceleration data and reduce the noise effect in highly noise-polluted circumstances (Pan et al., 1999). Wavelet transform denoising approach is applied to the noise-polluted acceleration data with an SNR of 35 using a “Discrete Meyer” wavelet type, and conducting 10 levels of decomposition with a threshold of 0.6 for standard deviation. The confusion matrix after denoising is presented in Figure 4.15 (f). This result showed an enhancement in classification metrics. Further investigations are needed to optimise wavelet transform parameters, such as the level and wavelet type, to enhance its noise reduction capabilities.

Table 4.6 classification metrics representing the effect of noise in acceleration data

SNR (dB)	Accuracy	Precision	Recall	F1 Score	MCC
55	0.989	0.952	0.940	0.939	0.937
45	0.986	0.937	0.924	0.923	0.920
40	0.980	0.908	0.891	0.892	0.886
37	0.967	0.643	0.832	0.524	0.518
35	0.960	0.604	0.800	0.502	0.492
35 (Denoised)	0.963	0.623	0.806	0.517	0.509

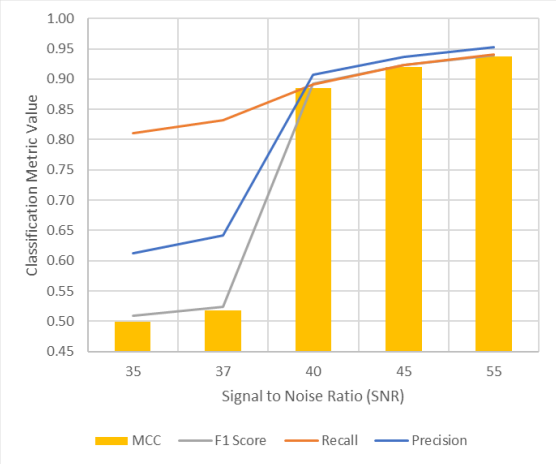


Figure 4.13 effect of noise in acceleration data on classification metrics

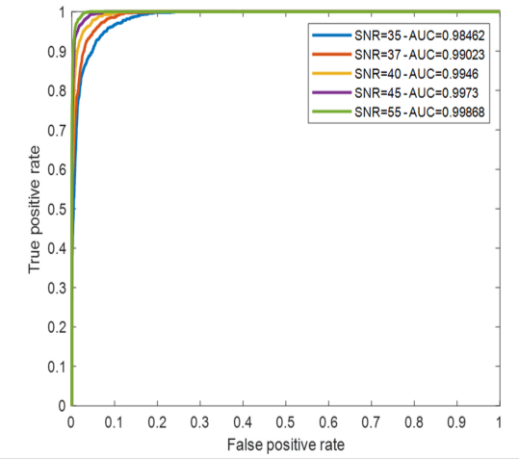
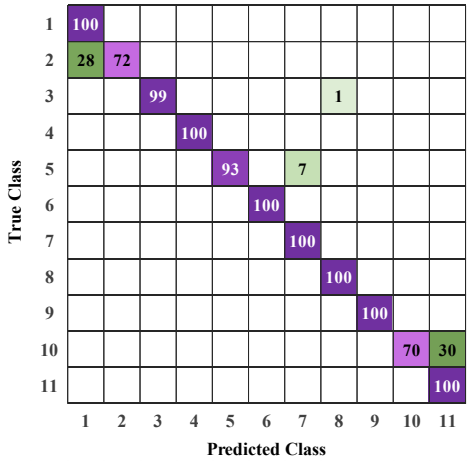
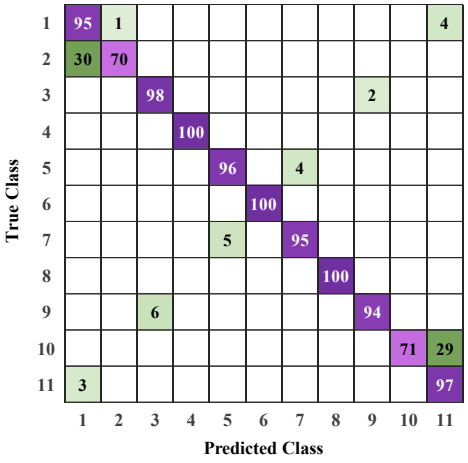


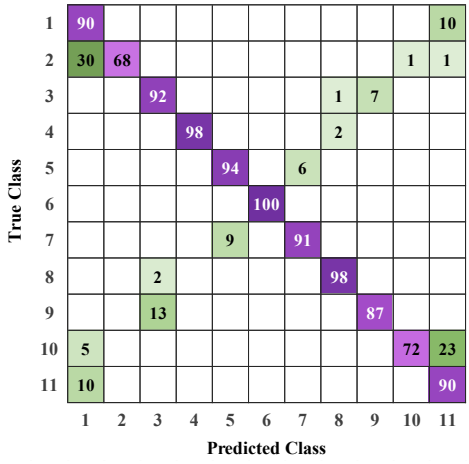
Figure 4.14 effect of noise in acceleration data on the ROC curve of the classification results



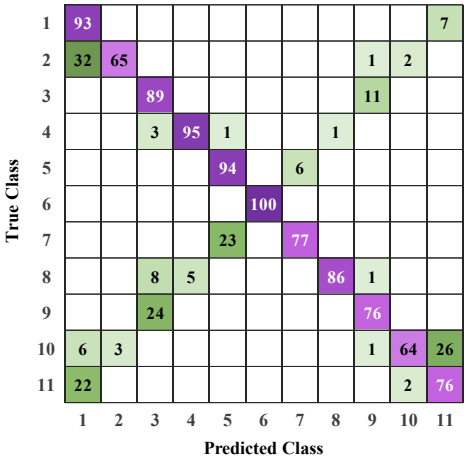
(a) SNR= 55 dB



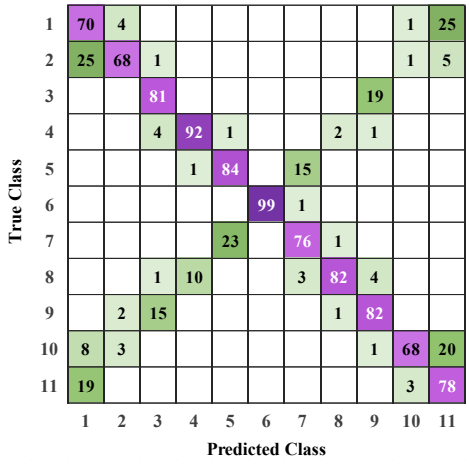
(b) SNR= 45 dB



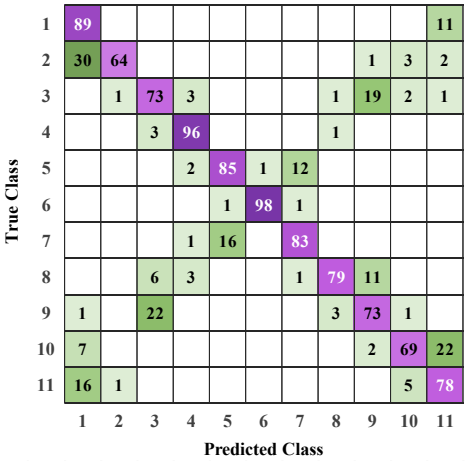
(c) SNR= 40 dB



(d) SNR= 37 dB



(e) SNR= 35 dB



(f) SNR= 35 dB (Denoised)

Figure 4.15 Confusion Matrices of the selected pretrained network (AlexNet+ESD) with different SNR in acceleration data

4.7.3 Investigating the Effect of Sensor Location on Classification Performance

VBI acceleration responses used in this study are extracted from a single transducer installed on the bridge span, which makes the proposed method more cost-effective based on the expensive installation and maintenance of transducers. On the other hand, it makes the location of the sensor worth to be examined, and investigation of the effect of sensor location on damage location classification is inevitable. In this section, five different locations from 0.1 to 0.5 of the span length of the bridge are considered as illustrated in Figure 4.16, and the performance of the proposed fine-tuned network has been investigated based on the data collected from each sensor.

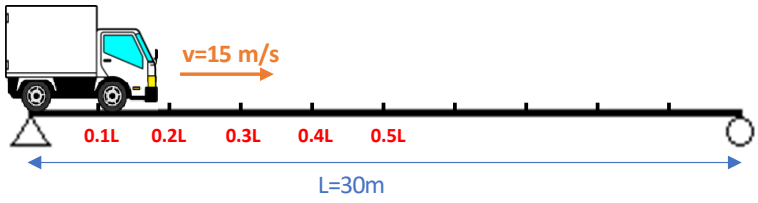


Figure 4.16 explored potential transducer installation locations on the bridge span

A separate training and test dataset is generated, and a separate but similar (AlexNet+ESD) model with the same hyperparameters is trained and tested for each vehicle speed. The classification metrics of these fine-tuned models are presented in Table 4.7 and Figure 4.17, and the ROC curves and AUCs of the classification test results are also illustrated in Figure 4.18. The results reveal that the sensor location does not have a noticeable impact on the damage location classification of the proposed method as long as the sensor is not located close to the bridge supports (< 0.1 L). The MCC is 0.859 if the sensor is placed in 0.1L, and it is between 0.956 and 0.968 if it is located elsewhere.

Table 4.7 Classification metrics representing the effect of sensor location

Sensor Location	Accuracy	Precision	Recall	F1 Score	MCC
0.1 L	0.977	0.875	0.871	0.871	0.859
0.2 L	0.993	0.967	0.964	0.963	0.961
0.3 L	0.993	0.963	0.959	0.959	0.956
0.4 L	0.995	0.972	0.970	0.970	0.968
0.5 L	0.993	0.966	0.961	0.961	0.958

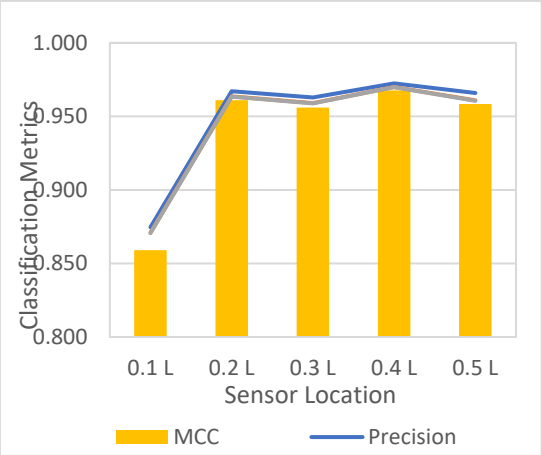


Figure 4.17 classification metrics representing the effect of sensor location

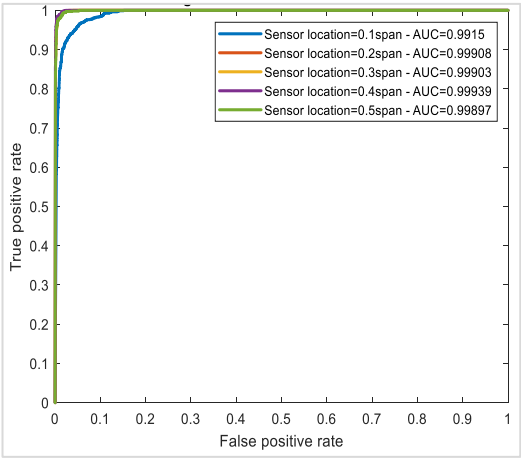


Figure 4.18 effect of sensor location on the ROC curve of the classification test results

4.7.4 Investigating the Effect of Vehicle Speed on Classification Performance

This subsection explores how the speed of a vehicle's passage over the bridge affects the suggested method's competence in classifying the location of damage. The speed range from 2.5 to 20 m/s is investigated. To have a better perspective, the images of time-varying features that are the inputs of the network's training process are presented in Figure 4.19. Because vehicle speed influences the duration of interaction, all samples were padded to obtain consistent input sizes for training.

It can be observed that higher vehicle speeds result in more visually recognisable features in the acceleration response of the bridge. This is because the magnitude of the VBI forces generated by the moving vehicle is directly proportional to its speed. Higher speeds generate larger dynamic loads, which result in larger vibrations in the beam structure. The VBI interaction forces corresponding to the front and rear wheels are illustrated in Figure 4.20. It can be observed that a rise in vehicle speed increases the VBI force magnitude. On the other hand, the time resolution of the time-varying features is higher at lower vehicle speeds. When a vehicle passes the 30m bridge span with a speed of 5 m/s, the time span of the recorded acceleration data is 6s, while it is 2s when the speed increases to 15 m/s. Moreover, the frequency resolution of the VBI contact force and, thus, the acceleration response is greater at slower vehicle speeds.

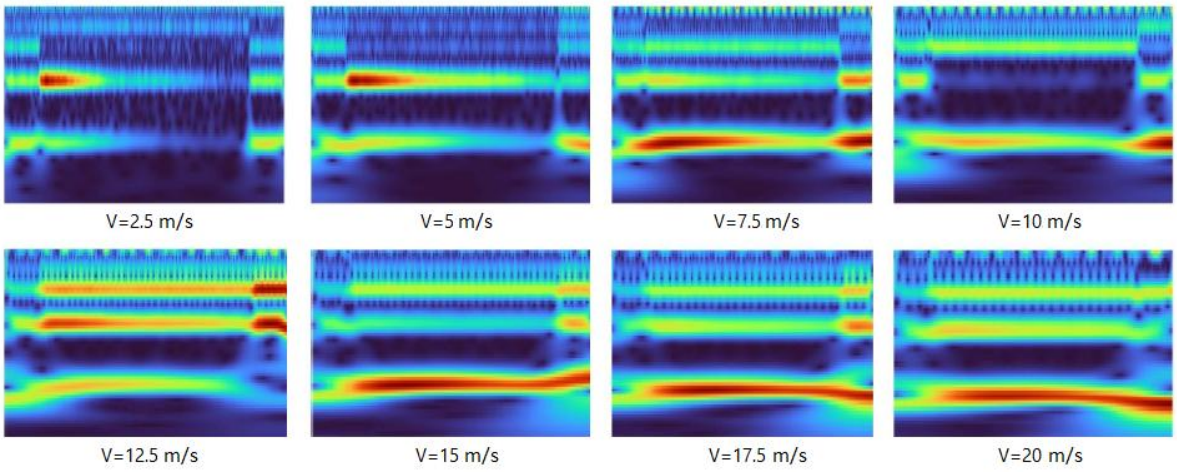


Figure 4.19 Samples of time-frequency damage-sensitive features in different vehicle speeds

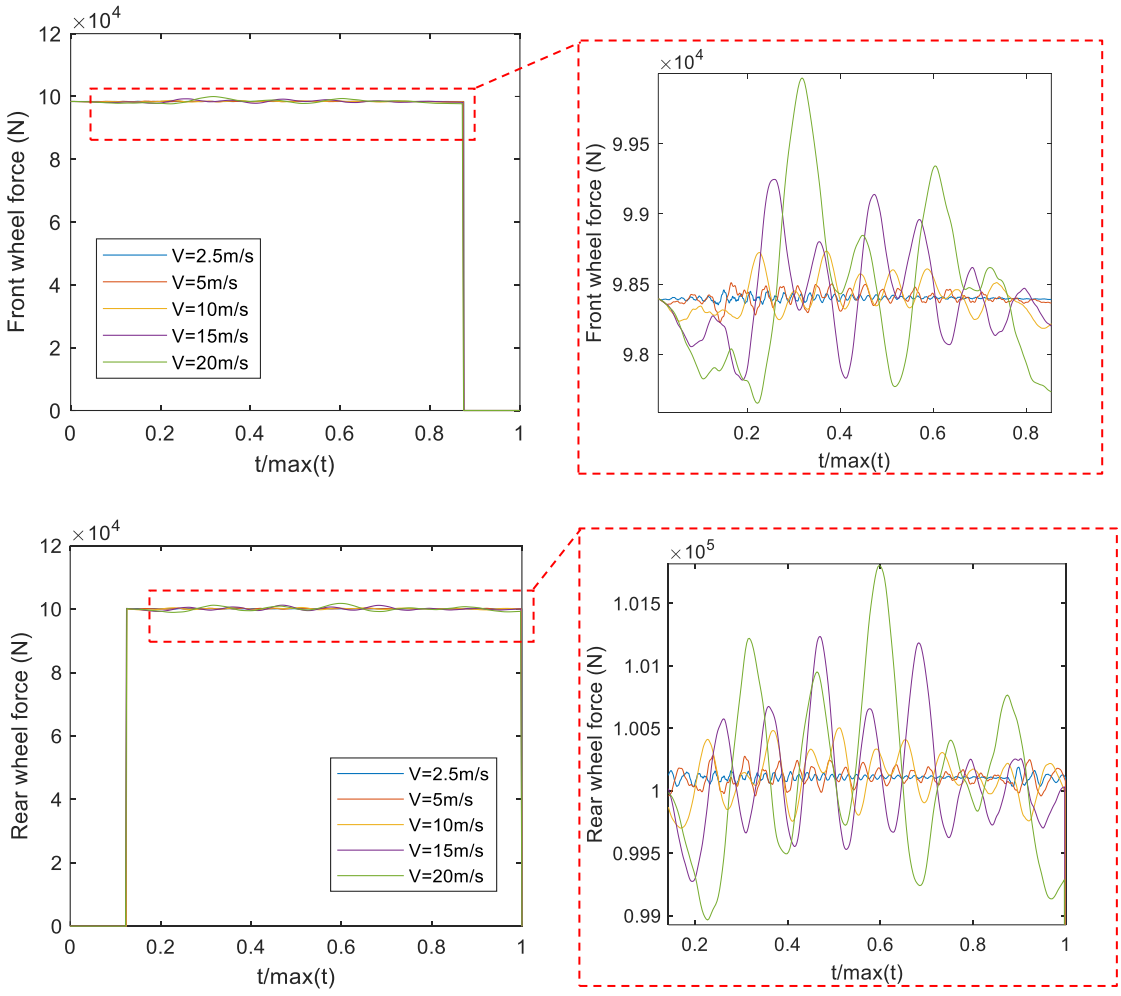


Figure 4.20 VBI interaction force of the front and rear wheels in different vehicle speeds

For each vehicle speed, a distinct dataset of labelled damage scenarios is constructed and utilised to train and test the proposed network (Alexnet+ESD). Table 4.8 and Figure 4.21 demonstrate the classification metrics of the models, while Figure 4.22 depicts the ROC curves and AUCs of the classification test outcomes. Figure 4.23 shows the confusion

matrices of the test samples for each vehicle speed level. The findings indicate that the suggested method performs better at greater vehicle speeds, as the more accentuated reaction makes it easier to identify alterations in the structural response caused by damage.

Table 4.8 Classification metrics of models trained with different vehicle speeds

Speed (m/s)	Time to pass the bridge (s)	Accuracy	Precision	Recall	F1 Score	MCC
2.5	12	0.938	0.659	0.658	0.657	0.624
5	6	0.967	0.833	0.816	0.815	0.803
7.5	4	0.985	0.920	0.917	0.917	0.910
10	3	0.985	0.920	0.916	0.916	0.909
12.5	2.4	0.984	0.917	0.910	0.909	0.903
15	2	0.993	0.967	0.964	0.963	0.961
17.5	1.7	0.993	0.969	0.965	0.965	0.963
20	1.5	0.994	0.975	0.971	0.971	0.969

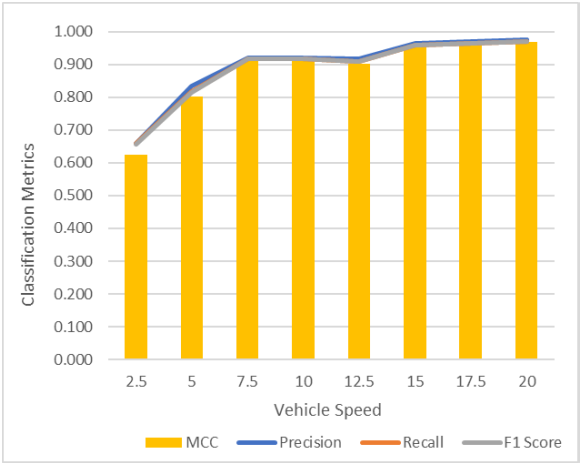


Figure 4.21 classification metrics of models trained with different vehicle speeds

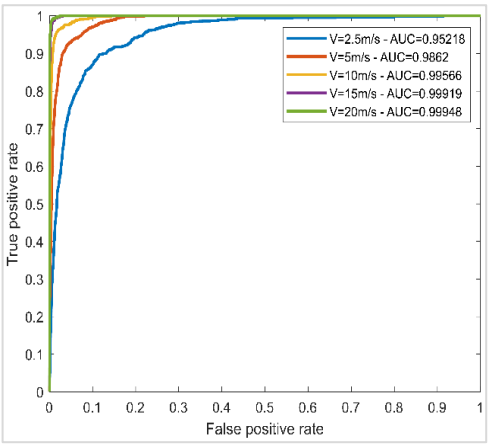
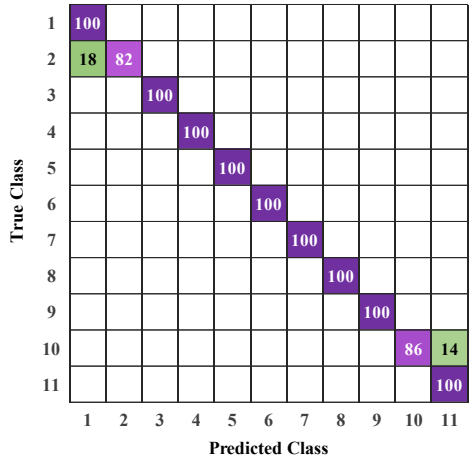
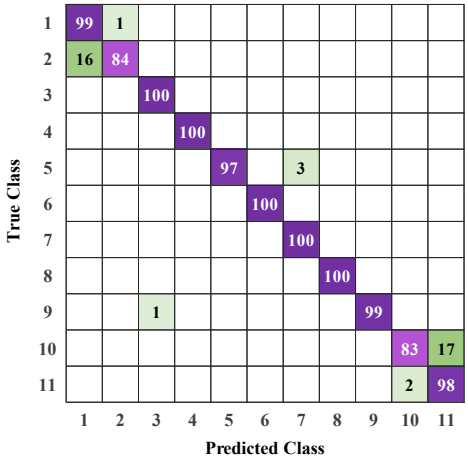


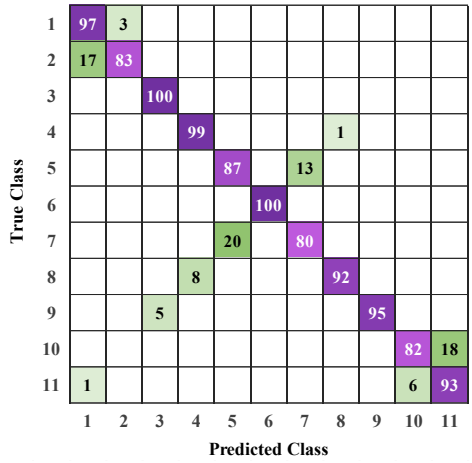
Figure 4.22 ROC curves of models trained with different vehicle speeds



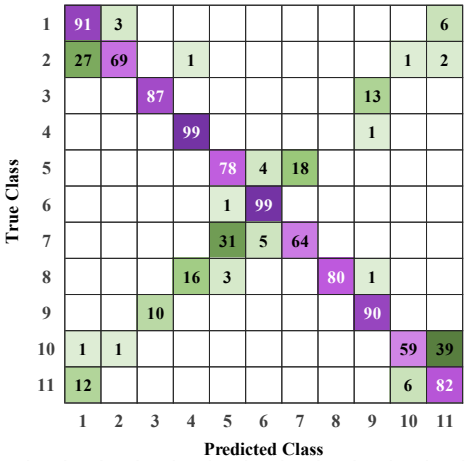
(a) V= 20 m/s



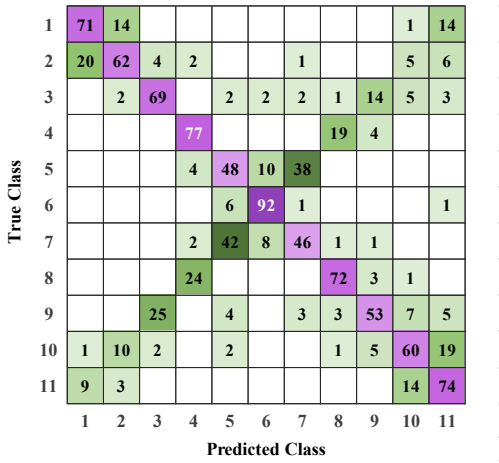
(b) V= 15 m/s



(c) V= 10 m/s



(d) V= 5 m/s



(e) V= 2.5 m/s

Figure 4.23 Confusion Matrices of the selected pretrained network trained and tested (AlexNet+ESD) with different vehicle speeds

To further assess the effectiveness of the proposed method, a global model was trained using the training datasets. with different vehicle speeds ranging from 10 to 20m/s to investigate the effect of the vehicle speed. By this global model, there is no need to predefine the vehicle speed as a parameter since a range of vehicle speeds contributes to the training dataset. The test data are selected from the data with different vehicle speeds to test the performance of the proposed method. Table 4.9 and Figure 4.24 show the classification metrics of the model, while Figure 4.25 depicts the ROC curves and AUCs of the classification test outcomes. The results show that the vehicle speed from 10 to 20 m/s does not have a large effect on the damage classification by the proposed method.

Table 4.9 Effect of vehicle speed on classification metrics

Speed (m/s)	Accuracy	Precision	Recall	F1 Score	Mcc
10	0.9815	0.905	0.898	0.898	0.890
12.5	0.9807	0.910	0.894	0.892	0.887
15	0.9902	0.958	0.946	0.945	0.944
17.5	1.0	0.960	0.948	0.947	0.946
20	0.9874	0.949	0.931	0.929	0.929

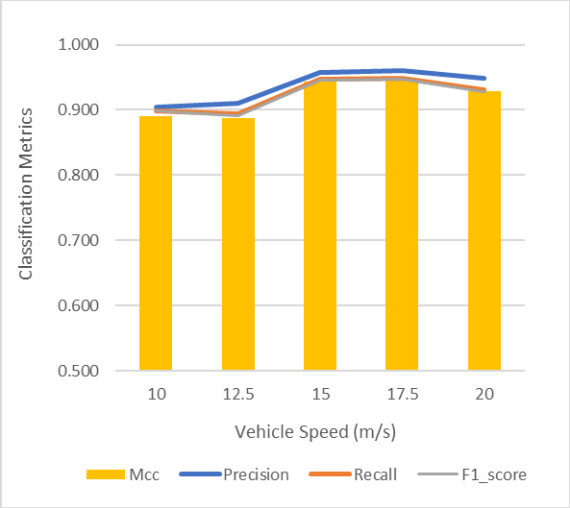


Figure 4.24 Effect of vehicle speed on classification metrics

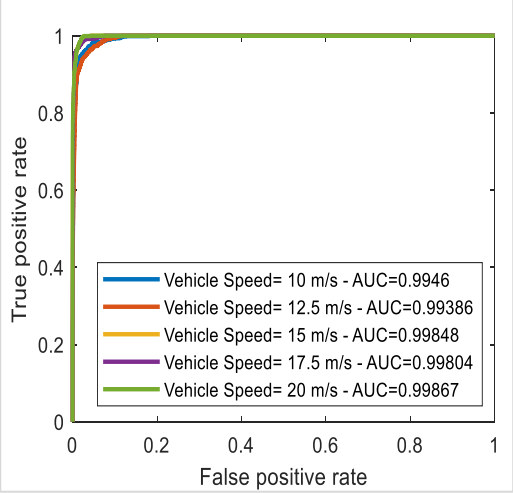


Figure 4.25 Effect of vehicle speed on the ROC curve of the classification test results

4.7.5 Investigating the Effect of Surface Roughness on Classification Performance

The dynamic interaction of the bridge and passing vehicle is directly related to the bridge surface roughness. Consequently, the effects of surface roughness should be carefully evaluated for the VBI-based bridge health monitoring (Jiantao Li et al., 2021). ISO 8608 standard classifies road surface roughness into 5 categories listed in Table 4.10 based on the range of the amplitude of the power spectral density function at the standard frequency $S_{rr}(\omega_0)$ (Tang et al., 2022).

Table 4.10 The classification of road surface roughness

Level	Range of $S_{rr}(\omega_0)$ 10^{-6} m^2/(cycle/s)	Mean of $S_{rr}(\omega_0)$ 10^{-6} m^2/(cycle/s)
Very good (A)	2 ~ 8	5
Good (B)	8 ~ 32	20
Normal (C)	32 ~ 128	80
Bad (D)	128 ~ 512	320
Very bad (E)	512 ~ 2048	1280

This section examines how surface roughness impacts the effectiveness of the proposed damage localisation method. Five labelled datasets of VBI acceleration responses are employed to train and test the proposed network (Alexnet+ESD). Table 4.11 and Figure 4.26 demonstrate the classification metrics of the models, and Figure 4.27 depicts the ROC curves and AUCs of the classification test results. The confusion matrices of the test samples are also presented in Figure 4.28. The results show that classification performance improves with increasing road surface roughness. This is attributed to rougher surfaces providing stronger excitation of the bridge system, thereby enhancing the observability of damage-sensitive features

Table 4.11 Classification metrics of models trained with different surface roughness

Surface Roughness Class	Accuracy	Precision	Recall	F1 Score	MCC
A	0.993	0.967	0.964	0.963	0.961
B	0.994	0.970	0.966	0.966	0.964
C	0.994	0.971	0.968	0.968	0.966
D	0.997	0.983	0.981	0.981	0.980
E	0.999	0.992	0.992	0.992	0.991

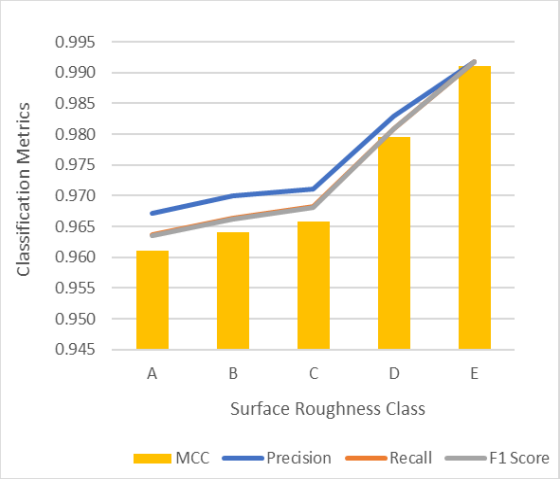


Figure 4.26 Classification metrics of models trained with different surface roughness

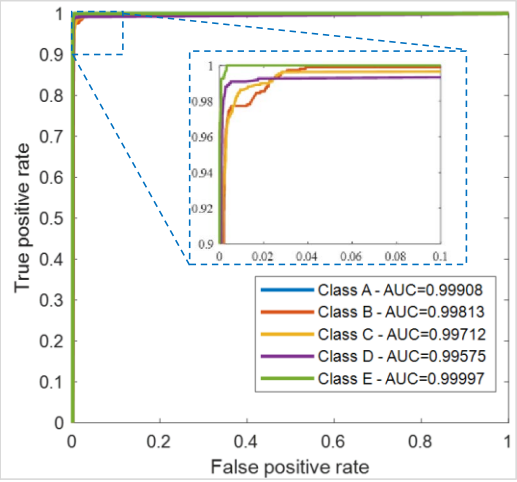
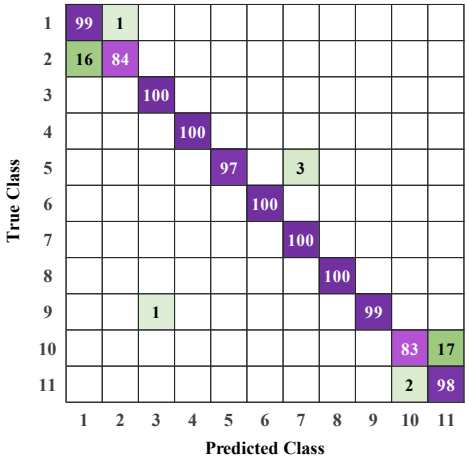
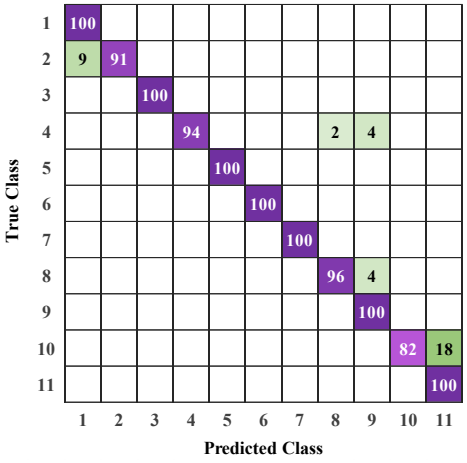


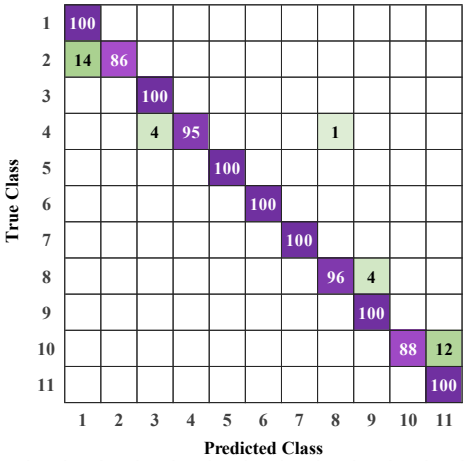
Figure 4.27 ROC curves of models trained with different surface roughness



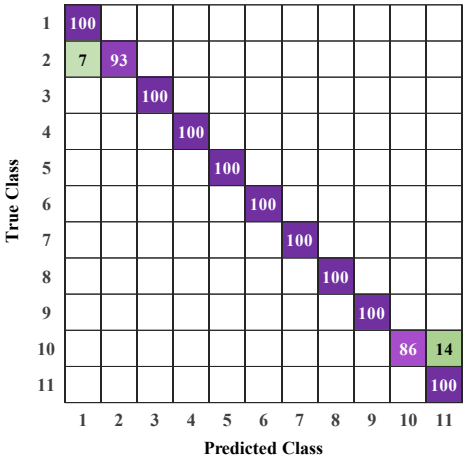
(a) Class A



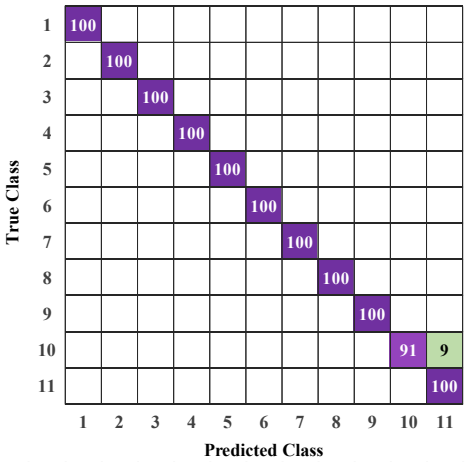
(b) Class B



(c) Class C



(d) Class D



(e) Class E

Figure 4.28 Confusion Matrices of the selected pretrained network (AlexNet+ESD) trained and tested with different surface roughness classes

To assess the effectiveness of the proposed methodology more thoroughly, a comprehensive model was trained using a dataset of various road surface roughness classes, examining the influence of surface roughness on the results. While the training dataset consists of the features corresponding to all surface roughness classes, the separate test data are selected from different classes of surface roughness to test the performance of the proposed method. Table 4.12 and Figure 4.29 demonstrate the classification metrics of the model, and Figure 4.30 depicts the ROC curves and AUCs of the classification test results. The results demonstrate that the proposed method is resilient to variations in road surface roughness when localising damage.

Table 4.12 Classification metrics representing the effect of surface roughness class

Surface Roughness Class	Accuracy	Precision	Recall	F1 Score	MCC
A	0.9631	0.8672	0.7973	0.7947	0.7976
B	0.9901	0.9545	0.9455	0.9447	0.9424
C	0.9899	0.952	0.9445	0.9439	0.941
D	0.9937	0.9703	0.9655	0.9652	0.9634
E	0.9936	0.9691	0.9645	0.9643	0.9623

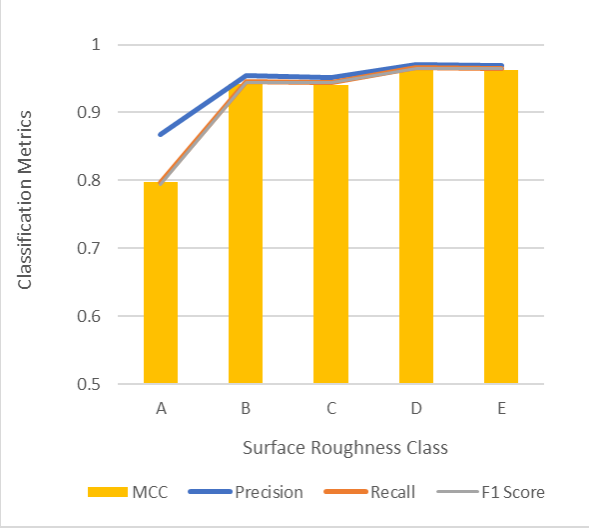


Figure 4.29 Effect of surface roughness on classification metrics

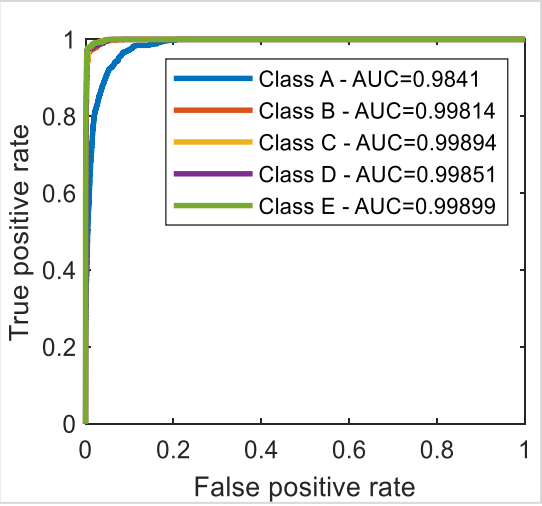


Figure 4.30 Effect of surface roughness on the ROC curve of the classification test results

4.8 Discussions

The three investigations above are discussed in the following section for a better understanding of the findings and advantages/shortcomings of the proposed approach. The robustness of the chosen fine-tuned network against various degrees of measurement noise pollution is investigated and related to its noteworthy performance in the presence of noise levels reported in commonly applicable accelerometers used for structural health monitoring (SHM). However, since only a single transducer is employed to capture vibration responses, any noise pollution with SNR less than 40dB would result in misclassifying damage in symmetric elements. When evaluating the maintenance expenses of a network of transducers, there is a trade-off between utilising a single high-quality accelerometer and a number of low-quality accelerometers. However, the proposed method may be paired with wavelet transform de-noising techniques to mitigate the impact of noise. This research also included a sensor location study, which revealed the need to situate the sensor at a distance of more than 10% of the bridge span from the supports to be less influenced by the signal disturbance caused by vehicle entry and departure from the bridge. In the next stage, the effect of the passing vehicle speed on the obtained damage features and performance of damage location classification was evaluated, reflecting that higher speeds lead to more highlighted features in time-frequency features and better classification results. Yet, raising the vehicle speed reduces the time resolution and frequency resolution of the response data. Thus, it is preferable to restrict the speed in practice to record a credible signal of VBI system acceleration response. This investigation requires thorough experimentation to determine the appropriate speed for each case study; nevertheless, speeds less than 10 m/s will not create enough VBI interaction force magnitude in this case study to attain a considerable level of precision in damage location detection and a speed of 15 to 17.5 m/s will lead into the best performance of the proposed method. At last, the effect of surface roughness on the classification performance was explored, revealing that more surface roughness will intensify the damage-sensitive features, leading to a better damage classification performance. The proposed method is much more robust to the road surface roughness in the damage classification.

4.9 Summary

This chapter develops a transfer-learning framework that fine-tunes well-known pretrained CNNs on continuous-wavelet-transform (CWT) time-frequency images of vehicle-bridge interaction (VBI) accelerations to localise damage in concrete bridges. Utilising vibrational data from a single accelerometer and employing CWT images as inputs for pretrained CNNs, reduces the measurement and training cost at the same time and provides an efficient yet accurate solution for structural health monitoring.

The key findings of this chapter are summarised as follows:

- Among all CNN + classifier combinations, CWT features extracted by AlexNet and classified with a Subspace Discriminant (ESD) achieved the highest localisation accuracy.
- The fine-tuned AlexNet + ESD pipeline remains accurate for signal-to-noise ratios up to 40 dB, and pairing it with wavelet de-noising preserves accuracy when lower-quality data are unavoidable.
- Positioning a single accelerometer at least 0.1 times the length of the bridge from each support captures the most informative damage signatures.
- Damage localisation accuracy peaks at 15-17.5 m/s, while speeds below 10 m/s produce insufficient VBI energy for reliable damage detection.
- Increased surface roughness amplifies damage-sensitive energy in the CWT images, improving the performance of the damage location classifier.

CHAPTER 5 - HYBRID DOMAIN ADAPTATION APPROACH FOR ESTIMATION OF PRESTRESSED FORCES IN PSC BRIDGES

5.1 Overview

This chapter advances TL-based bridge condition assessment by introducing a hybrid domain-adaptation approach that performs with only one label in target samples, bringing the framework closer to real-world deployment.

Reliable estimation of prestressed forces (PFs) in prestressed concrete (PSC) bridges is vital for operational performance and public safety. In this chapter, a novel approach is proposed to predict the PF of PSC bridges under moving vehicles.

The finite element model for Vehicle-Bridge Interaction (VBI) systems is established for the PSC bridge. This model is then used to generate a synthetic training dataset of the source domain. Continuous Wavelet Transform (CWT) is employed for feature extraction from VBI acceleration responses, capturing their time-frequency properties. Preliminary feature extraction is enhanced through the use of a pre-trained AlexNet network. Following this initial step, a novel hybrid domain adaptation (DA) approach is applied to close the gap between the synthetic and real-world data. Specifically, Maximum Mean Discrepancy (MMD) and adversarial DA techniques are synergistically combined. Three types of loss functions are incorporated: Regression Loss for precise force prediction, MMD Loss to align the synthetic and real-world data domains, and Adversarial Loss to ensure domain invariance. The effects of uncertainties in the VBI system, such as errors of the bridge length, flexural rigidity, density, damping, boundary conditions, errors of the vehicle model, moving speed and road surface roughness, have been discussed.

The results indicate that the integrated approach mitigates the challenges posed by domain shift, enabling reliable predictions of PFs in actual bridge structures. The finding of this study offers a comprehensive, data-driven solution with significant implications for the future of structural health monitoring and bridge condition assessment.

5.2 Related Works

Prestressed concrete (PSC) bridges, which utilise compressive stresses to counteract tensile stresses, represent a significant advancement in bridge technology (Ho et al., 2012). This prestressing technique increases bridges' service load capacity, controls crack formation, and reduces deflection, making PSC bridges both efficient and cost-effective (Yang & Huang, 2021). However, the prestress force is expected to reduce over time due to factors such as creep and shrinkage of concrete, relaxation of steel, frictional loss, and corrosion. Excessive reduction in the prestress force can severely impair a bridge's structural integrity and safety (Kim et al., 2004). Therefore, it is crucial to monitor the prestress force in PCB bridges to ensure that it remains within a safe range.

The development of non-destructive methods for estimating the prestressed force in PSC bridges has attracted the interest of researchers and engineers. Traditional local non-destructive techniques are time-consuming and costly and require special equipment (Bonopera et al., 2020). Vibration-based methods, on the other hand, have been developed and used to predict the prestress loss through changes in structural dynamic responses. Yi et al. (2019) presented a clustering number determination for output-only modal identification. Chan and Yung (2000) presented that natural frequencies of PCB bridges reduce with the increase of prestress forces due to the compression softening effect. Breccolotti (2018) stated that an increase in prestressing leads to a rise in natural frequencies at low prestressing force levels. However, as the prestressing force intensifies, the growth rate of the natural frequency decreases. Thedy et al. (2020) also verified that the vibrational frequencies of internally prestressed concrete beams are significantly influenced by the profile of the tendon and boundary conditions. Wu et al. (2021) presented a method for damage detection of tension pendulums using structural frequency variance. As noted above, variations in prestress force do not have a significant effect on the dynamic behaviour of PC bridges (Bonopera and Chang (2021), and the natural frequency is insensitive or uncertain to the change of the prestress force in PSB bridges.

Recently, the forced vibration responses have been used to estimate the prestress force through inverse calculations. Lu and Law (2006) utilised time-domain responses of the beam subjected to both sinusoidal and impact forces to estimate the prestressed force in a PSC girder, utilising the orthogonal polynomial function and Tikhonov regularisation techniques to mitigate noise effects. Other approaches that leverage the vehicle bridge

interactive (VBI) responses are highly valued in bridge condition monitoring, as moving vehicles are the primary live load on bridge structures (Zhu & Law, 2006). Law et al. (2008) introduced a wavelet-based method to identify changes in prestressing forces with wavelet techniques to identify moving forces and prestress in bridge-vehicle systems simultaneously. Li et al. (2013) utilised VBI responses to identify the elemental prestress force. A finite element model updating method based on dynamic response sensitivity was proposed to identify the prestress force, but a large error was observed due to the rough road conditions. Xiang et al. (2017) proposed a method to identify the moving load and prestressed force of a simply supported prestressed concrete beam. Their method involved transforming the prestressed force into an external pseudo-load localised in each beam element using the Virtual Distortion Method (VDM). Utilising dynamic responses of a bridge under moving vehicles for prestressed force identification offers advantages. The identification is conducted under operational conditions, yielding more reliable and accurate results for real-world applications. Moreover, compared to conventional bridge monitoring systems, the number of required sensors can be reduced significantly. Nevertheless, the effectiveness of these methods relies on how accurately the VBI model is constructed. Factors such as road surface roughness, modelling errors, the wheel-bridge contact model, and measurement noise induce uncertainties, affecting identification results (Talaei et al., 2023).

Data-driven methods, utilising the power of machine learning (ML) algorithms, present a promising avenue to address these challenges. These methods can process vast amounts of high-dimensional data to evaluate bridge conditions (Avci et al., 2021). While traditional ML techniques are heavily reliant on feature extraction, DL methods can independently identify the most useful features from raw data, making the process more efficient and effective, enhancing identification performance (Azimi & Pekcan, 2020). Yang and Huang (2021) adopted CNNs, to identify damages in prestressed concrete beam bridges. Their method combines the flexibility, curvature, and CNN to predict the damage location and extent of prestressed concrete beams. Nguyen et al. (2022) proposed an impedance-based damage detection method using deep learning. A one-dimensional CNN model is used to extract the optimal PF-sensitive features from raw impedance signals. However, the effective training of DL networks necessitates a considerable volume of high-quality labelled data corresponding to different damage states of the structure for training, which is not feasible to obtain from a real bridge structure (Azimi

& Pekcan, 2020). Still, a DL can be trained using a dataset built by an updated finite element model. The updated FE model needs to imitate the structural behaviour impeccably. Model updating techniques can be employed to reduce the domain shift between the FE model and the actual structure. However, the modelling errors can never be completely eliminated, and the results of health/condition monitoring are always dependent on the performance of the updated model.

Transfer learning methods are a great way to overcome this challenge. They allow for the sharing of knowledge gained from a FE model directly with real-world structures, eliminating the need for detailed low-level model updates. (Gardner et al., 2021). Transfer learning methods, such as Domain Adaptation (DA), can minimise the domain shift between the finite element model and the real structure. The main idea of DA is to use knowledge from a source domain, where there is plenty of labelled data, to boost learning performance in a target domain with limited labelled data. (Farahani et al., 2021). One of the prominent approaches in DA is the Maximum Mean Discrepancy (MMD) method, which aims to minimise the distribution discrepancy between the source and target domains by matching their feature means in a Reproducing Kernel Hilbert Space (RKHS) (Kumagai & Iwata, 2019). However, the application of DA in structural condition monitoring is limited so far. Gardner et al. (2021) utilised joint domain adaptation (JDA) to bridge the gap between the FE model and an experimental setup of a three-story building. They used the first three natural frequencies as damage-sensitive features and demonstrated that their approach could effectively label unknown damage states in a real structure. However, their study assumed that the same damage classes were available in both the source and target domains, which is not often the case in real-world applications. Poole et al. (2022) introduced a partial DA framework that accounts for class imbalance between the source and target domains. Their approach also utilised natural frequencies as features and is more realistic as it assumes that not all damage scenarios are labelled in the target domain. However, their method has a significant drawback, i.e. it requires time-consuming pre-processing every time a new condition appears in the structure, making it unsuitable for real-time monitoring.

5.3 Proposed Methodology

To maintain the integrity of knowledge, this section provides a detailed introduction to the fundamental concepts of the proposed method. The overall workflow of the methodology is provided in Figure 5.1.

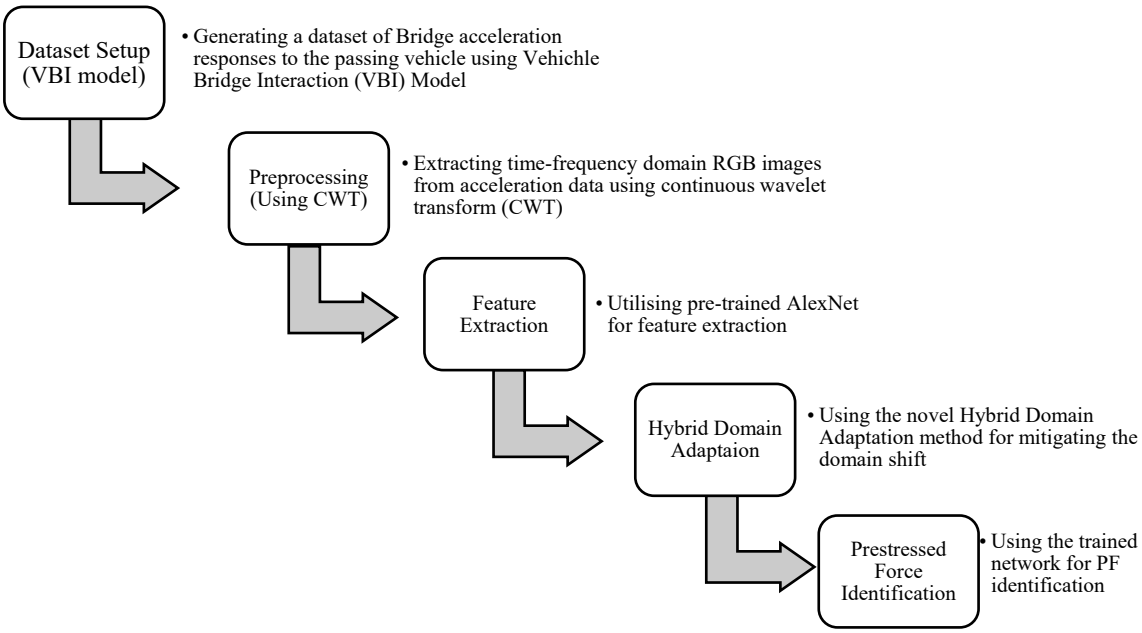


Figure 5.1 Overall methodology of the proposed approach

5.3.1 Prestressed Force Simulation in VBI Model

The VBI model utilised in this chapter is the same coupled model as that described in section 4.2.1. A simply supported bridge with a uniform cross-section is subjected to a two-axle vehicle moving at a constant speed. However, the bridge in this study is a prestressed concrete bridge (PCB). The effect of the prestressed force (PF) in the FEM of the bridge is reflected in the bridge's stiffness matrix as:

$$\bar{\boldsymbol{K}}_b = \boldsymbol{K}_b - \boldsymbol{K}_{Gb} \quad \text{Eq.5.1}$$

where K_b is the global stiffness matrix without PF. K_{Gb} is the geometric matrix due to the prestressing effect, which can be written as:

$$\boldsymbol{K}_{Gb} = \frac{P_0}{l_e}\begin{bmatrix} \frac{6}{5} & \frac{l_e}{10} & -\frac{6}{5} & \frac{l_e}{10} \\ \frac{l_e}{10} & \frac{2l_e^2}{15} & -\frac{l_e}{10} & -\frac{l_e^2}{30} \\ -\frac{6}{5} & -\frac{l_e}{10} & \frac{6}{5} & -\frac{l_e}{10} \\ \frac{l_e}{10} & -\frac{l_e^2}{30} & -\frac{l_e}{10} & -\frac{2l_e^2}{15} \end{bmatrix} \quad \text{Eq.5.2}$$

where P_0 is the PF and le is the element length (Law, 2008).

5.3.2 DL-Based PF-Sensitive Feature Extraction

In this section, a DL-based PF-sensitive feature extraction method is proposed and numerically validated. Firstly, the VBI model is utilised to generate the acceleration responses of the bridge with a range of prestressed forces. Secondly, CWT is applied to represent the acceleration data in the time-frequency domain and form the 2D-images corresponding to each PF label. This is followed by utilising a pre-trained AlexNet network for feature extraction.

5.3.2.1 Nominated VBI Model for Data Generation

In this study, a two-axle vehicle passing the bridge is adopted to analyse the VBI system. The length of the bridge is 30 m. The parameters of the bridge model are the flexural rigidity of 2.5×10^{10} N·m², the mass per unit length of 5000 kg/m and the prestress force ranging from 0 to 10×10^{6} N. The parameters of the vehicle are as (Mulcahy (1983): the mass of the vehicle, front and rear wheels are $m_v = 17735kg, m_1 = 1500kg, m_2 = 1000kg$ respectively; the pitching inertia of the vehicle body is $I_v = 1.47 \times 10^5 kg \cdot m^2$; the stiffness and damping coefficients of the suspension system for the front and rear axles are $K_{s1} = 2.47 \times 10^6 N/m$ and $C_{s1} = 3.00 \times 10^4 Nm/s$, and $K_{s2} = 4.23 \times 10^6 N/m$ and $C_{s2} = 4.00 \times 10^4 Nm/s$ respectively; the stiffness and damping coefficients of the front and rear wheels are $K_{t1} = 3.74 \times 10^6 N/m$; $K_{t2} = 4.60 \times 10^6 N/m$, and $C_{t1} = 3.90 \times 10^3 Nm/s$, $C_{t2} = 4.30 \times 10^3 Nm/s$ respectively. The axle spacing and the corresponding parameters are S=4.27m, and $a_1 = 0.519, a_2 = 0.481$ respectively.

In this study, VBI analysis is conducted with a time step of 0.001 seconds. The vehicle is set to move at a speed of 15 m/s, and the road surface roughness is categorised under Class A. For each level of PF, 20 samples are generated with a Signal to Noise Ratio (SNR) of 55 (50 for targets). The PF is labelled in percentages, ranging from 0% to 100% (10,000 KN), with increments of 5%. These settings aim to create a synthetic dataset for training a regression model.

5.3.2.2 Feature Extraction Using Pre-trained Alex-Net Network

In this study, the time-frequency representation of the acceleration responses captured by three distinct sensors located on the bridge span is obtained by CWT. These three sensors are located at 3/10L, 4/10L, and 5/10L of the bridge span, respectively.

For each sensor location, the CWT of the measured acceleration response produces a single-band grayscale image representing the signal's time-frequency characteristics. Three individual single-band images are obtained by performing the same procedure at three distinct sensor positions (3/10L, 4/10L, and 5/10L of the bridge span). These grayscale images are subsequently merged along their channel dimension, forming a composite three-band image analogous to an RGB format. As illustrated in Figure 5.2, This multi-channel representation integrates time-frequency information from three sensors, providing a more comprehensive input for subsequent feature extraction using AlexNet.

AlexNet revolutionised DL-based image recognition, opening up new possibilities in the field (Krizhevsky et al., 2017). It was among the earliest convolutional neural networks to successfully incorporate ReLU activations and Dropout to improve training stability and prevent overfitting. The selection of AlexNet over other known pre-trained networks, such as VGG16 and ResNet-50, is based on its high performance in feature extraction from CWT images of the VBI model (Talaei et al., 2023). Notably, integrating data from multiple sensors into a 3-band coloured image enriches the dataset by combining various sensor readings into a unified, informative representation. Additionally, this approach enables optimal utilisation of the pre-trained AlexNet architecture, which is inherently designed to process colored images.

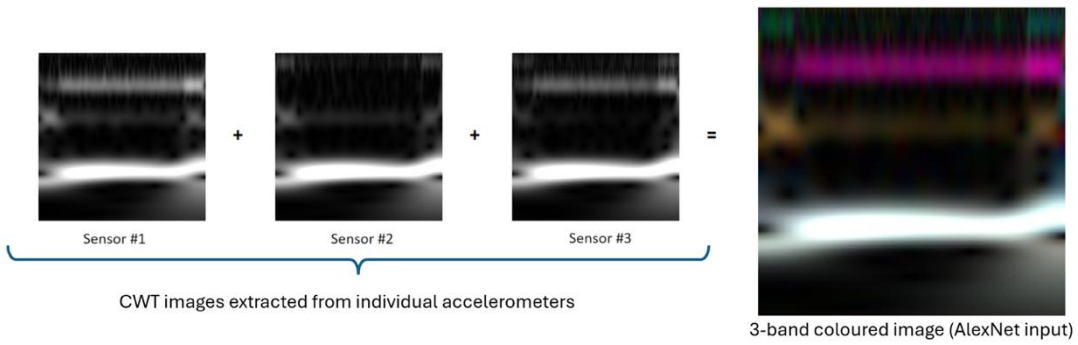


Figure 5.2 Composing 3-band colored image by combining CWT data from three sensors

5.3.2.3 Performance of the AlexNet for Predicting PF

The flowchart to evaluate the performance of the AlexNet model for predicting PFs is presented in Figure 5.3. As demonstrated, the acceleration responses at three measurement locations of the bridge subjected to a moving vehicle are utilised to obtain the time-frequency representation by CWT, and then they are combined to form the image as the input of the AlexNet. The features extracted by the pre-trained AlexNet are used to

predict the PF using a support vector regression (SVR). Figure 5.4(a) shows the first three principal components of the extracted features and the correlation between the features and PFs. As shown in Figure 5.4(a), there is a continuous and gradual relationship between the feature vector extracted from the AlexNet network and PF. Including a regression layer allows the network's output to predict the PF. The results in Figure 5.4(b) show that the predicted prestress force agrees well with the true value.

As mentioned above, both the training and testing datasets are generated using the numerical model. There is no domain discrepancy, and this is not the case in practice. In practice, the training dataset is generated using the FE model, but the test data are measured from the real bridge structure. The testing data may have significant discrepancies from the training dataset due to the effect of the operational environment. This domain shift will significantly affect the performance of the AlexNet model. To overcome this issue, a novel DA method is introduced in the next section.

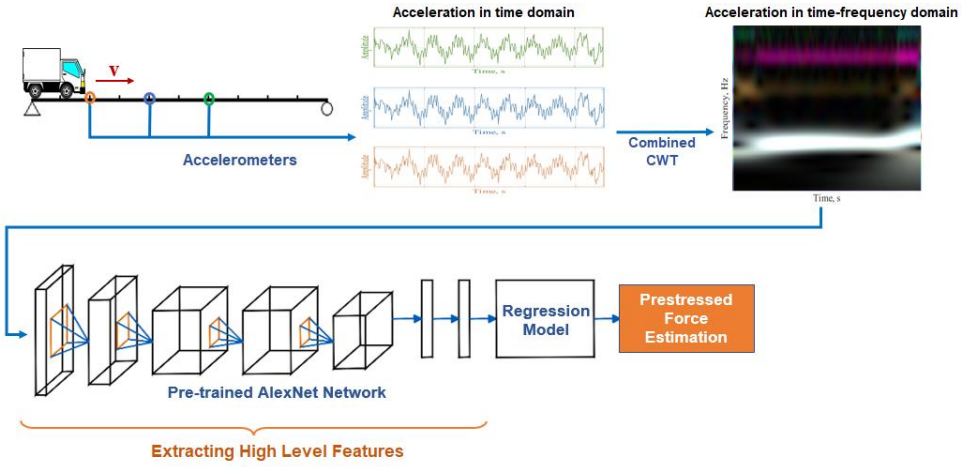
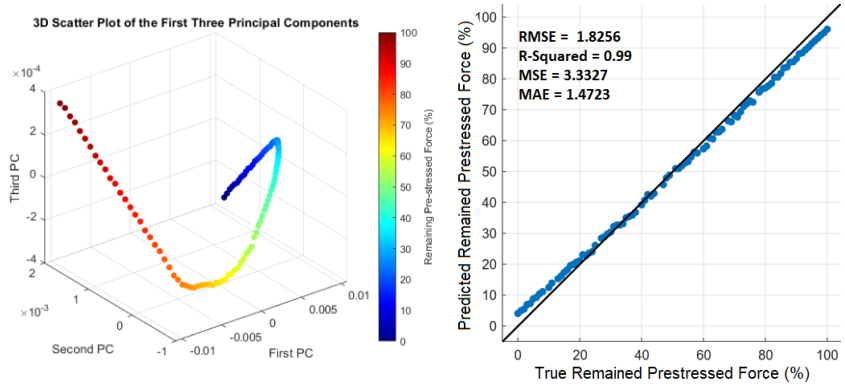


Figure 5.3 The flowchart for evaluating the performance of the AlexNet model



(a) First three PCs of Extracted Features (b) Correlation of features to PFs using SVR

Figure 5.4 Feature extraction using AlexNet and correlation of features to PF

5.4 A Novel Hybrid Domain Adaptation Method for PF Prediction

5.4.1 Domain Adaptation (DA)

Transfer Learning (TL) is to use the knowledge learned from a source domain to conduct a different task in the target domain. A *Domain* that can be represented as $D = \{X, P(x)\}$ is consist of a feature space *X* and a marginal probability distribution *P(x)*, where $x \in X$. A *Task* that can be represented as $T = \{Y, f(x)\}$ is consist of a label set *Y* and a classifier *f(x)*, and $f(x) = P(y|x)$ is the conditional probability distribution where $y \in Y$.

Assuming a labelled source domain $D_s = \{(x_1, y_1), \dots, (x_{ns}, y_{ns})\}$ and an unlabelled target domain $D_t = \{x_{ns+1}, \dots, x_{ns+nt}\}$ with respect to the assumptions that $X_s = X_t, Y_s = Y_t$. The idea is that learning a task (source) can facilitate to perform a similar task (target) when the source and target domains have certain similarities (Pan & Yang, 2010). The concept can be used for structural health monitoring problems. For example, when the data from an FE model are used to train an ML model, the modelling knowledge can be leveraged for a real bridge despite variations in different structural properties, resulting in a domain shift between the structural health indicator features of the FE model and the real structure.

DA is a technique for transductive TL that seeks to transfer the knowledge of feature representation. With $P_s(x_s) \neq P_t(x_t)$, $P_s(y_s|x_s) \neq P_t(y_t|x_t)$, DA technique aims to learn a feature representation in which the distribution differences are visibly reduced between:

a) marginal distributions: $P_s(x_s)$ and $P_t(x_t)$;

b) class-conditional distributions: $P_s(y_s|x_s)$ and $P_t(y_t|x_t)$.

DA maps the source and target data in a representation feature space by minimising the differences between their distributions while maintaining the primary structure of the original source and target data (Rashidi et al., 2022). If a regression or classification task can be done in the source feature space, it can also be done in the transformed feature space. In this way, the regression or classification knowledge can be transferred into the target domain and the target unlabelled data can find their own labels (Guan & Liu, 2021).

5.4.2 A Novel Hybrid DA-Based Method

Figure 5.5 shows the framework of the proposed hybrid DA method. Two DA techniques, e.g. discrepancy-based DA and adversarial-based DA, are synergistically integrated in this study. **Discrepancy-based DA:** In this method, the discrepancy between marginal and conditional distributions is reduced by minimising the Maximum mean discrepancy (MMD) objective functions (Kumagai & Iwata, 2019). **Adversarial-based DA:** The cross-domain discrepancy is minimised using an adversarial objective function (Ganin & Lempitsky, 2015). The original idea was inspired by Goodfellow et al. (2014), introducing Generative Adversarial Networks (GAN). In this method, a domain discriminator and a feature extractor are used simultaneously. The domain discriminator is responsible for determining whether input data is related to the source or target domain, and a domain confusion loss is defined for that purpose. The feature extractor is meant to extract features from data to perform a better classification (or regression) and instantaneously deceive the domain discriminator, which means the source and target domain features could not be recognised (i.e. extracting domain-invariant features). The feature extractor trains to minimise classification (or regression) loss for labelled source samples and maximise the domain confusion loss for all the samples at the same time (Li et al., 2021).

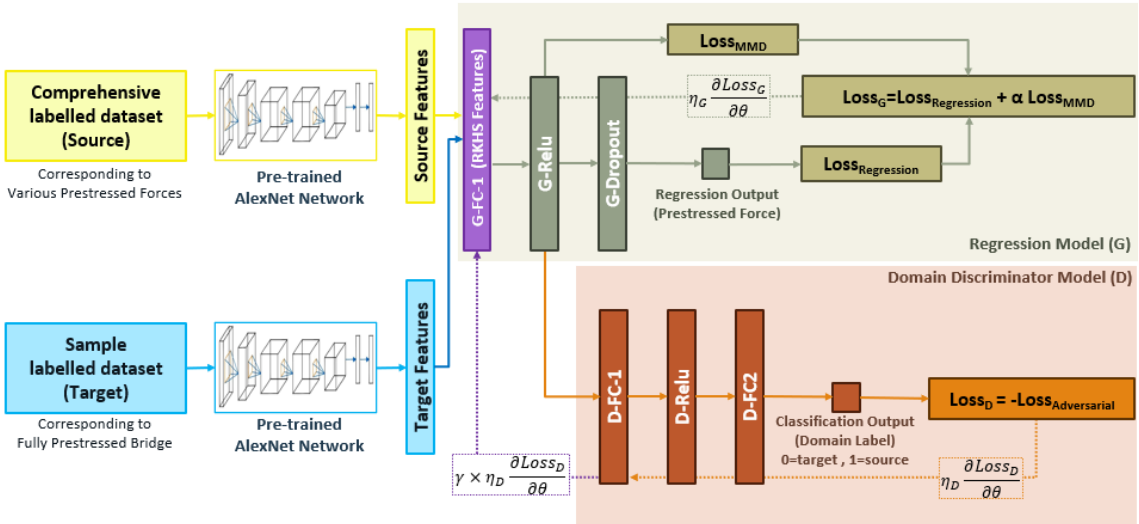


Figure 5.5 Framework of the proposed hybrid DA method

To implant both Maximum Mean Discrepancy (MMD) loss and adversarial loss in a DA-based regression model, three types of loss functions are used:

1. Regression Loss: is a Mean Squared Error (MSE) loss computed between the predicted PF and the actual force.

$$\mathcal{L}_{regression} = mean\left((Y_{predicted} - Y_{actual})^2\right) \quad \text{Eq.5.3}$$

2. Maximum Mean Discrepancy (MMD) Loss: is employed to align the source and target data distributions within a Reproducing Kernel Hilbert Space (RKHS). Its primary aim is to minimise the domain discrepancy for data corresponding to label A, which in our case represents a fully-prestressed bridge with 100% PF. This label is particularly significant as it is the only label present in the target domain.

$$\mathcal{L}_{MMD} = MMD\left(F_s(X_s|Y_s = A), F_t(X_t|Y_t = A)\right) \quad \text{Eq.5.4}$$

where F_s and F_t are feature mapping functions for the source and target domains, respectively. Considering $f_s(x_i)$ as the feature representation of the *i*th source sample and $f_t(x_j)$ as the feature representation of the *j*th target sample, the squared MMD loss is calculated as follows:

$$\text{MMD}^2(f_s, f_t) = \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} k\left(f_s(x_i), f_s(x_j)\right) + \frac{1}{m(m-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{m} k\left(f_t(x_i), f_t(x_j)\right) - \frac{2}{n \times m} \sum_{i=1}^{n} \sum_{j=1}^{m} k\left(f_s(x_i), f_t(x_j)\right) \quad \text{Eq.5.5}$$

where *n* is the number of samples in the source domain, *m* is the number of samples in the target domain, and *k(x, y)* is the Gaussian (RBF) kernel function, defined as:

$$k(x, y) = e^{\frac{-||x-y||^2}{2\sigma^2}} \quad \text{Eq.5.6}$$

where σ is a parameter controlling the width of the kernel.

3. Adversarial Loss: aims to make the final feature space domain-insensitive. It contains two parts for source and target features:

$$\mathcal{L}_{Adversarial} = mean\left(-log\left(D(F_s)\right)\right) + mean\left(-log\left(1 - D(F_t)\right)\right) \quad \text{Eq.5.7}$$

where D denotes the discriminator output. Unlike MMD, adversarial loss does not require choosing a kernel or its parameters.

Training processes for the regression model (G) and the discriminator model (D) are integrated into a unified optimisation scheme. For network (G), the total loss is a composite of the regression loss and the Maximum Mean Discrepancy (MMD) loss, while the loss function for network (D) is defined as the negative of the adversarial loss:

$$\mathcal{L}_G = \mathcal{L}_{Regression} + \alpha \mathcal{L}_{MMD} \quad \text{Eq.5.8}$$

$$\mathcal{L}_D = -\mathcal{L}_{Adversarial} \quad \text{Eq.5.9}$$

These two losses can be complementary. While MMD ensures that the source and target distributions are close in an RKHS, adversarial training ensures that a discriminator cannot easily distinguish between the source and target domains. Loss_G is backpropagated through the regression model with a learning rate of η_G, while Loss_D is backpropagated through the discriminator model with a learning rate of η_D. The first fully connected layer is shared between the two models and is updated with a learning rate of $\eta_G + \gamma \times \eta_D$. This shared layer is designed to capture domain-invariant features in RKHS. The optimisation process simultaneously minimises the regression loss, the Maximum Mean Discrepancy (MMD) loss, and maximises the adversarial loss. This enables the network to learn features that are robust across different domains using a combination of two powerful DA methods.

It should be mentioned that training the hybrid domain adaptation model can be computationally demanding. However, this training process is performed offline. Once the model is trained, the estimation of the prestress force can be performed online in real-time in practice.

5.5 Evaluation of the Proposed Method

To evaluate the proposed method, the VBI model described in Section 3.1 is employed to generate the fully labelled source domain dataset. Utilising the proposed hybrid DA technique, PF estimation knowledge is transferred from this fully characterised source domain to a variety of target domains with limited data availability. This section explores how different types of modelling errors and domain shifts affect the hybrid DA model's performance and demonstrates the robustness of the proposed method.

5.5.1 Training Data Setup

In the source domain, a dataset of 420 samples is generated for training, including 21 labels representing PF levels ranging from 0% to 100% with an interval of 5%. Each label comprises 20 samples, augmented by introducing a 55 dB SNR white Gaussian noise to the VBI acceleration responses. In the target domain, the only label available in the training dataset is the fully-prestressed condition (Y=100% PF), as acquiring data for different PF levels of the bridge in real-life scenarios is impractical. For each target

domain, 10 samples for 100% PF label were augmented using a 45 dB SNR white Gaussian noise.

5.5.2 Model Architecture and Training Process

The model's architecture comprises two fully connected layers for the generator (G) and discriminator (D), initially configured with 50 and 5 neurons, respectively. The training process is governed by a batch size of 16, with learning rates set at 0.5e-3 for the generator (ηG) and 1e-6 for the discriminator (ηD), conducted over 20,000 epochs. For consistency, the model structure is kept similar for training the DA for all the target domains. During this process, domain-invariant features are extracted from PF-sensitive features, resulting in features that are both PF-sensitive and domain-invariant simultaneously.

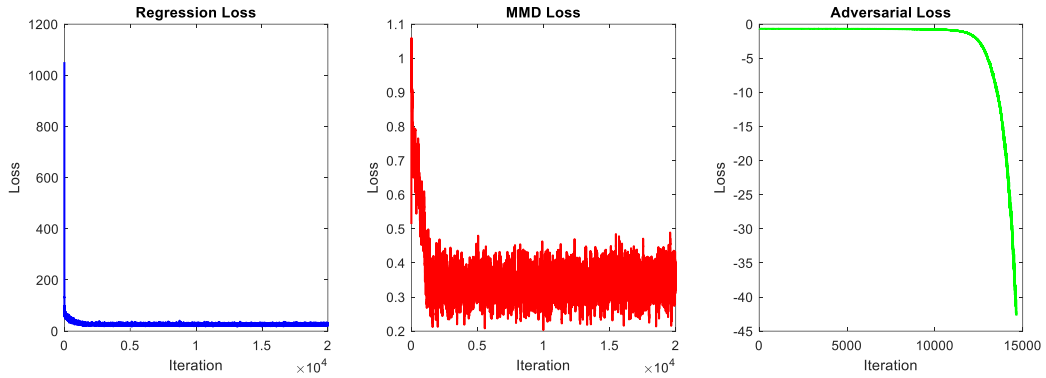


Figure 5.6 Evolution of loss values during training the proposed hybrid DA model

During the training phase, all labelled source domain data are utilised to compute the regression loss and predict the PF, while the MMD loss and adversarial loss are calculated using data corresponding to the fully-prestressed condition (Y=100% PF) from both the source and target domains. The loss values over the training process for a sample target are illustrated in Figure 5.6.

It is notable that during the training process, where the adversarial loss trends towards negative infinity indicates that the discriminator has become excessively proficient. Under such conditions, the discriminator fails to provide meaningful gradient information for the generator's updates. To counteract this imbalance, a safeguard mechanism is employed: when the discriminator's loss ($Loss_D$), falls below a threshold of one, its learning rate is linearly reduced to zero. This effectively 'freezes' the discriminator's parameters, preventing further updates and thereby stabilizing the adversarial game. This

strategy aims to restore a dynamic balance between the generator and the discriminator, ensuring that both can continue to learn effectively.

As a result of optimising the MMD and adversarial loss, the feature distribution discrepancy between the source and target domains is reduced, minimizing the effect of domain shifts in X_t and X_s for the fully-prestressed force condition. The kernelized stein discrepancy (KSD) of the source and target domain representative features corresponding to fully-prestressed condition (Y=100% PF) before and after DA is presented in Figure 5.7.

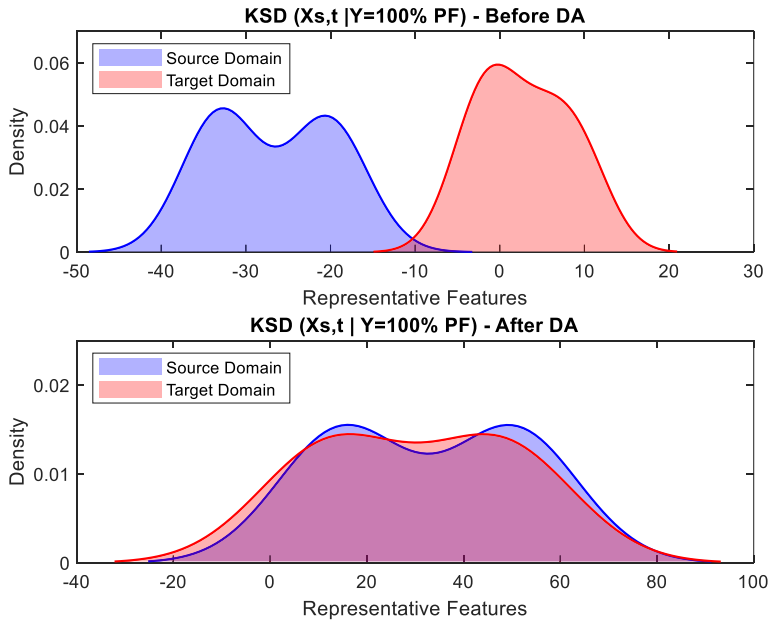


Figure 5.7 Feature distribution of source and target domain representative features at 100% PF condition before and after hybrid DA

5.5.3 Testing and Evaluation

Table 5.1 Definition of different target domains for VBI systems

Category	Parameters	Source Domain	Target Domains								
			Deviations from the Source Domain								
			#1	#2	#3	#4	#5	#6	#7	#8	#9
Bridge	Length	30 m	3%	-	-	-	-	-	-	-	3%
	Flexural rigidity (EI)	2.5E+10 Nm^2	-	-12%	-	-	-	-	-	-	-12%
	Mass for unit length	5000 Kg/m	-	-	-4%	-	-	-	-	-	-4%
	Damping ratio (ξ)	0	-	-	-	0.02	-	-	-	-	0.02
	Boundary conditions	P.P.	-	-	-	-	F.P.	-	-	-	F.P.
Road	Roughness class	A	-	-	-	-	-	B			B
Vehicle	Speed	15 m/s	-	-	-	-	-	-	20%	-	20%
	Parameters	A	-	-	-	-	-	-	-	B	B

In this section, the target domain data for unseen PF conditions (35% to 95% PF levels) are studied for testing the performance of the hybrid DA. In this study, the typical values are taken as examples to study the effect of modelling errors on the domain shift and the performance of the proposed hybrid model using different target domains. These target domains are divided into three sub-categories: the bridge and vehicle models, and the road conditions. A detailed breakdown of the target domains is presented in Table 5.1.

5.5.3.1 Effect of Bridge Modelling Errors

Modelling discrepancies, including incorrect lengths, variations in flexural rigidity (EI), mass per unit length and damping ratios, and different boundary conditions, are significant sources of domain shift. These modelling discrepancies correspond to Targets #1-#5 in Table 1.

Figure 5.8 shows the effect of these variables by illustrating the distribution of representative features for each target. Figure 5.9 to Figure 5.13 show the tSNE representation of the source and target domain features before and after applying hybrid

DA for Targets #1-#5, and their PF estimation results for the target domain test data. From the results, a 3% increase in length (Target #1) results in the minimal domain shift. In contrast, a 0.02 increase in damping ratio (Target #4) and a 4% decrease in mass (Target #3) induce more substantial shifts. The alteration of boundary conditions from pinned-pinned to fixed-pinned (Target #5) causes a notable domain shift. The most significant effect is observed with a 12% decrease in EI (Target #2), which dramatically transforms the feature distribution.

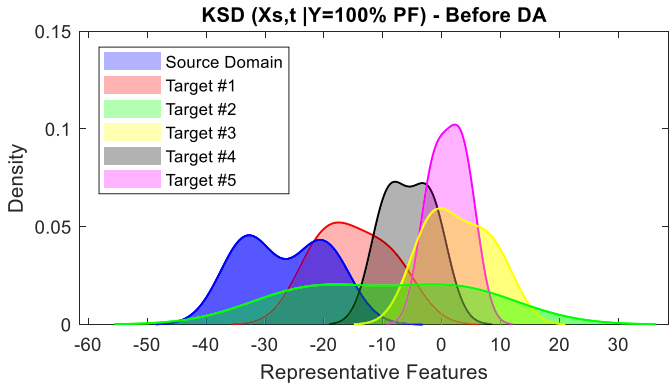


Figure 5.8 Feature distribution of target domain representative features for the targets with bridge modelling error at 100% PF condition

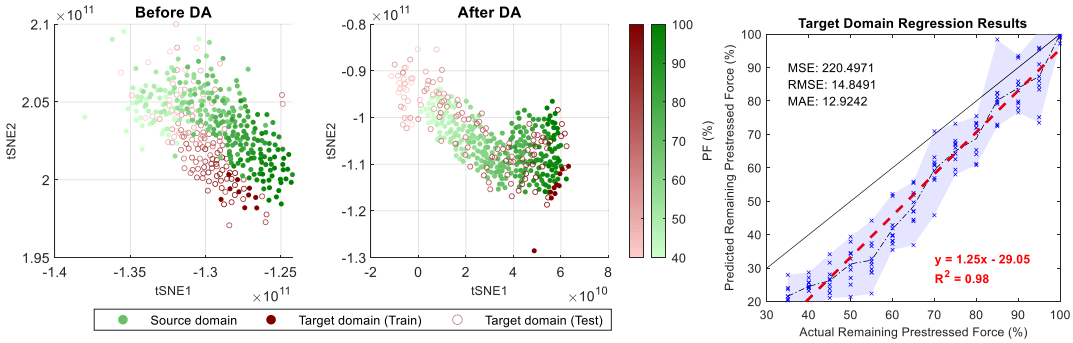


Figure 5.9 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 1 (bridge's length modelling error)

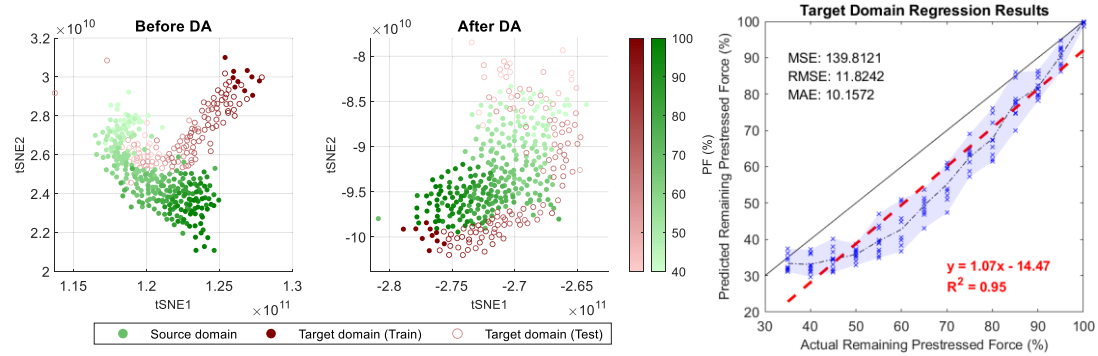


Figure 5.10 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 2 (bridge's rigidity modelling error)

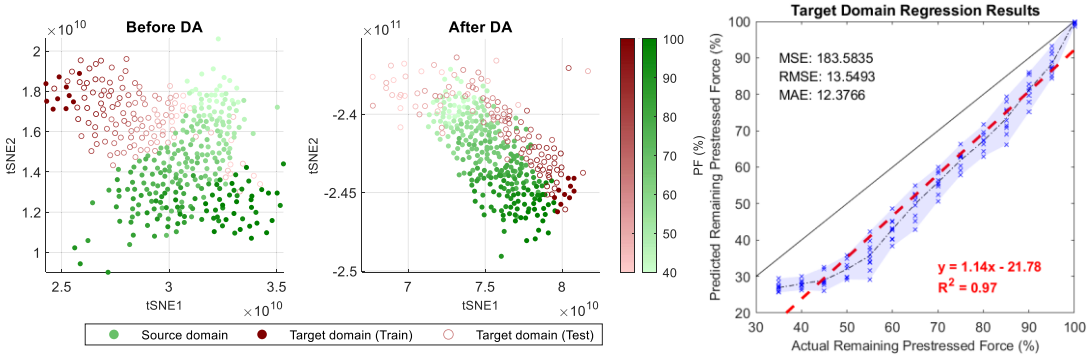


Figure 5.11 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 3 (bridge's mass modelling error)

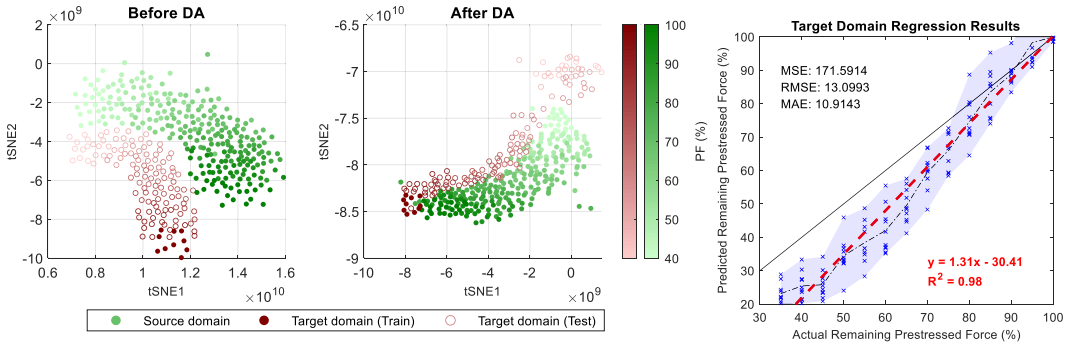


Figure 5.12 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 4 (bridge's damping modelling error)

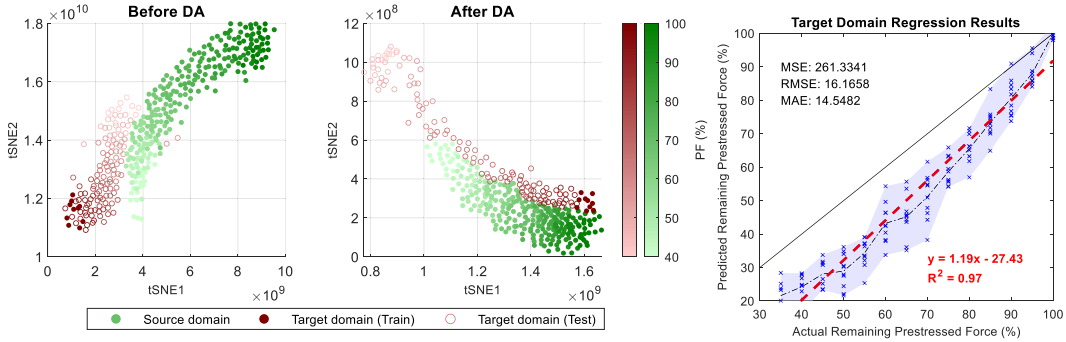


Figure 5.13 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 5 (bridge's boundary conditions modelling error)

5.5.3.2 Effect of Road Surface Roughness

Variations in road surface roughness can notably alter the acceleration responses of the VBI model, impacting both feature extraction and the accuracy of predictions. To evaluate the efficiency of the proposed hybrid DA in adapting to such domain shifts, a target model (Target #6) is set with Class B road roughness, representing increased roughness, whereas the source model featured Class A road roughness. Figure 5.14

depicts the effects of surface roughness on domain shift by demonstrating the distribution of representative features for the source and target domains. There is a clear shift in the feature space.

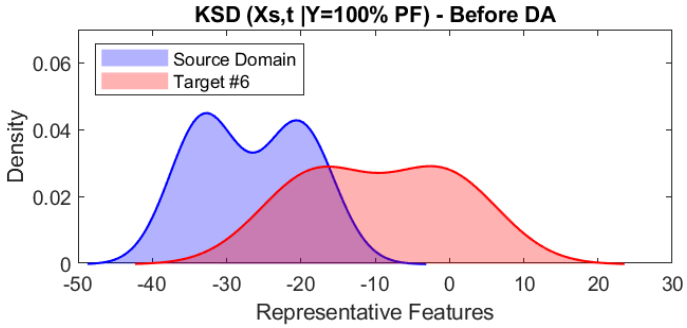


Figure 5.14 Feature distribution of target domain representative features for the target with road roughness modelling error at 100% PF condition

Figure 5.15 shows the tSNE visualization of source and target domain features by the proposed hybrid approach and the predicted prestressed forces. The results show that the domain feature shift caused by road surface roughness is reduced by the proposed hybrid model. There is an approximate linear relationship between the predicted and actual prestressed forces.



Figure 5.15 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 6 (road roughness modelling error)

5.5.3.3 Effect of Errors in Vehicle Parameters and Speed

Variations in vehicle parameters and speed significantly impact the dynamic interaction between the vehicle and the bridge, influencing PF-sensitive features and PF estimation results. Target 7 involves reducing vehicle speed from 15m/s to 12m/s, while Target 8 entails switching from Vehicle A to Vehicle B. Vehicle A's parameters are as listed in Section 3.1, with natural frequencies of 1.63, 2.30, 10.35, and 15.10 Hz. For Vehicle B,

both mass and spring stiffness are increased by 20%, resulting in natural frequencies of 1.78, 2.11, 10.35, and 15.10 Hz.

Figure 5.16 illustrates the effect of modelling errors in vehicle parameters and speed on domain shift by displaying the distribution of representative features for the source and each referenced target domain. It is evident that these errors induce more significant domain shifts than those typically observed in scenarios involving bridge modelling errors.

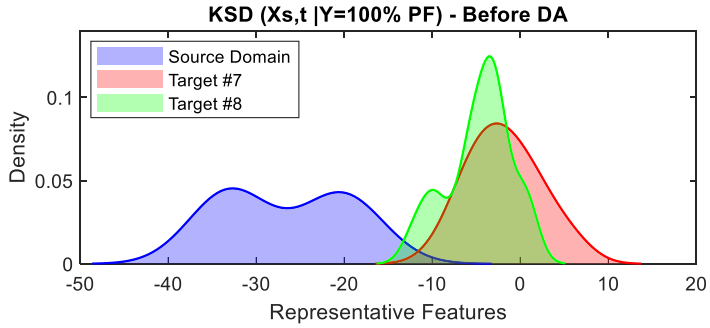


Figure 5.16 Feature distribution of target domain representative features for the target with vehicle modelling error at 100% PF condition

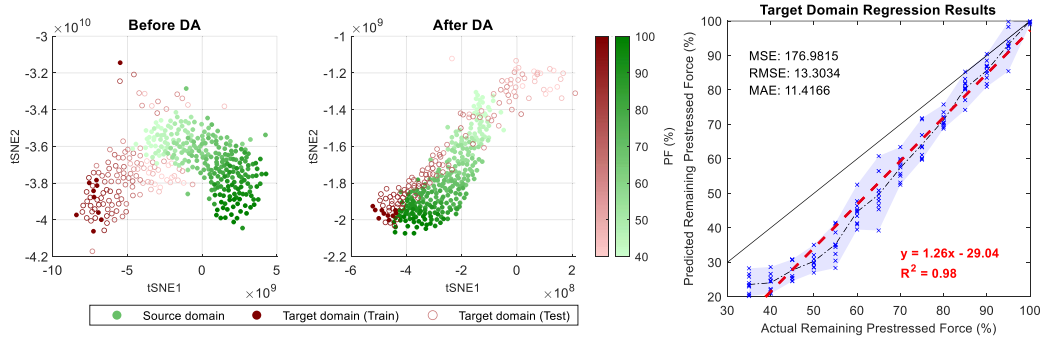


Figure 5.17 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 7 (vehicle speed error)

Figure 5.17 and Figure 5.18 depict the efficiency of the hybrid DA approach in mitigating domain shifts attributed to variations in road vehicle speed and modelling inaccuracies. Additionally, these figures provide PF estimation results for the target domain test data, demonstrating the approach's capability in scenarios where discrepancies exist between the speed or parameters of the modelled and actual passing vehicle, ensuring reliable prestress force predictions.

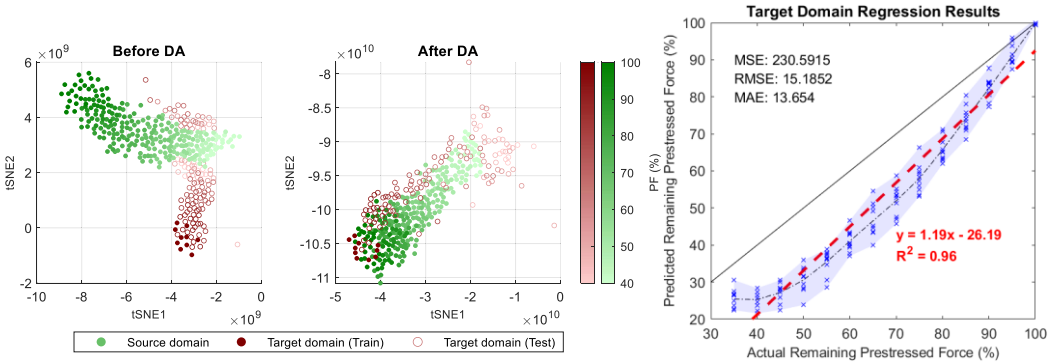


Figure 5.18 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 8 (vehicle modelling error)

Since the discrepancy between the finite element model of the bridge and the actual bridge is not precisely determined in reality, any of the aforementioned targets could occur. Evaluation results demonstrate that the proposed hybrid DA approach effectively minimises domain shifts in all scenarios, enabling PF estimation to be performed within an acceptable error range, which generally leans towards conservative estimates. Further analysis of this approach is presented in the following section, where the impact of a multi-variational target, the selection of hyperparameters, and the method's ability to handle unseen domains are examined.

5.5.4 Further Explorations on a Target with Multiple Deviations

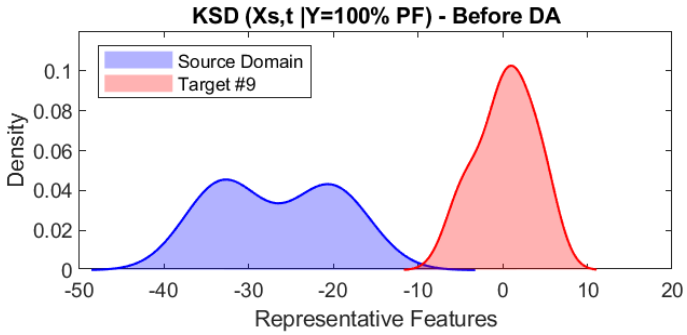


Figure 5.19 Feature distribution of target domain representative features for the target with multiple deviations at 100% PF condition

A target model was set up to incorporate all the specified deviations, presenting a scenario with modelling errors in the bridge, road, and vehicle simultaneously. This target was particularly important as it combined multiple sources of discrepancies, testing the model's ultimate capability in domain adaptation. Figure 5.19 illustrates the domain shift

between this target domain and the source domain by showing the distribution of their representative features at 100% PF condition.

Figure 5.20 highlights the capabilities of the hybrid DA technique in adjusting to significant variations between source and target domains. tSNE representation of the source and target domains illustrates changes in feature distributions before and after the application of hybrid DA. The PF estimation outcomes for test data in the target domain confirm the method's resilience and efficacy, even in the face of significant domain shifts, showing its practical applicability.

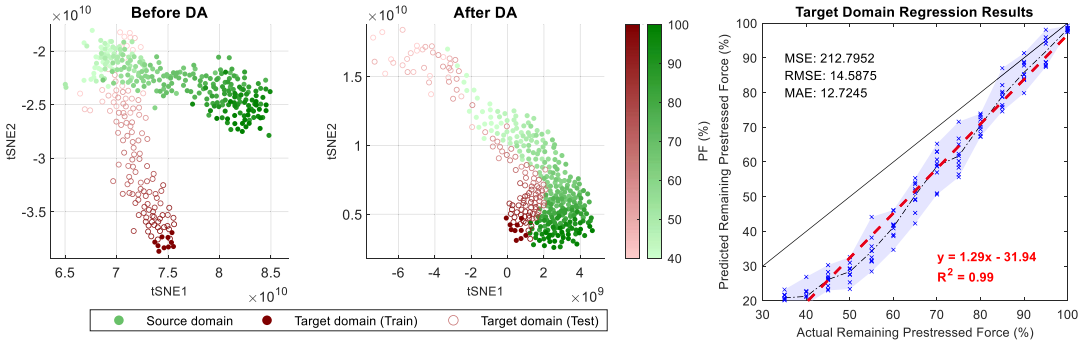


Figure 5.20 tSNE analysis of source and target domain features before and after hybrid DA, with PF estimation results for Target 9 (multiple deviations)

5.5.5 Network Hyperparameters Study

In this section, the effect of hyperparameter variations on the model's performance is investigated through a series of tests, with each hyperparameter being modified individually. Adjustments were made to the number of neurons in both the generator and discriminator layers, batch size, learning rates, number of epochs, MMD kernel size, the ratio of MMD loss to regression loss, and the adversarial loss factor. Table 5.2 presents a comparison, showing the impact of each hyperparameter adjustment on the performance of the model in transferring PF estimation knowledge from the source to Target 9.

These results show that a proper selection of the hyperparameters can effectively increase the performance of the Hybrid DA approach. Reducing the batch size and the learning rate of the regression model (Generator G) improves PF identification performance, suggesting that a finer update during training increases the model's ability to learn and generalise from the data. A substantial reduction in the number of neurons in Generator G's hidden layers (from 50 to 20) leads to improved performance, indicating that the model can achieve optimal performance with a relatively simpler architecture. This could

imply that the features extracted from the pre-trained CNN are robust and the regression model does not require excessive complexity to capture the underlying patterns, thereby promoting a more efficient and less overfitting-prone model structure.

Table 5.2 Model Hyperparameters Studies

		MAE	RMSE
Benchmark model		12.72	14.59
Comparison models			
Hyperparameter	Modifications		
No. of Neurons for G	50 to 20	9.82	11.88
No. of Neurons for D	5 to 3	13.99	15.72
Batch Size	16 to 32	14.76	17.07
Learning Rate G (η_G)	0.5e-3 to 1e-4	10.88	12.66
Learning RateD (η_D)	1e-6 to 0.5e-6	12.77	14.69
No. of Epochs	20000 to 10000	15.02	16.48
MMD kernel size (σ)	5 to 10	13.07	15.09
MMD/Reg. Loss ratio (α)	1000 to 100	11.09	12.9
Adv. Loss Factor (γ)	100 to 10	12.98	15.07

In contrast, reducing the number of neurons in the Discriminator D's hidden layers (from 5 to 3) tends to deteriorate the model's performance, as reflected by increased MAE and MSE. This suggests that the Discriminator D may require a certain level of complexity to effectively distinguish between the source and target domain features, and effectively play its role in the adversarial component of the training. Reducing the number of epochs to half (from 20000 to 10000) shows an increase of the MAE from 12.72 to 15.02, signifying that the model benefits from extended training periods due to the complexity of the domain adaptation task.

In terms of the MMD kernel size (σ), the study suggests that a smaller kernel size yields better performance in aligning the distributions of the source and target domain features. Moreover, adjusting the MMD to regression loss ratio (α) from 1000 to 100 results in a noticeable improvement in performance, indicating the importance of balancing the domain adaptation component (MMD loss) with the task-specific component (regression loss) in the final loss function of the generator.

While the adversarial loss factor (γ) does not show high sensitivity to changes, its optimal adjustment maintains a harmonious adversarial training process, ensuring that the feature extractor and the discriminator are well-aligned in their objectives.

Based on this parameter study, the model was trained on the target 9 datasets with 20 neurons in network G, 5 neurons in network D, a batch size of 16, a learning rate of 1e-4 for G, a learning rate of 1e-6 for D, 20,000 epochs, an MMD kernel size of 5, an MMD/Reg. Loss ratio of 1000, and an Adversarial Loss Factor of 100. The final regression results, presented in Figure 5.21, demonstrate a notable improvement in network performance. The MAE was reduced from 12.72 to 9.28, and the RMSE was reduced from 15.72 to 10.71.

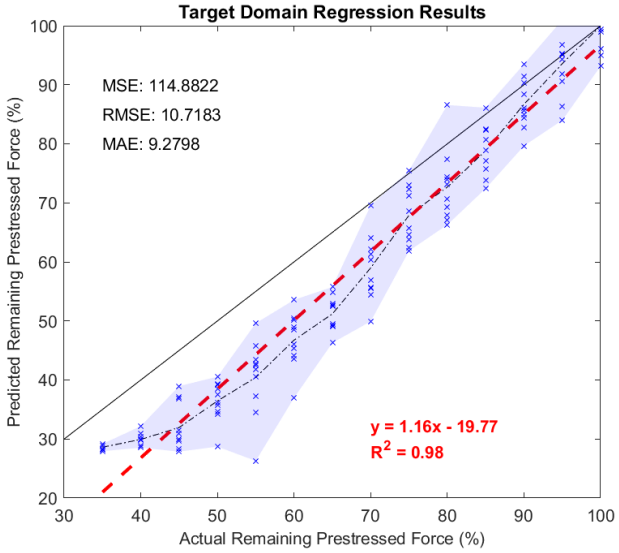


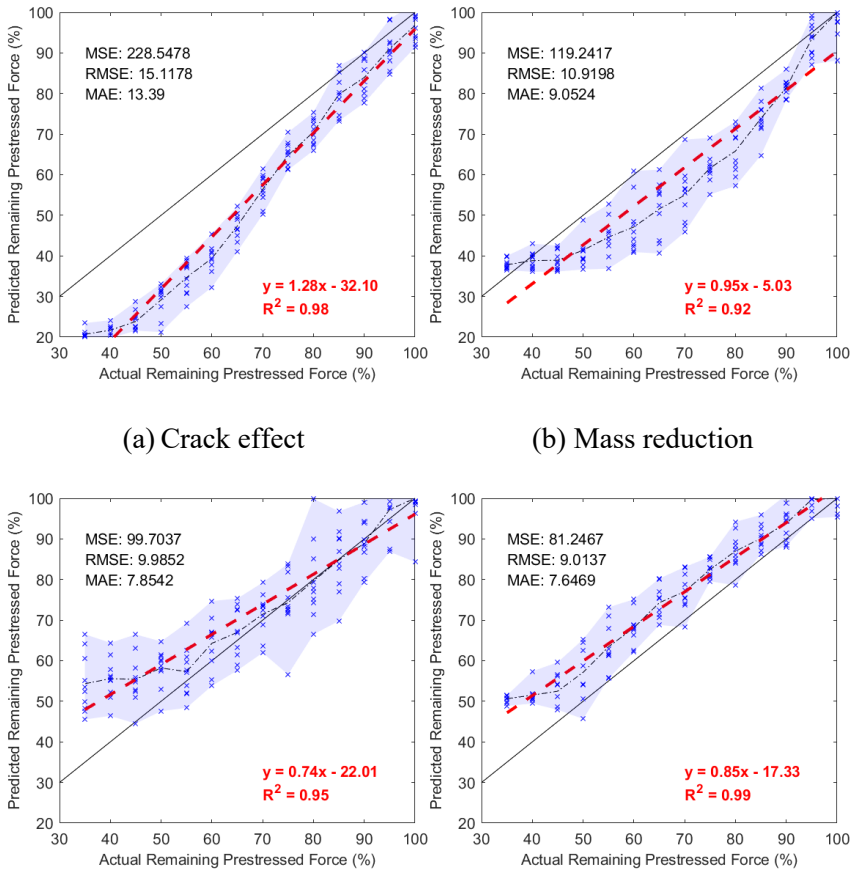
Figure 5.21 Regression Results for Target 9 with Carefully Selected Hyperparameters

5.5.6 Evaluating Model Robustness Across Unseen Domains

The proposed method is designed to leverage the data from a single label of the target domain to adapt the source and target domains for the PF estimation task. This is a practical method to utilise a limited set of available data from a desired domain. This approach relies on having the structural VBI response in the bridge's fully prestressed state and assumes that other bridge factors remain relatively stable, in the event of a loss in prestressed force. However, real-world scenarios can be more complex; structural properties of bridges change over time, potentially leading to variations in VBI responses. Moreover, damages to bridges may not solely pertain to PF loss. For instance, local stiffness reductions due to cracks can alter how a bridge reacts to the passing vehicle.

Additionally, the vehicle parameters and speed during the testing phase may vary from those employed in the domain adaptation process.

This section investigates the impact of such unseen domains on the effectiveness of the proposed method. The hybrid DA model is trained using data from "Target 9", wherein Vehicle B crosses the bridge at a speed of 12 m/s, with the bridge's total mass assumed to be 4800 kg. To assess the model's robustness, four unseen domains are introduced for evaluation: two related to changes in bridge parameters and two concerning vehicle parameters and speed.



(a) Crack effect (b) Mass reduction

(c) Vehicle parameter variations (d) Speed variations

Figure 5.22 Model performance evaluation across unseen domains

The first unseen domain aims to assess the model's performance in the presence of cracks, utilising the Wahab et al. (1999) model for crack introduction. A crack with a severity (γ) of 20%, a length (β) of 0.2 in the damaged zone, and a variation in Young's modulus (n) set to 2, affects one-quarter of the bridge's length. The regression outcomes for this domain are depicted in Figure 5.22 (a). The second unseen domain explores the model's

response to a reduction in the bridge's mass to 4600 kg, with corresponding regression results shown in Figure 5.22 (b). The third domain considers variations in the passing vehicle's parameters, slightly altered from Vehicle B, with natural frequencies of 1.70, 2.22, 10.35, and 15.10 Hz. The outcomes for this domain are illustrated in Figure 5.22 (c). Finally, in the fourth domain, the vehicle speed is adjusted to 8 m/s, with results presented in Figure 5.22 (d).

These evaluations reveal that the model's performance experiences a slight decline across unseen domains, which are likely encountered during a bridge's lifespan testing. Nonetheless, the outcomes remain sufficiently reliable, underscoring the proposed method's robustness in addressing unseen domains. This adaptability highlights the method's potential for practical application in monitoring and maintaining bridge safety.

5.6 Discussions

This section discusses the effectiveness and limitations of the proposed hybrid domain adaptation approach for prestressed force estimation in bridges:

- This approach demonstrates the feasibility of transferring finite element model knowledge to real-world applications, closing the simulation-to-reality gap.
- The hybrid domain adaptation approach successfully mitigates domain shifts caused by variations in bridge parameters, vehicle characteristics, and road surface conditions. Results show a significant reduction in MAE from 37.14 to 9.28 and RMSE from 46.40 to 10.71 before and after domain adaptation for Target 9. Reducing the batch size, adjusting the learning rates, and simplifying the generator architecture further enhanced performance, with a reduction in MAE from 12.72 to 9.28 and RMSE from 15.72 to 10.71 for the most challenging target domain (Target 9).
- The method effectively handled unseen domain shifts, such as a 20% reduction in vehicle speed, a 12% decrease in bridge rigidity, and variations in road surface roughness, with MAE remaining below 15 for all the unseen domains.
- Despite the demonstrated effectiveness of the hybrid domain adaptation approach, certain constraints remain. For instance, the method requires baseline responses from the structure in its fully prestressed state, which requires initiating monitoring at an early stage. The proposed hybrid

model could be updated continuously through continuous learning from continuous monitoring data. Additionally, while the model adapts well to various domain shifts, highly complex or rapidly shifting real-world conditions may still be challenging. Future work includes extending the approach to a broader range of bridge types and performing comprehensive experimental validations under diverse modelling uncertainties and environmental factors.

5.7 Summary

In this study, a novel Hybrid Domain Adaptation (Hybrid DA) approach has been proposed for the prediction of the remaining prestressed force in bridge structures to address the highly challenging issues of limited labelled data in real-world scenarios and discrepancies between simulations and actual structures. The method innovatively uses a single label in the target domain to mitigate the domain shift. The PF identification regression model is trained on AlexNet features extracted from the time-frequency responses of a finite element model of the Vehicle-Bridge Interaction (VBI) system. A second finite element model, chosen as the target domain, acknowledges common discrepancies between synthetic and real-world data. The Hybrid DA strategy, which integrates Maximum Mean Discrepancy (MMD) and adversarial learning, aligns features from both domains, creating a unified feature space for accurate PF prediction in the target domain. This approach makes a significant contribution to the field of structural health monitoring and bridge condition assessment. The following conclusions are obtained,

- The results show that the proposed hybrid DA model could be reliable and efficient in predicting the prestressed force from the VBI responses.
- The proposed approach could reduce the domain shift due to the uncertainty caused by the modelling errors of the VBI system with only the fully-prestressed label.
- In practice, there are datasets from limited damage scenarios. The proposed approach could be used to predict the prestressed forces for unseen datasets.
- While this research represents a significant leap in structural health monitoring, the complexities of real-world applications necessitate further experimental validations to ensure the robustness, adaptability, and overall reliability of the proposed method under diverse conditions.

CHAPTER 6 - DOMAIN-GENERALISED AUTOENCODER FOR BRIDGE DAMAGE IDENTIFICATION

6.1 Overview

This chapter addresses the most restrictive setting in transfer learning, where no target labels are available, and demonstrates the application of domain generalisation to the condition assessment of unseen bridges, utilising vehicle acceleration responses.

Drive-by techniques, which involve dynamic measurements recorded on a moving vehicle, have proven that there is a strong correlation between vehicle acceleration and the structural conditions of bridges. The acceleration time-history of vehicle-bridge interaction (VBI) offers a direct, information-rich basis for bridge condition assessment. Deep autoencoders have demonstrated their ability to learn compact and physically meaningful latent representations directly from vibration signals.

This chapter concentrates on developing an autoencoder-based method for reconstructing multi-channel bridge accelerations from single-vehicle accelerometer data. The extracted latent space features are then utilised to detect crack severity as scalar damage indicators. Additionally, a dual-output autoencoder is trained to reconstruct a time-history crack signature function, representing the crack as a stiffness reduction function through the bridge span. A hybrid loss function incorporating time, frequency, and phase components is used to minimise reconstruction errors, optimised through an Optuna search.

A significant challenge addressed is the domain shift between simulated finite element (FE) training data and real-world data using the domain generalisation technique. To achieve this, a comprehensive dataset is generated using the Latin Hypercube Sampling method, which encompasses a range of flexural rigidity (EI), span length, structural mass, vehicle speed, and road surface roughness, enabling robust training against modelling uncertainties.

6.2 Related Works

The VBI acceleration responses of both the vehicle and the bridge can provide valuable insights for damage detection. Specifically, drive-by techniques, which involve dynamic measurements recorded on a moving vehicle, have been shown to effectively correlate between vehicle acceleration and the structural conditions of bridges (McGetrick et al., 2015). Vehicle acceleration responses contain significant information related to bridge conditions. Demirlioglu and Erduran (2024) have employed wavelet analysis of vehicle acceleration signals to identify bridge damage. A recent benchmark study highlights the growing emphasis on data-driven approaches: four new drive-by methods (including deep autoencoders and clustering algorithms) were tested on a common simulated dataset with varying spans, damage levels, road roughness, and vehicle properties (Cantero et al., 2024). S. Li et al. (2024) indicate that deep autoencoders can effectively learn compact, physically meaningful latent representations from vibration sequences.

Moreover, autoencoders can extract rich information from vibration signals, making them highly beneficial for structural condition assessment in several ways:

a) Feature extraction: Deep autoencoders have proven effective in extracting compact, physically meaningful features from structural vibration data (Etem, 2025). These features can then be employed as inputs for various supervised machine learning techniques aimed at condition assessment (Bacsa et al., 2025).
b) Reconstruction error: In unsupervised SHM settings, autoencoders are often trained on healthy-condition data so that damage can be flagged by increases in the signal reconstruction error (Römgens et al., 2024). However, to differentiate between measurement noise or ambient errors and structural damage, many unsupervised learning-based methods require the signal reconstruction error threshold to be set manually for different damage types (Rastin et al., 2021).
c) Reconstruction of damage signature functions: A novel method introduced in this chapter represents the effect of a crack on a concrete beam as a time-history stiffness reduction function (crack signature function). By reconstructing this function from vehicle/bridge acceleration data, one can infer the crack's location, severity, and shape.

Several studies in the recent years have reported successful applications of autoencoders for structural damage detection. Li et al. (2023) developed a real-time drive-by damage detection method using a deep autoencoder trained on short-time vehicle vibration segments in the frequency domain to improve noise robustness. The autoencoder's latent features served as damage indicators for passing vehicles, achieving about 86% accuracy in detecting bridge damage of varying severities in simulations. X. Li et al. (2024) Introduced a physics-informed autoencoder that continuously learns an adapted baseline model for the structure from a short duration of undamaged vibrational data. Their method improved minor damage detection rates by roughly 35% compared to a standard autoencoder. Autoencoder variants such as variational autoencoders (VAEs) have also been employed for SHM; Koune and Cicirello (2024) proposed that adding physics-based latent space constraints in a VAE can disentangle damage-related factors from environmental influences.

This chapter concentrates on the development of an autoencoder-based method for reconstructing multi-channel bridge accelerations utilising data obtained from a single vehicle accelerometer. The extracted latent space features are then used for detecting of the crack severity as an scalar damage indicator. In the next step, the proposed framework expands to reconstruct the crack signature function, represented as a time-series function, which identifies the damage location, depth, width, and shape.

As deep learning continues to advance in the field of civil engineering, researchers are investigating innovative network architectures and hybrid models to improve structural condition assessment. Convolutional neural networks (CNNs) and recurrent networks (RNNs) are often combined to leverage both spatial and temporal patterns in structural response data (Huang et al., 2022). For instance, Yessoufou and Zhu (2023) designed a two-stage CNN-LSTM encoder-decoder for long-term bridge monitoring, where the CNN extracts local vibration features and the LSTM captures long-term dependencies; this framework was shown to perform robustly for damage identification under varying temperature conditions in a continuous monitoring setting.

A major challenge addressed in this chapter is the domain shift between simulated training data and real-world bridge data. In the previous chapters, fine-tuning and domain adaptation techniques have been employed to address the domain shift issue. However, in situations where no data from the target domain is available, these techniques cannot be applied (Liu et al., 2024).

Unlike domain adaptation techniques that require target domain samples, this study employs a domain generalisation approach without needing any data from the target domain. By sampling a range of structural and loading parameters in simulation (e.g. varying bridge stiffness, span, vehicle speed, surface roughness) and creating a comprehensive generalised training dataset, an autoencoder can be exposed to different scenarios and learn in a domain-invariant manner.

The comprehensive dataset is generated using Latin Hypercube Sampling, a space-filling experimental design technique proposed by McKay et al. (2000) that avoids exponential growth in sample sizes while maintaining coverage across each domain variation. This method functions as a feature augmentation strategy by introducing structural variability and different noise levels to the autoencoder during the training process. This strategy aims to ensure that moderate deviations between the FE model and the actual bridge do not compromise the performance of damage detection.

In this study, multiple encoder-decoder architectures, CNN, LSTM , and a hybrid model combining CNN and LSTM, were trained to reconstruct the bridge acceleration response from vehicle acceleration. Their performances were compared to identify the optimal architecture for reconstructing the bridge response. Latent space features from the selected architecture were subsequently utilised for damage severity regression. Subsequently, a dual-output autoencoder was proposed, pairing the bridge acceleration decoder with a crack signature decoder. This model was trained to reproduce a continuous damage pattern derived from stiffness-reduction factors, following the formulation provided by Wahab et al. (1999) for the crack shape. A hybrid loss function incorporating time, frequency, and phase components was defined to minimise reconstruction errors. Hyperparameters for the network and the hybrid loss function were tuned through an Optuna search combined with a Hyperband pruner. The optimised model aims to predict the crack signature function while maintaining high-quality bridge acceleration reconstruction.

6.3 Data Acquisition and Preprocessing

6.3.1 Data Acquisition from the VBI Finite-Element Model

The vehicle-bridge interaction (VBI) model presented in Chapter 3 consists of a simply supported Euler-Bernoulli beam representing the bridge, coupled with a vehicle through a time-dependent tyre-road contact constraint. At each discrete time step, the coupled equations of motion are integrated using an explicit Newmark-β integration scheme. The simulation provides the following outputs:

- **Vehicle acceleration $\mathbf{X}_v[n]$**: The vertical acceleration at the vehicle centre of gravity.
- **Bridge acceleration $\mathbf{X}_b[n] \in \mathbb{R}^9$**: Acceleration histories at nine equally spaced locations on the bridge deck.

Each signal is sampled at 200 Hz and standardised per record, resulting in zero mean and unit variance. Both acceleration data are augmented using 5% to 10% noise level.

6.3.2 Signal Preprocessing Utilising Empirical Mode Decomposition

Initial analysis of the vehicle acceleration response identified a dominant, near-sinusoidal low-frequency component (< 0.2 Hz) that sits well below the bridge's natural frequencies and therefore carries no information about bridge dynamics or health condition. Empirical-mode decomposition (EMD) consistently isolates this drift as the sixth intrinsic mode function (IMF-6). This mode can be a result of long-wavelength road-roughness variations, quasi-static suspension effects, etc. and does not encode the bridge's dynamic or damage-sensitive behaviour. Figure 6.1 shows a representative decomposition: IMFs 1-5 capture the informative vibration content, whereas IMF-6 corresponds to unwanted low-frequency drift. IMF-6 acts as a baseline drift; In order to establish an EMD-based baseline-correction step, IMF-6 is discarded, and the filtered signal is reconstructed by summing IMFs 1-5.

$$X_v^{\text{clean}}[n] = \sum_{\text{K}=1}^{5} \text{IMF}_k[n] \qquad \text{Eq.6.1}$$

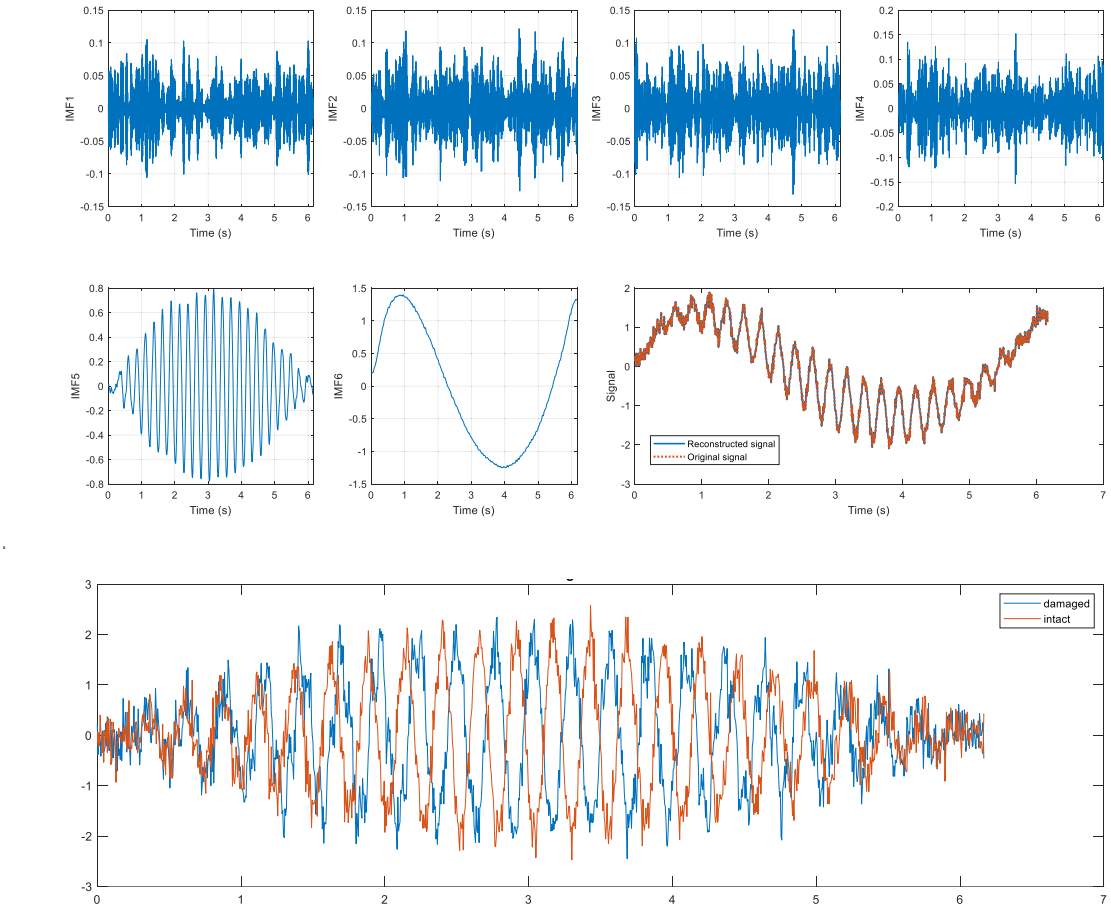


Figure 6.1 Baseline correction of the vehicle acceleration responses

6.4 Autoencoder Architecture Study for Reliable Bridge Acceleration Reconstruction

To design a reliable autoencoder for damage detection of bridges, firstly, it is important to create a robust structure for the model that could capture the damage-sensitive information from the data. In this study, the goal is to reconstruct the bridge acceleration responses from the vehicle acceleration response. This will make the trained autoencoder perform as a transfer function similar to the VBI model, which transforms the vehicle vibration into the bridge acceleration response. Three encoder designs (CNN, LSTM, and a parallel CNN + LSTM hybrid) are examined for their capacity to map pre-processed vehicle acceleration to the associated nine-channel bridge responses.

All other components (decoder, loss, optimiser, and training schedule) remain fixed to ensure that performance differences arise solely from the architecture of the proposed autoencoder. The structure of each autoencoder is presented in Table 6.1.

Table 6.1 Three different AE architectures

Encoder	Key layers	Latent Space dimension
CNN	1-D Conv(Unput Length→128) → BN → LeakyReLU → Conv(128→64) → BN → LeakyReLU → FC	100
LSTM	3× stacked LSTM (100 hidden units)	100
CNN + LSTM	CNN branch + LSTM branch run in parallel; bottleneck concatenated	200

All encoders feed a shared decoder with the mirrored structure of the encoder. The signal is split into non-overlapping windows of 50, and the data are spilled to the training and validation sets with an 80/20 ratio. The autoencoder is trained using the mean squared error (MSE) between the reconstructed and target bridge response windows as the loss function. The Adam optimiser is used, with specified learning rate and weight decay values. Each training iteration processes a batch of 32 windows. Early stopping is implemented, such that training stops if there is no improvement in validation loss for 10 consecutive epochs.

After training, full-length bridge signals are reconstructed by concatenating the output windows; the reconstructed signals are illustrated for each AE architecture in Figures 6.2 to 6.4.

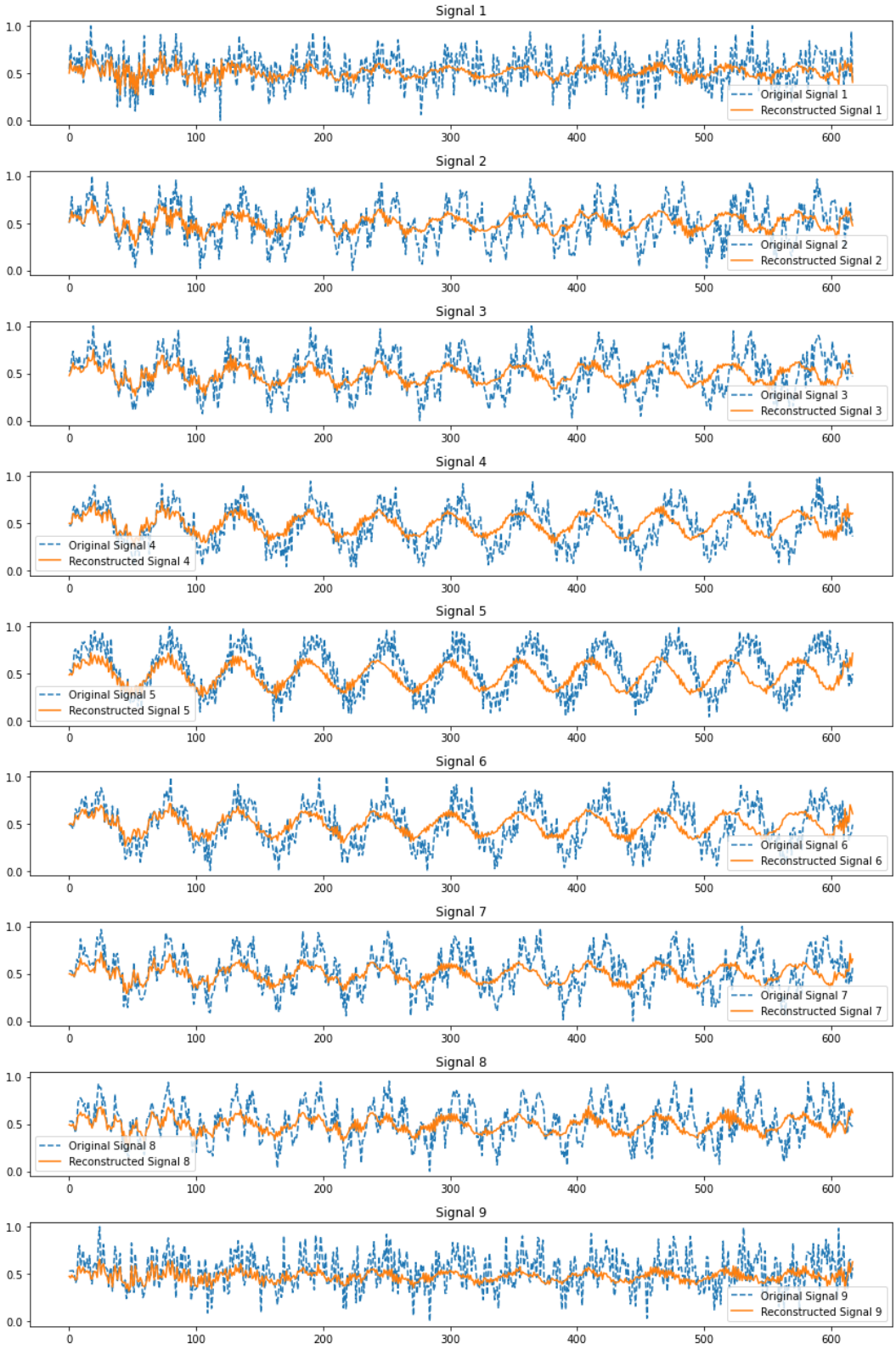


Figure 6.2 A sample of the reconstructed bridge acceleration signals using CNN-based AE

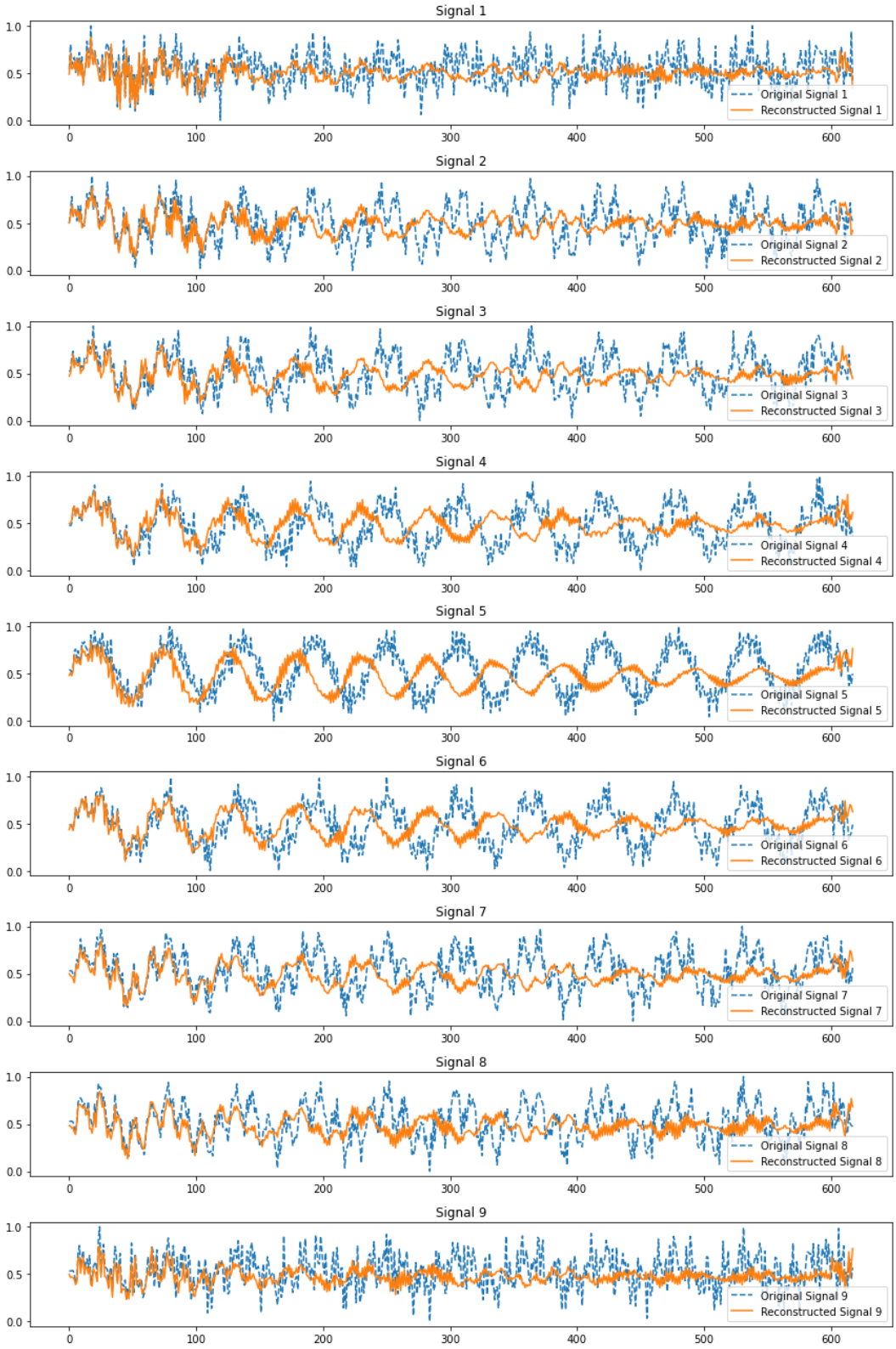


Figure 6.3 A sample of the reconstructed bridge acceleration signals using LSTM-based AE

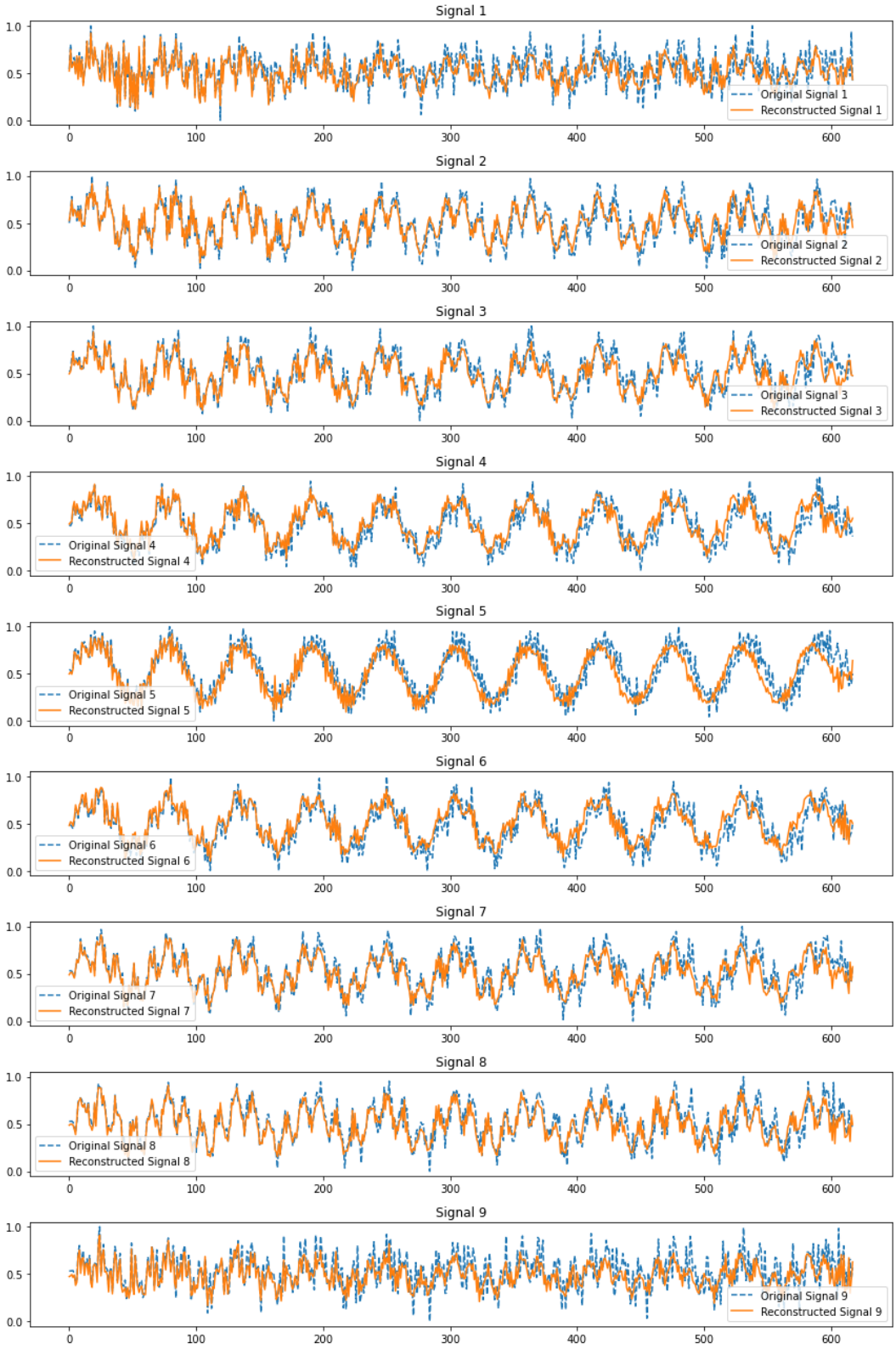


Figure 6.4 A sample of the reconstructed bridge acceleration signals using CNN+LSTM-based AE

It can be observed that the CNN branch effectively preserves the signal's high-frequency characteristics, while the LSTM branch accurately captures the low-frequency modal envelopes. The fusion of these two branches results in the reconstruction of the signal's low- and high-frequency content with a more precise phase alignment, leading to better performance compared to either branch when employed in isolation. For a better comparison, validation MSE of each model is tabulated in Table 6.2.

Table 6.2 Comparison of the preformance between the three different AE architectures

Encoder Architecture	Reconstruction MSE ($\times 10^{-3}$)
CNN	18.58
LSTM	25.12
CNN + LSTM	14.29

The performance of the hybrid CNN + LSTM architecture demonstrates superior reconstruction accuracy, achieving the lowest validation mean squared error (MSE). Consequently, it has been selected as the preferred autoencoder structure for this study. The illustrative architecture of the selected autoencoder is presented in Figure 6.5. Both the CNN and LSTM pathways are utilised in the feature extraction process via the encoder and the signal reconstruction process via the decoder.

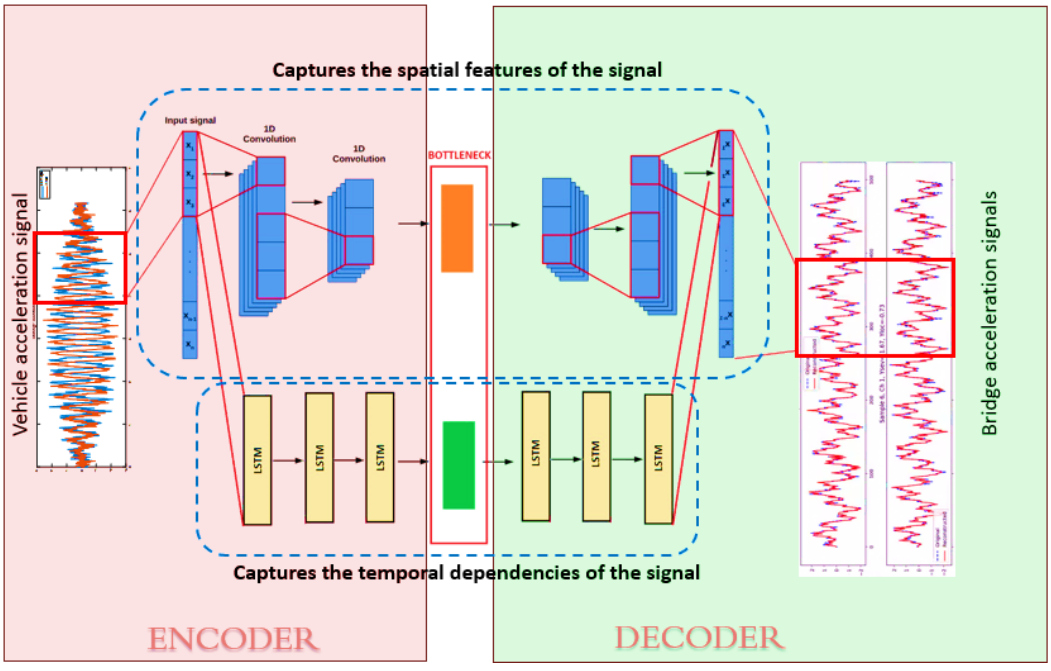


Figure 6.5 The selected AE architecture combining CNN and LSTM layers

An overview of the selected architecture is presented as follows:

Encoder

CNN Path

- Input shape: (batch, channels, Window Size).
- Multiple Conv1d layers (kernel_size=k, padding=1)
- Each layer's channel capacity in each layer is scaled by a factor N, which will be optimised
- A Flatten + Linear layer projects to a hidden size-dimensional vector.
- Output: a feature vector $z_{cnn} \in R^{hidden_size/2}$.

LSTM Path

- Input shape: (batch, Window Size, 1).
- A multi-layer LSTM extracts *temporal patterns* from the 1D signal.
- Final hidden state $z_{lstm} \in R^{hidden_size/2}$

Combined Latent

- Concatenate along the feature dimension.
- Final latent dimension = CNN hidden size + LSTM hidden size.

Decoder

LSTM Path

- The decoder reintroduces a multi-layer LSTM with the same size as the encoder.
- Output is reshaped/transposed to (batch, channels, Window Size)

ConvTranspose Path

- A series of ConvTranspose1d layers with kernel_size=k, padding=1
- Each layer reduces or modifies channels in a mirrored manner to the encoder.

Final Convolution

- Outputs the final multi-channel bridge signal (Xb)
- Shape: (batch, channels, Window Size).

6.4.1 Feature Extraction and Crack Severity Detection

Following the establishment of the CNN + LSTM auto-encoder as the preferred architecture, its latent representation of the data was extracted as damage-sensitive features intended for supervised damage detection. Every cleaned, normalised vehicle acceleration response is forwarded through the trained encoder, and the resulting 200-dimensional latent code is stored. The t-SNE representation of the extracted features, colored by the crack severity label, is presented in

Figure 6.6. A feed-forward network, consisting of three fully connected layers with 10 neurons and Relu activation, is trained on the extracted features to predict the damage severity. The dataset is split 80/20 into training and test sets. The optimiser used is Adam, with an initial learning rate of 5×10−4.

The performance of the crack depth (Ysev) regression task on the test data is presented in the Figure 6.7, showing an MSE of 2.65e-3. The low MSE on the test data demonstrates that the autoencoder's latent space provides damage-sensitive features despite the fact that the network was trained in a completely unsupervised manner, with the main focus on the reconstruction task. This error can be further optimised by introducing a more powerful loss function and fine-tuning the model hyperparameters.

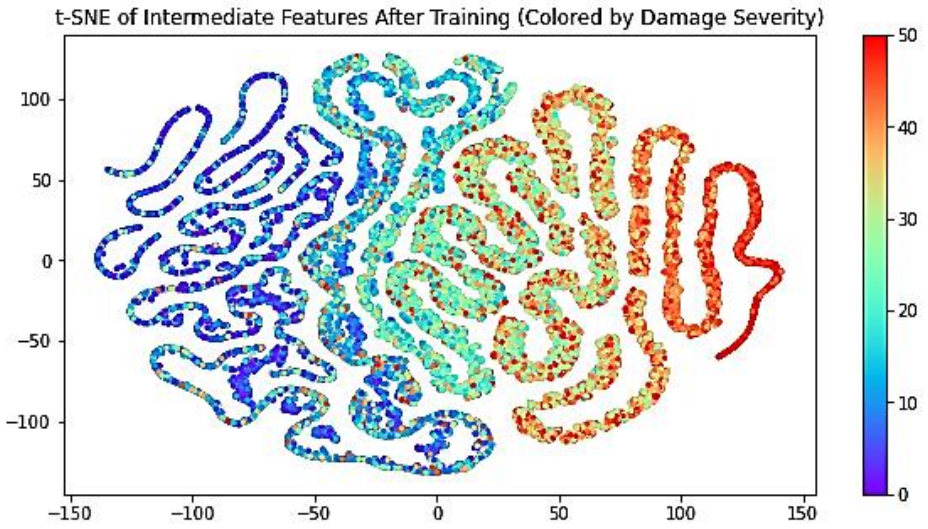


Figure 6.6 Extracted Features from the AE latent space

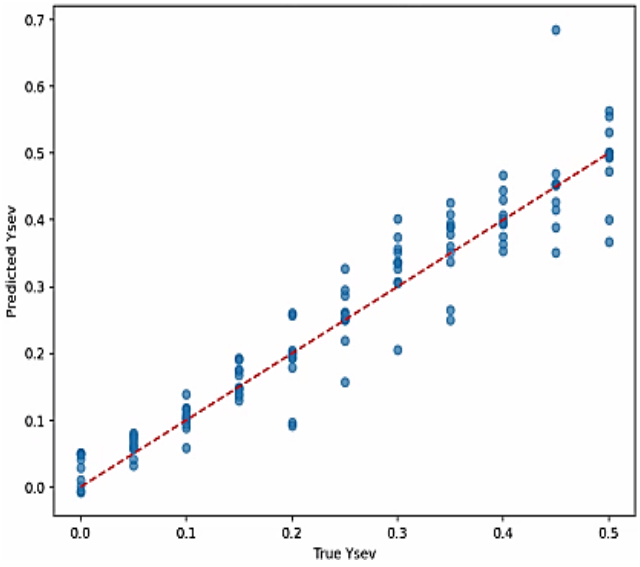


Figure 6.7 Damage severity regression results

6.5 Domain Generalisation

As previously discussed, a domain shift between simulation and real-world data is unavoidable. Various methodologies have been assessed to bridge the gap between the source and target domains. Fine-tuning and domain adaptation are two transfer learning techniques that can be used for this purpose. However, a representative sample or a number of representative samples of the target domain is essential for each of these approaches. The domain generalisation approach, as an alternative transfer learning method, exhibits superiority over the two aforementioned approaches in this particular aspect. In the domain generalisation framework, it is assumed that multiple labelled source domains are accessible or that the training dataset encompasses various domain variations. The model is subsequently trained concurrently across all domains, facilitating the learning of domain-invariant features of significance. The primary distinction between domain generalisation and domain adaptation lies in the requirement for domain adaptation networks to reference the labelled or unlabeled data from different classes within the target domain. Conversely, domain generalisation techniques seek to establish a domain-invariant feature representation space by aligning the labelled features across multiple domains. In other words, assuming N labelled source domains $\mathrm{D_s} = \left\{\mathrm{D_s^n} = \{(\mathrm{x_1^n}, \mathrm{y_1^n}), \ldots, (\mathrm{x_{ns}^n}, \mathrm{y_{ns}^n})\}\right\}_{n=1}^{N}$ with a joint distribution for all the samples $P_s^n(y_s^n|x_s^n) \neq P_s^{n'}(y_s^{n'}|x_s^{n'})$ while $n \neq n'$ and $n, n' \in \{1, \ldots, N\}$ the goal of domain generalisation is to learn $f(X) = Y$ using only the data of source domains in a way that

the differences between $f(X_T)$ and Y_T is reduced. X_T and Y_T are unseen in the training phase. A comparison between the domain generalisation method and the other proposed approaches is presented in Table 6.3.

Table 6.3 Comparison between domain generalisation and other methods (Zhou et al., 2022)

	Number of Source domains/task		$P_s(y_s\|x_s)$ vs $P_t(y_t\|x_t)$		Y_s **vs** Y_t		Access to $P_t(x_t)$
	=1	>1	=	≠	=	≠	
Supervised learning	✓		✓		✓		
Fine-Tuning	✓	✓		✓		✓	✓
Domain Adaptation	✓	✓		✓	✓	✓	✓
Domain Generalization	✓	✓		✓	✓	✓	

6.5.1 Domain Generalised Dataset Generation Utilising Latin Hypercube Sampling

To ensure that the proposed approach can generalise beyond the nominal bridge and vehicle configuration, a population of VBI scenarios was generated utilising Latin Hypercube Sampling (LHS) to cover a range of domain shifts that can be caused by modelling error.

Latin-hypercube sampling (LHS) is a stratified-random Monte-Carlo technique for covering the entire parameter space by generating a set of parameter values from a multidimensional distribution**.**

LHS creates a space-filling design with only N points in a D-dimensional parameter space, where a full factorial grid would result is N^D samples. The step-by-step procedure of LHS is broken down here:

1. Probability-space mapping: For every input variable $X_i(i = 1, \dots, D)$ define its cumulative distribution function (CDF) F_{X_i}. Transform the variable to the unit interval by:

   $$U_i = F_{X_i}(X_i), U_i \sim \mathcal{U}(0,1) \quad \text{Eq.6.2}$$

2. Stratification: Partition the interval [0,1] into N equal-probability strata

$$I_j = ((j-1)/N, j/N], j = 1, \dots, N \quad \text{Eq.6.3}$$

3. One draw per section: For each variable U_i draw one value $u_{i,j}$ uniformly from every section I_j. After this step each marginal distribution is exactly represented by the sample set $\{u_{i,j}\}_{j=1}^{N}$.
4. Independent permutation: Independently permute the N draws of every variable:

$$\{u_{1,\alpha_1(j)}, u_{2,\alpha_2(j)}, \dots, u_{D,\alpha_D(j)}\}, j = 1, \dots, N \quad \text{Eq.6.4}$$

where α_i is a random permutation of $\{1, \dots, N\}$. This step incorporates N multidimensional points in such a way that no two points are positioned within the same section.

5. Inverse transformation: Map each component back to the physical space with the inverse CDF:

$$x_{i,j} = F_{X_i}^{-1}(u_{i,\alpha_i(j)}) \quad \text{Eq.6.5}$$

Because every stratum of every variable is used exactly once, the sample uniformly "fills" all one-dimensional projections. Consequently, LHS attains much lower integration error than simple random sampling and achieves comprehensive coverage of the parameter space with orders-of-magnitude fewer model evaluations than a full factorial design *(McKay, Beckman, & Conover, 1979)*.

In order to construct the domain generalised dataset, LHS is applied to five uncertain physical parameters (bridge span length, flexural rigidity, mass, vehicle speed v and road roughness class). It is assumed that in drive-by methods, the dynamic parameters of the vehicle are known; therefore, they were not included as domain shift sources. While looping separately over three crack parameters, each LHS cycle selects parameter ranges in Table 6.4, resulting in 5500 unique training samples and 1100 independent test samples, which are obtained with a different random seed.

Table 6.4 Parameter ranges to create the domain-generalised dataset

Parameter	Train Range / Values	Variation from mean (%)
Bridge Length (L)	[29.85,30.15] (m)	± 0.5%
Bridge Rigidity (EI)	[1.9x10^7,2.1x10^7] (kN.m^2)	±5%
Bridge Mass (Pa)	[4750,5250] (kg/m)	±5%
Vehicle Speed (v)	[38,42] (km/hr)	±5%
Road Roughness Class	A, B	-

Each sample calls the VBI program function, obtains the vehicle and bridge signals. The signals have a variable length due to the varying vehicle speed. Therefore, all the signals are truncated to a length of 500 to be suitable for training a single autoencoder. If the signal length is less than 500, zero padding is performed to the end of the signal. and adds a 5% to 10% noise level as follows:

$$X_{\text{noisy}} = X + \eta \cdot \sigma_X \cdot \mathcal{N}(0,1) \qquad \text{Eq.6.6}$$

Where η denotes the noise level, σX is the standard deviation of the signal, and N(0,1) is the random noise extracted from a standard normal distribution. As a result, the sample space will cover the whole range of domain shifts as depicted in Figure 6.8.

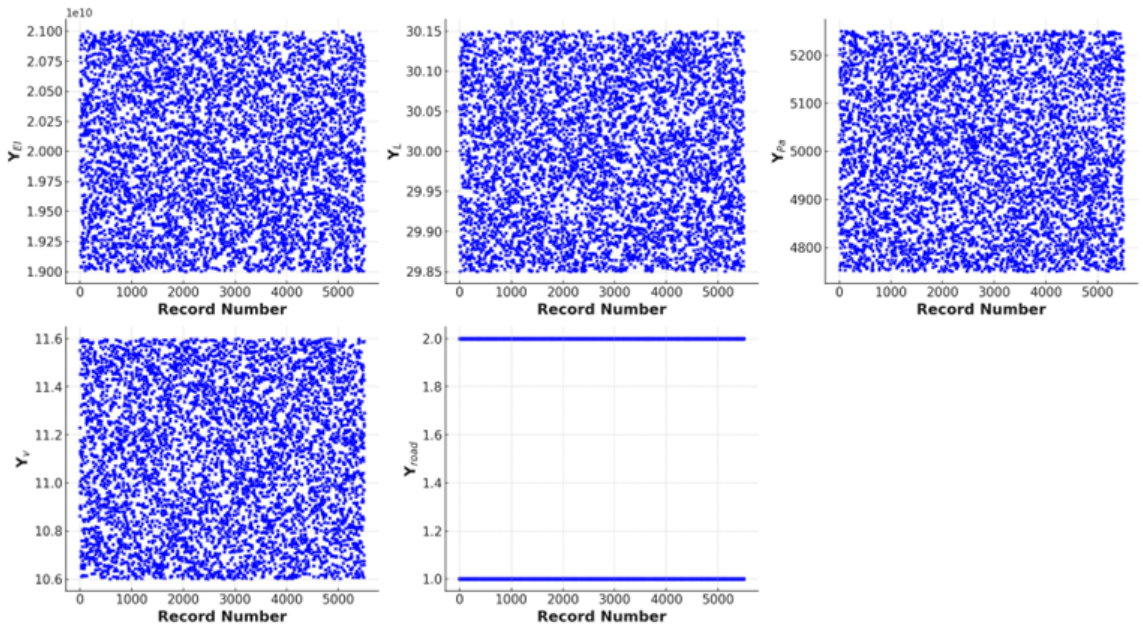


Figure 6.8 LHS generated samples across different labels

6.6 Autoencoder Loss and Hyperparameter Optimisation

Following the design of the autoencoder architecture and the generation of a comprehensive domain-generalised training dataset, it is important to optimise the model even further. Defining a robust loss function can significantly improve the signal reconstruction accuracy, and more effective reconstruction will result in a more informative feature space and, consequently, a higher accuracy in damage detection. This need for robustness is even more highlighted when the training data contains inherent domain shifts across various parameters. Furthermore, the optimisation of the model's hyperparameters is conducted utilising Optuna to reach the optimum performance of the model.

6.6.1 Hybrid Loss Function Definition

Previously, the only loss function minimised during the training process was the MSE of the reconstructed signal and the original signal in the time domain. However, for a demanding task like reconstructing a time series of data, this loss function has proven insufficient to capture all the details in the signal's time history. Therefore, a hybrid loss is designed to break down the signal reconstruction objective into more detail:

1. Time-domain fidelity between predicted and actual bridge signals,
2. Frequency-domain fidelity to capture spectral alignment,
3. Phase alignment to ensure the correct phase information,

6.6.1.1 Time Domain Loss Component

This is the classic mean squared error comparing the predicted signal to the true signal at each time step in the time domain.

Assuming $\hat{x}_b \in \mathbb{R}^{(B\times C\times W)}$ is the predicted amplitude for batch b, channel c, time index w, and $x_b \in \mathbb{R}^{(B\times C\times W)}$ is the true amplitude, then the time domain loss is calculated as:

$$\mathcal{L}_{\text{time}} = MSE(\hat{x}_b, x_b) = \frac{1}{BCW}\sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{w=1}^{W}\left(\hat{x}_{b\,b,c,w} - x_{b\,b,c,w}\right)^2 \quad \text{Eq.6.7}$$

6.6.1.2 Frequency Domain Loss Component

To extract the frequency domain fidelity, the real Fast Fourier Transform (FFT) is calculated for both the predicted and ground-truth signals. The real magnitude of the FFT is magnified with a light exponential up-weighting of higher bins. signals

$$P_{b,c}(k) = \left|\text{FFT}\{x_{b,c}^{\text{pred}}\}[k]\right|, T_{b,c}(k) = \left|\text{FFT}\{x_{b,c}^{\text{true}}\}[k]\right| \quad \text{Eq.6.8}$$

where $k = 0, \ldots, K-1$ shows the FFT bins. The real magnitude of the FFT is magnified with a light exponential up-weighting of higher frequency bins as:

$$w(k) = e^{\alpha_{\exp} k},\ \alpha_{\exp} = 0.01 \quad \text{Eq.6.9}$$

Then the MSE weighted frequency contents is calculated and averaged through the batch and channels to form the frequency domain loss as:

$$\ell_{b,c}^{\mathrm{FFT}} = \frac{1}{K}\sum_{k=0}^{K-1} \left[w(k)\left(P_{b,c}(k) - T_{b,c}(k)\right)\right]^2 \quad \text{Eq.6.10}$$

$$\mathcal{L}_{\mathrm{Freq}} = \frac{1}{BC}\sum_{b=1}^{B} \sum_{c=1}^{C} \ell_{b,c}^{\mathrm{FFT}} \quad \text{Eq.6.11}$$

6.6.1.3 Phase Loss Component

The Hilbert transform is performed on predicted and true signals to derive the instantaneous phase. Then, the MSE of the signal phases is calculated to ensure proper phase alignment. the phase loss is fined as:

$$\mathcal{L}_{\mathrm{phase}} = \frac{1}{BC}\sum_{b=1}^{B} \sum_{c=1}^{C} \mathrm{MSE}\left(\mathrm{Phase}\left(\hat{x}_{b\,b,c,.}\right), \mathrm{Phase}\left(x_{b\,b,c,.}\right)\right) \quad \text{Eq.6.12}$$

Combined Hybrid Loss

In conclusion, the final loss is represented as a weighted summation of the components above, as follows:

$$\mathcal{L}_{\mathrm{hybrid}} = \alpha_{\mathrm{time}} \cdot \mathcal{L}_{\mathrm{time}} + \alpha_{\mathrm{freq}} \cdot \mathcal{L}_{\mathrm{freq}} + \alpha_{\mathrm{phase}} \cdot \mathcal{L}_{\mathrm{phase}} \quad \text{Eq.6.13}$$

During backpropagation, each component's gradient influences the model's parameters, forcing the autoencoder to balance signal reconstruction in time, frequency, and phase.

The weighting factors (α) facilitate the prioritisation or deprioritisation of each loss and are selected based on an optimization approach.

6.6.2 Hyperparameter Optimisation Utilising Optuna

In order to minimise computational expenses, two of the bridge responses from a total of nine sensor locations have been selected. The selection of these two sensors, alongside other network hyperparameters, is executed utilising the Optuna search framework. Optuna is an automated hyperparameter optimisation framework provided by Akiba et al. (2019), specifically designed for machine learning applications. It employs Bayesian sampling of hyperparameters and trains the model based on a set of hyperparameters for a few epochs to evaluate the model performance and find the optimum value for each hyperparameter. It also incorporates an early-stopping rule called pruning, which allows

the model to terminate if a combination of hyperparameters results in suboptimal network performance. Therefore, Optuna can evaluate numerous configurations while eliminating ineffective ones after only a few epochs.

Given that there are weighting factors present in the loss function that require optimisation, the loss function cannot serve as the objective function for Optuna optimisation. Thus, the optimal configuration of the hyperparameters is determined based on the maximum Structural Similarity Index (SSIM) of the reconstructed signals.

$$\text{Objective} = \text{maximise}\left(\text{SSIM}\left(\hat{X}_b, X_b\right)\right) \quad \text{Eq.6.14}$$

SSIM is a comparison metric originally introduced for comparing images, but it can be applied just as well to 1D signals such as accelerations. SSIM measures likeness by comparing three quantities of the two signals: their mean (luminance), their variances (contrast) and their covariance (structure). For 1D signals *x* and *y* of the same length, SSIM is computed as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy})}{(\mu_x^2+\mu_y^2)(\sigma_x^2+\sigma_y^2)} \quad \text{Eq.6.15}$$

Where μ_x and μ_y are the mean and σ_x^2 and σ_y^2 are the variances of the signal x and y respectively, and σ_{xy} is the covariance of the two signals. When the means and variances are equal, and the signals share an identical structure, SSIM equals one. If the signals have no structural resemblance, SSIM will be zero.

The search range and the optimised values for the hyperparameters of the model are presented in Table 6.5.

Table 6.5 Optuna search range and selected hyperparameters

Parameter	Search Range /Values	Selected
Bridge Sensor Numbers	Two out of 9 sensors	2 , 6
Window size	{25,50,75,100,125,500}	500
Learning rate	[1e-5, 1e-3]	4e-4
Weight decay	[1e-7, 1e-3]	1.3e-5
Dropout cnn	[0.0, 0.6]	0.308
Dropout Istm	[0.0, 0.6]	0.357
Hidden size	{100,200,300,400,1000}	1000
Number of LSTM layers	[1, 3] (integer)	2
Number of CNN layers	[1,3] (integer)	3
Convolutional kernel size	[16, 32, 64] (integer)	32
Alpha time	[0.0, 2.0]	0.847
Alpha freq	[0.0, 2.0]	2.389
Alpha phase	[0.0, 2.0]	0.122

6.7 Evaluation of the Autoencoder Performance

The optimised model has been trained on the domain-generalised dataset, and the loss curves are illustrated in Figure 6.9. The reconstructed signals from the test dataset exhibit an SSIM of 0.9022. Three samples of the reconstructed signals are presented in Figure 6.10 Channel 0 in the reproduced signals corresponds to sensor number 2, while Channel 1 corresponds to sensor number 6. The first 3 PCA components, along with the tSNE representation of the extracted features from the latent space, are also depicted in Figure 6.11.

When the feature representations are coloured by road-roughness class, two distinct clusters appear, which confirms that the road-roughness is the strongest label retained by the encoder. Crack severity changes gradually across both representations, creating a continuous band that is well-suited for a regression task. In contrast, global bridge properties such as flexural rigidity (EI), span length (L), and mass (Pa) do not exhibit any visually detectable pattern. This indicates that the network has successfully suppressed these variations in the dataset, resulting in the extracted features being domain-invariant

across these domain shifts. In other words, the latent space features are damage sensitive, while filtering out most of the domain shifts caused by the modelling error.

The only variable that forms distinguished clusters is road roughness. However, this will not adversely affect the generalisation as long as the label can be predicted with high certainty. In practice, the road surface can be identified and removed from the generalised parameters. A dataset can be generated and used that only contains that specific road roughness class.

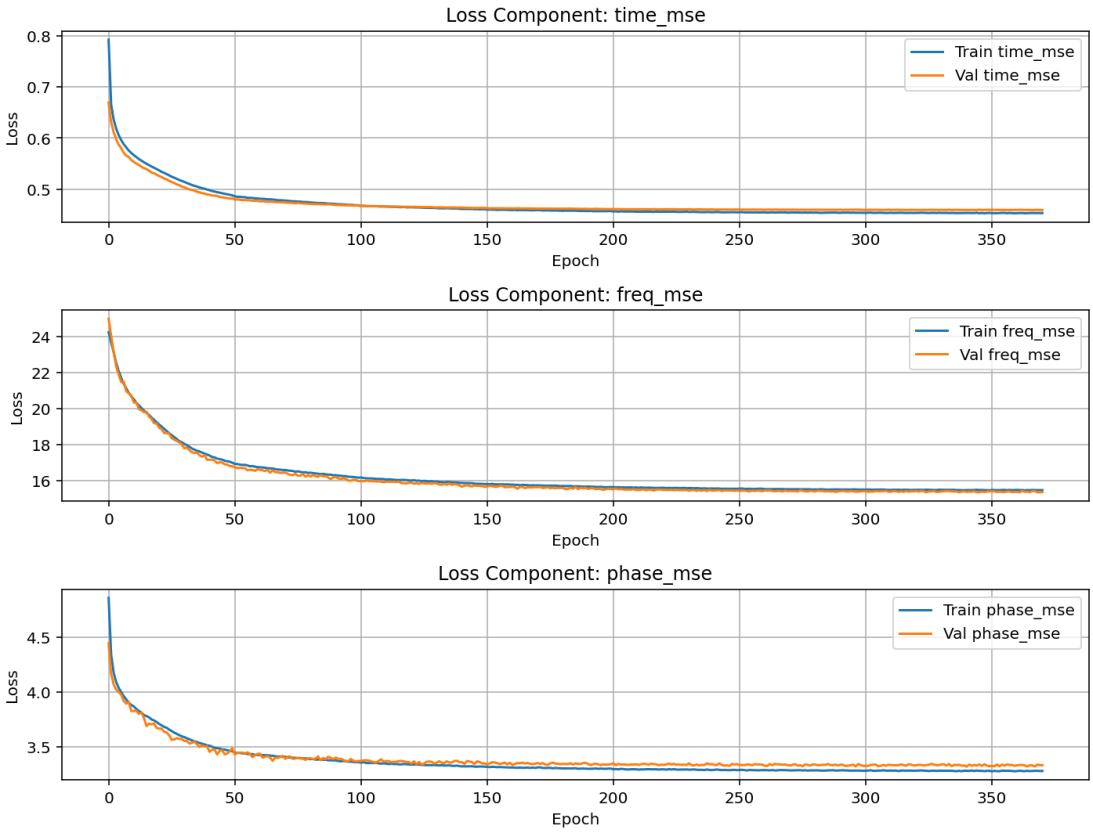


Figure 6.9 Train and validation loss curves for different loss components

It is important to note that this feature extraction process is entirely unsupervised, and no labels associated with the data are introduced to the network during the training phase. The first three PCs of the Ysev label illustrate that the extracted features exhibit a reasonable correlation with crack severity (Ysev). Consequently, a Support Vector Machine (SVM) regression is conducted on the extracted features to investigate their effectiveness in determining crack severity, utilising a 20/80 split of the data for training and testing purposes. The results of the regression analysis are presented in Figure 6.12, indicating an MSE of 4.3e-3. These regression outcomes demonstrate that the crack severity can be detected with a precision of approximately ±20%. While this level of

precision may not be considered ideal, it is acceptable when taking into account that the training dataset encompassed a variety of domain shifts.

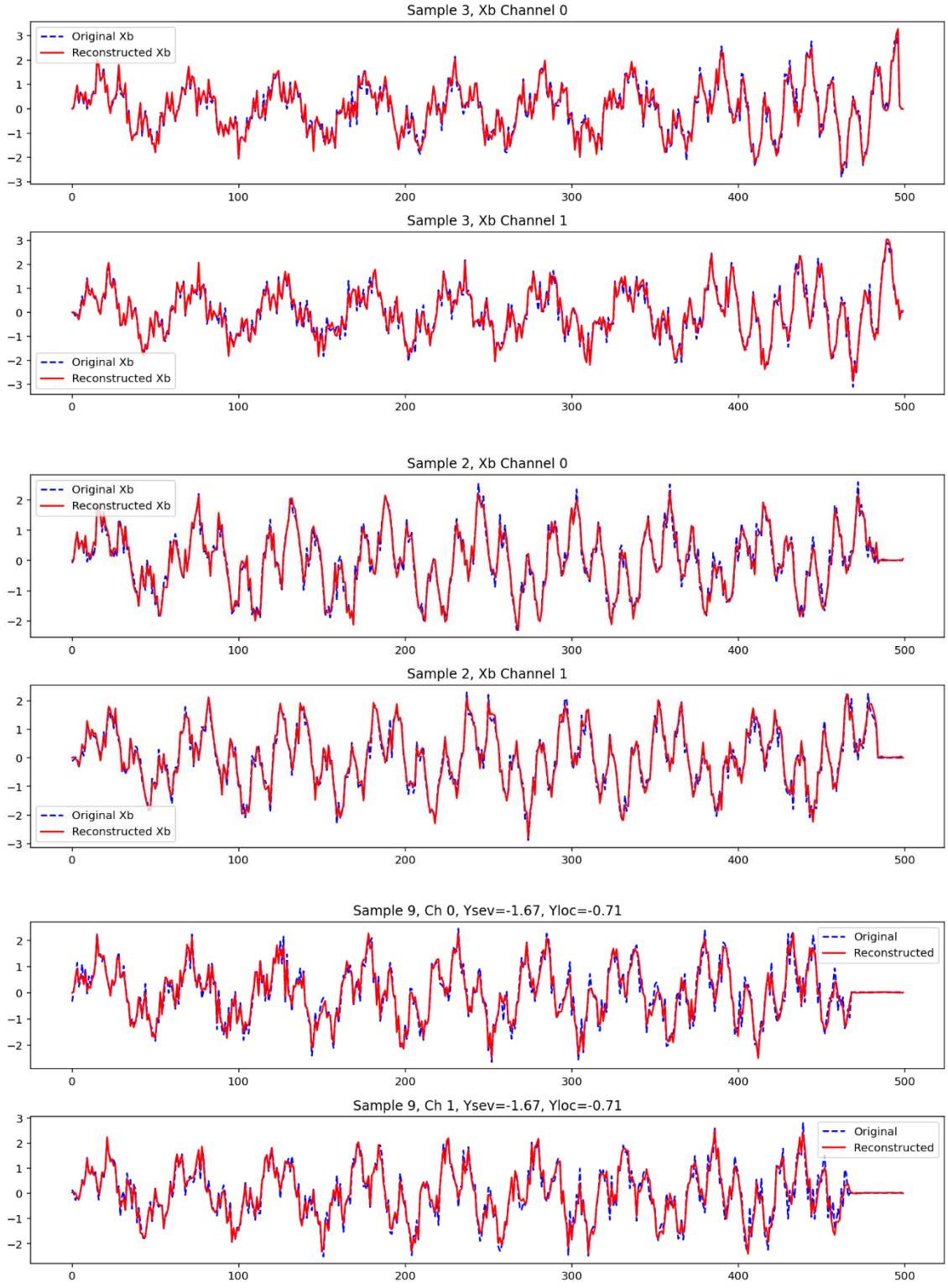


Figure 6.10 Examples of the reconstructed signals by the optimised AE

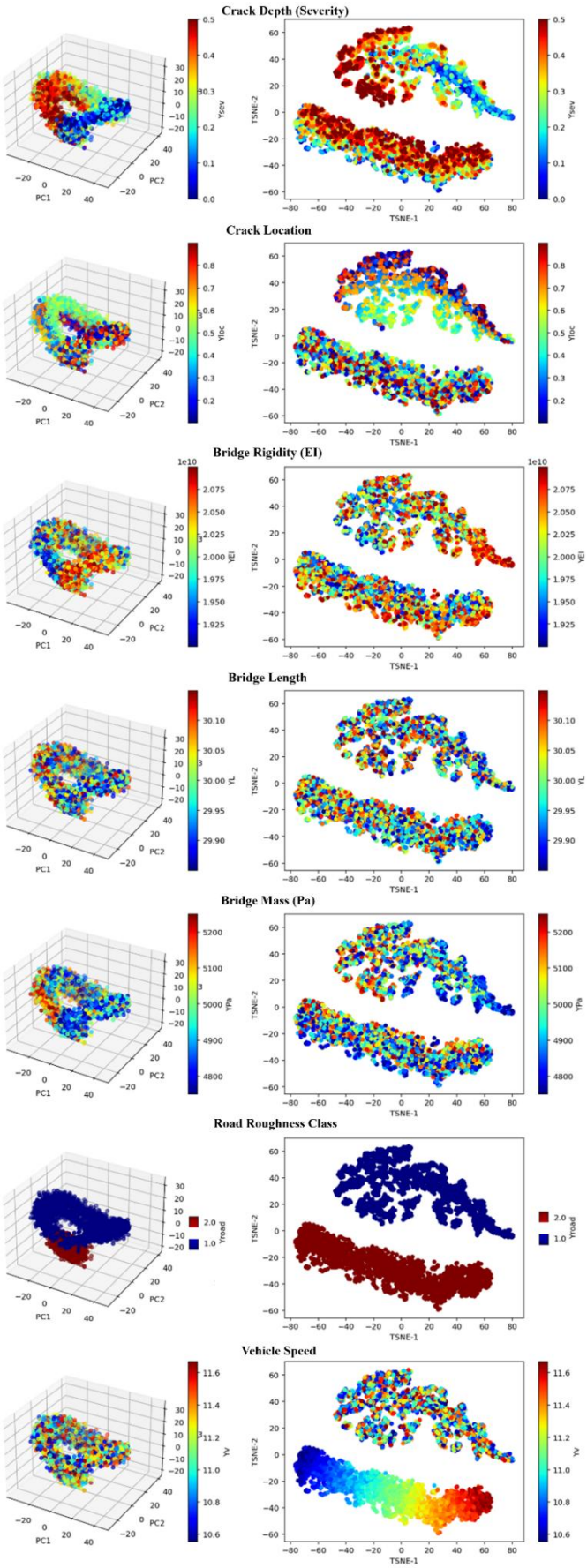


Figure 6.11 The first 3 PCA components (left) and the tSNE representation (right) of the extracted features coloured by the different domain-variation labels.

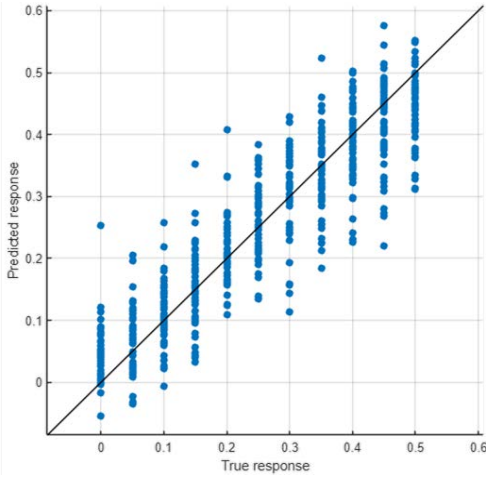


Figure 6.12 Damage severity regression results on the generalised dataset

6.8 Dual-Head Autoencoder for Crack Signature Function Reconstruction

The final section of this chapter is dedicated to a novel methodology for predicting the crack location and depth in a single process. The objective is to utilise the structure of the proposed Autoencoder to extract latent features while simultaneously reconstructing the vehicle bridge acceleration from its own data. Moreover, an additional decoder will be trained concurrently on the latent space features to construct the crack shape as a signature function. This function is derived from the Wahab et al. model, which offers a stiffness reduction function along the concrete beam. The proposed dual-head autoencoder architecture is presented in Figure 6.13. It is designed to reconstruct this signature function across the domain-generalised dataset, which encompasses multichannel records of bridge acceleration and corresponding histories of crack signature functions.

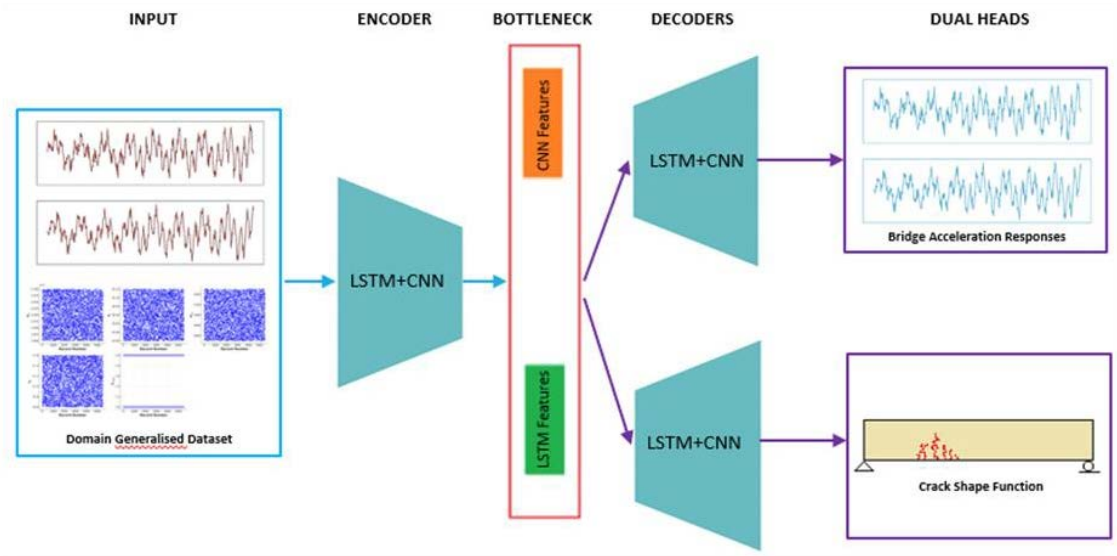


Figure 6.13 The proposed dual-head autoencoder architecture

6.8.1 Crack Signature Function

Wahab et al. (1999) proposed a specified method for simulating the crack zone in a reinforced concrete beam. The core idea of the method is stiffness reduction. Young's modulus of the concrete is assumed to be reduced in a damaged zone due to crack as shown in Figure 6.14. The formulation of this model is as follows:

$$EI(x) = E_0 I\left\{1 - \gamma\cos^2\left[\frac{\pi}{2}\left(\frac{|x-l_c|}{\beta L/2}\right)^n\right]\right\} \quad (l_c - \beta L/2 < x < l_c + \beta L/2) \qquad \text{Eq.6.16}$$

Where E_0 is the Young's modulus of the intact beam and L is the length of the beam. γ, β, and n are the damage parameters. l_c denoted the midpoint location of the damage zone on the beam The paramer β which is defined between 0 and 1, corresponds to the length of the damaged zone, i.e. if damage is in the midspan and β is 1, the damage will cover the entire beam. The parameter γ which is also defined between 0.0 and 1.0 reflects the damage magnitude, e.g. zero means no damage and 1.0 means no bending stiffness in the midpoint of the damaged zone. n illustrates how Young's modulus varies from the centre to the edge of the damage zone.

When applying the stiffness reduction function to the beam, elements that lie outside the damaged region maintain their original rigidity (E_0I). The resulting vector EI (1…N) exhibits the spatial variation of bending stiffness along the span. To align with the time domain of the bridge response, the EI vector undergoes a linear re-interpolation to achieve the same resolution as the acceleration data. Ultimately, the first 500 values are normalised by the intact stiffness to derive a dimensionless crack profile:

$$\text{Ycrack}\ (x) = 1 - \frac{EI(x)}{E_0 I} \qquad \text{Eq.6.17}$$

In the training dataset, the Ycrack function values range from 0 (no stiffness loss) to 0.5 (50% stiffness loss at the crack centre) and serve as the crack signature function.

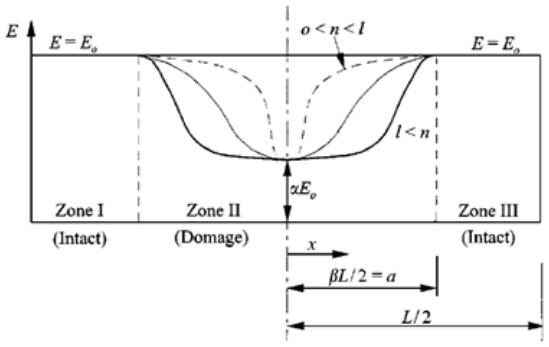


Figure 6.14 Crack stiffness reduction function (Wahab et al. 1999)

6.8.2 Dual-Head Autoencoder Model Description and Training

The autoencoder architecture is mainly similar to the previously described model. However, it features two distinct decoder heads:

1) A head that reconstructs the bridge acceleration response (X_b) , and
2) A head that reconstructs the crack signature function (Ycrack).

The bridge acceleration response is derived from the same two sensors selected in the previous study (sensors 2 and 6). Additionally, the model's loss function has been modified to account for the loss in both reconstruction heads as demonstrated below:

$$\mathcal{L}_{\text{hybrid}} = \left(\alpha_{\text{time}} \cdot \mathcal{L}_{\text{time}} + \alpha_{\text{freq}} \cdot \mathcal{L}_{\text{freq}} + \alpha_{\text{phase}} \cdot \mathcal{L}_{\text{phase}}\right)_{X_b} + \beta\left(\alpha_{\text{time}} \cdot \mathcal{L}_{\text{time}} + \alpha_{\text{freq}} \cdot \mathcal{L}_{\text{freq}} + \alpha_{\text{phase}} \cdot \mathcal{L}_{\text{phase}}\right)_{Y_{Crack}} \quad \text{Eq.6.18}$$

The model hyperparameters are being fine-tuned using Optuna. The optimised weight decay and learning rate were determined to be 2.5e-4 and 3.6e-6, respectively. Additionally, the detailed model architecture, along with its hyperparameters, is presented in Table 6.6.

The model was trained on the domain-generalised dataset. The training and validation loss curves are depicted in Figure 6.15.

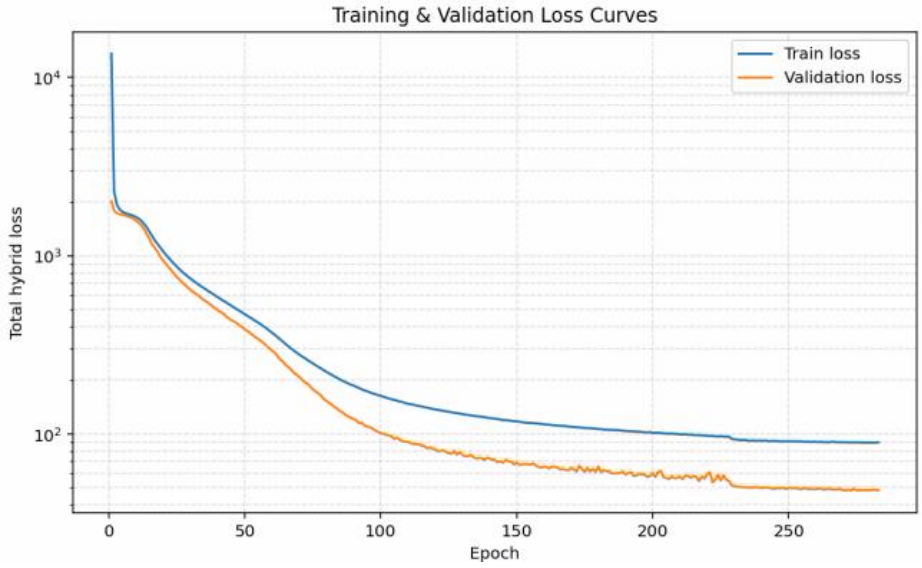


Figure 6.15 Total Train and validation loss curves for the dual-head AE

Table 6.6 Dual-head AE model description

Category	Hyperparameter / Property	Value / Description
Windowing	Window size	500 samples
Data loader	Batch size	64
Encoder - CNN branch	Convolution layers	3x Conv1D
	Kernels per layer	48 filters
	Kernel size	3 (padding =1)
	Dropout	0.02
Encoder - LSTM branch	Layers	3
	Hidden units per layer	1000
	Dropout	0.47
Latent code	Dimension	2000(1000+1000)
Decoder (two heads)	Dense expansion	2000 → 384 x 500
	LSTM in decoder	3 layers, 384 units, dropout 0.47
	Transposed-conv block 1	384 → 192, kernel =3
	Transposed-conv block 2	192 → 96, kernel =3
	Transposed-conv block 3	96 → 96, kernel =3
	Output layer	96 → 2(Xb) or 1 (Ycrack), kernel =3
Training	Learning rate	2.5 e-4
	Weight decay	3.6 e-6
	Number of epochs	500
	Early stopping patience	10
Loss - Xb weights	Alpha time	1.74
	Alpha freq	1.72
	Alpha Phase	0.84
Loss - Ycrack weights	Alpha time	2.87
	Alpha freq	1.11
	Alpha Phase	2.99
	Beta	2.8

6.8.3 Crack Signature Function Reconstruction Results

The dual-head autoencoder generates two outputs: the reconstructed bridge-acceleration signals (Xb), and the crack signature function (Ycrack). After training of the model on the domain-generalised dataset and testing with a test dataset generated utilising a distinct random feed, it was observed that for the unseen test dataset, the mean SSIM for the reconstruction of Xb is 0.9949, and the mean SSIM for the reconstruction of Ycrack is 0.8290.

Figures 6.16 to 6.21 illustrate representative examples of the reconstruction results. It is evident that even when the crack magnitude is as low as 0.1 (representing a 10% reduction in stiffness), the model effectively captures both the shape and location of the crack.

These promising results clearly indicate that the proposed approach effectively reconstructs both high-resolution dynamic responses and intricate damage characteristics, thereby enabling the identification of reduced stiffness and well-localised damage, even in the presence of various domain shifts within the training data.

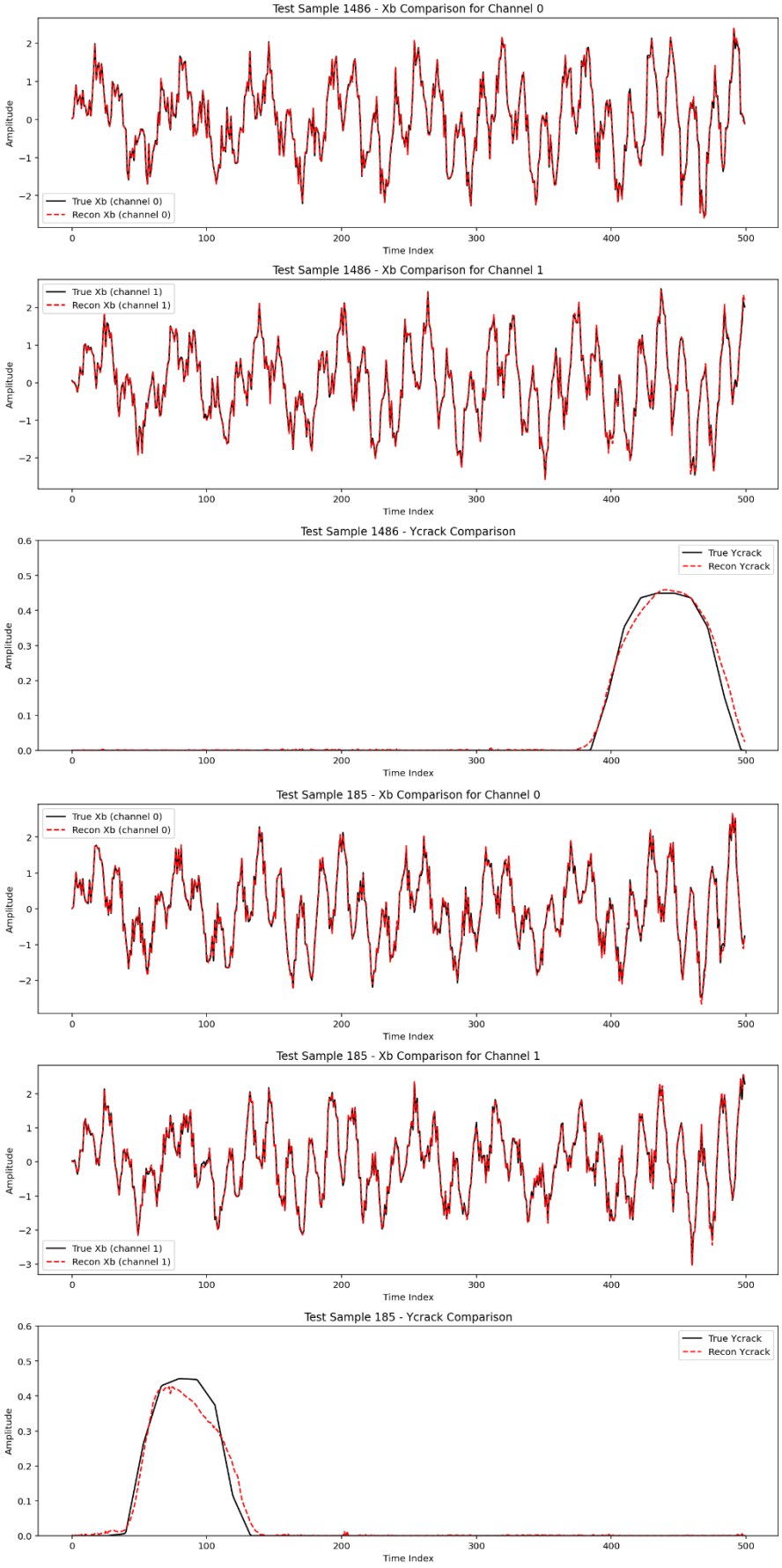


Figure 6.16 Examples of the dual-head AE reconstruction results

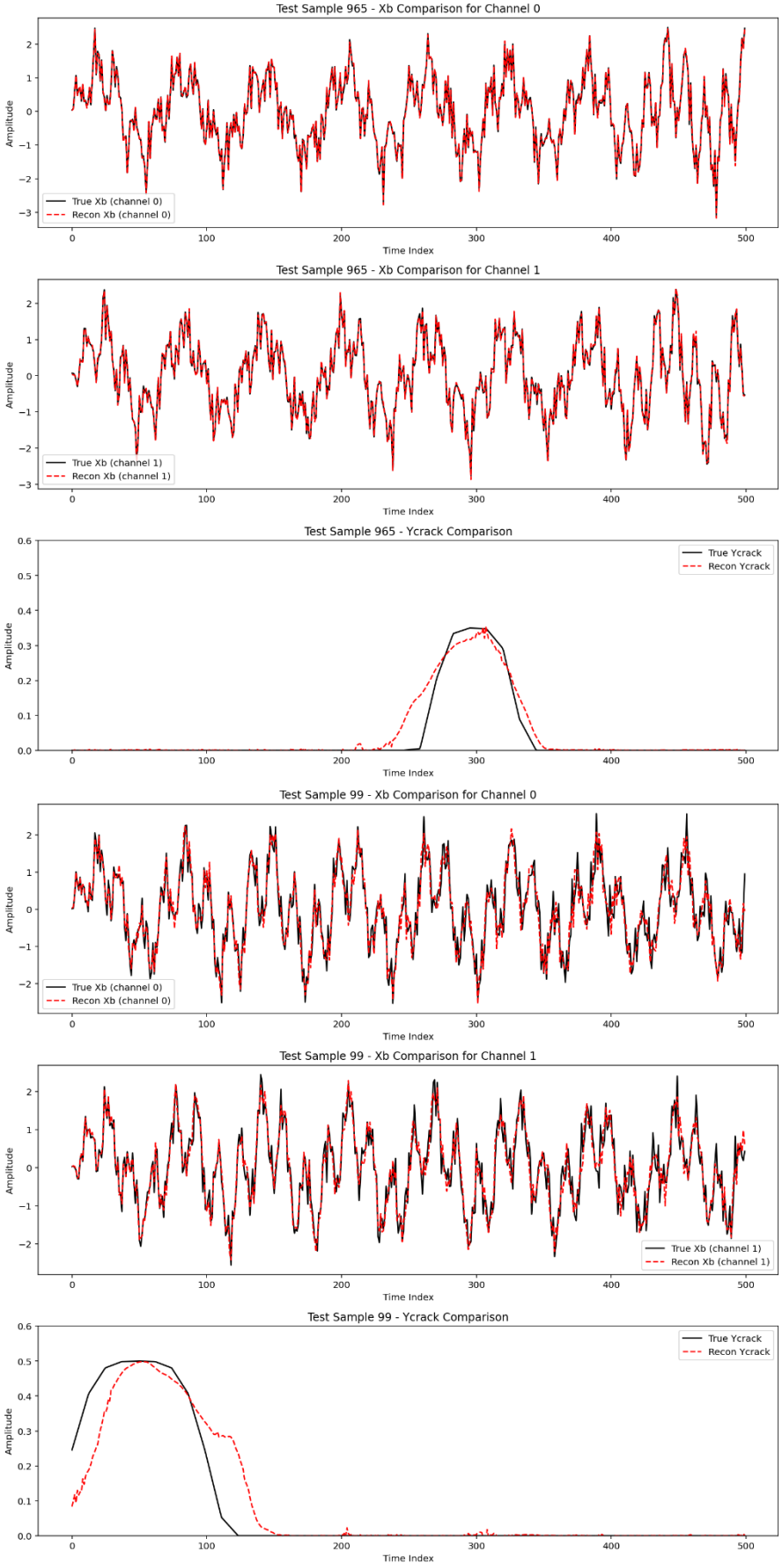


Figure 6.17 Examples of the dual-head AE reconstruction results

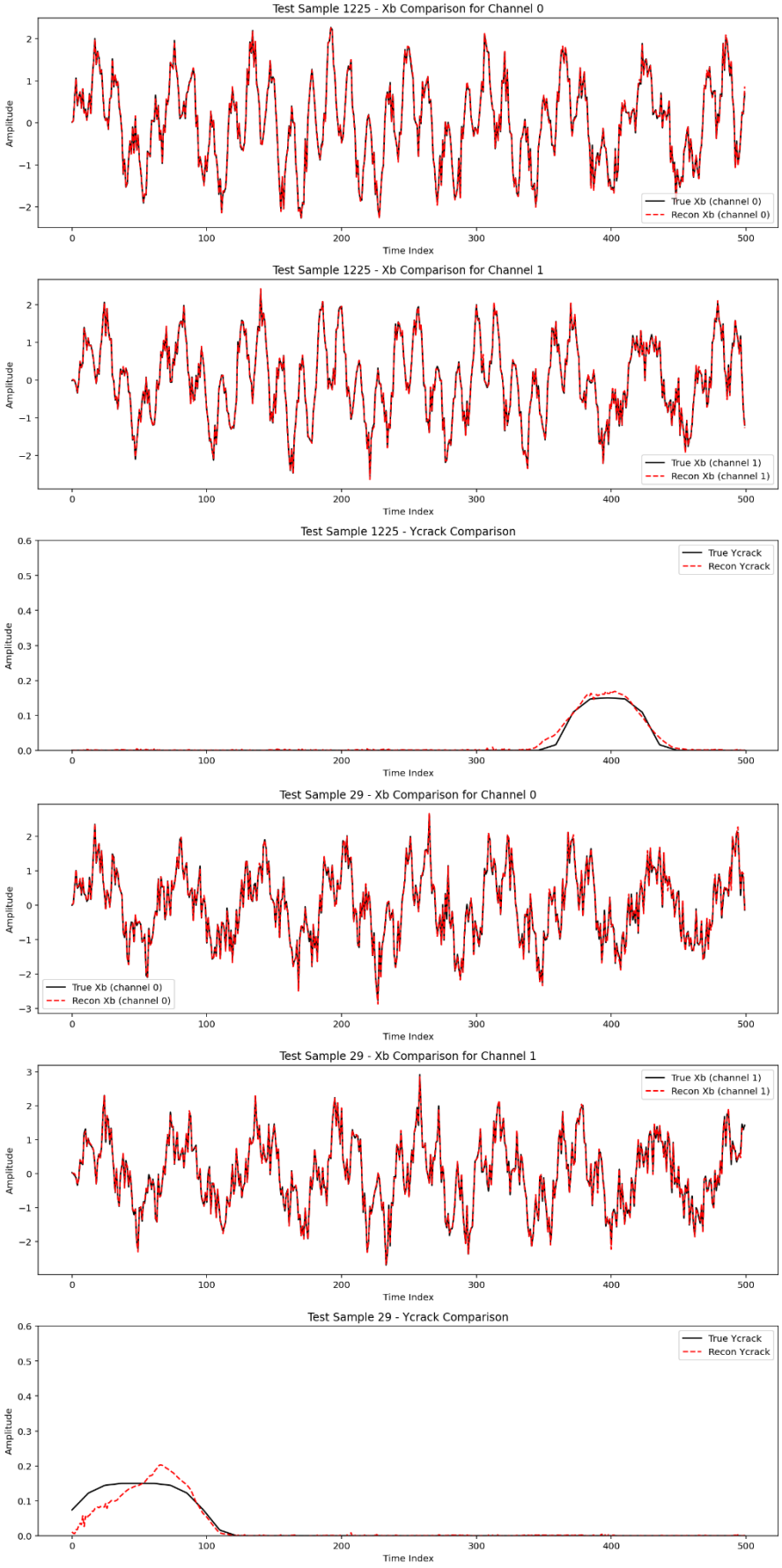


Figure 6.18 Examples of the dual-head AE reconstruction results

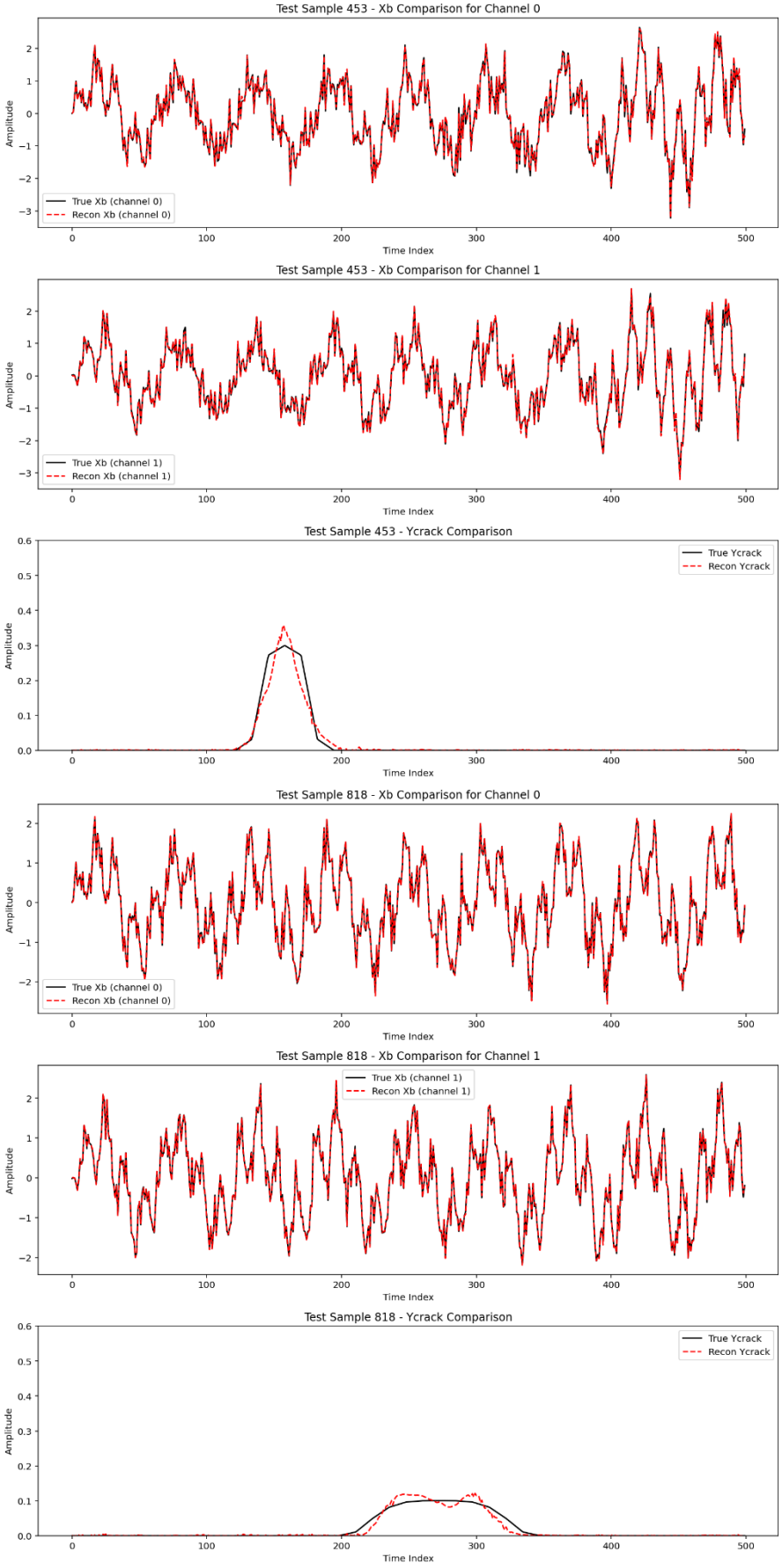


Figure 6.19 Examples of the dual-head AE reconstruction results

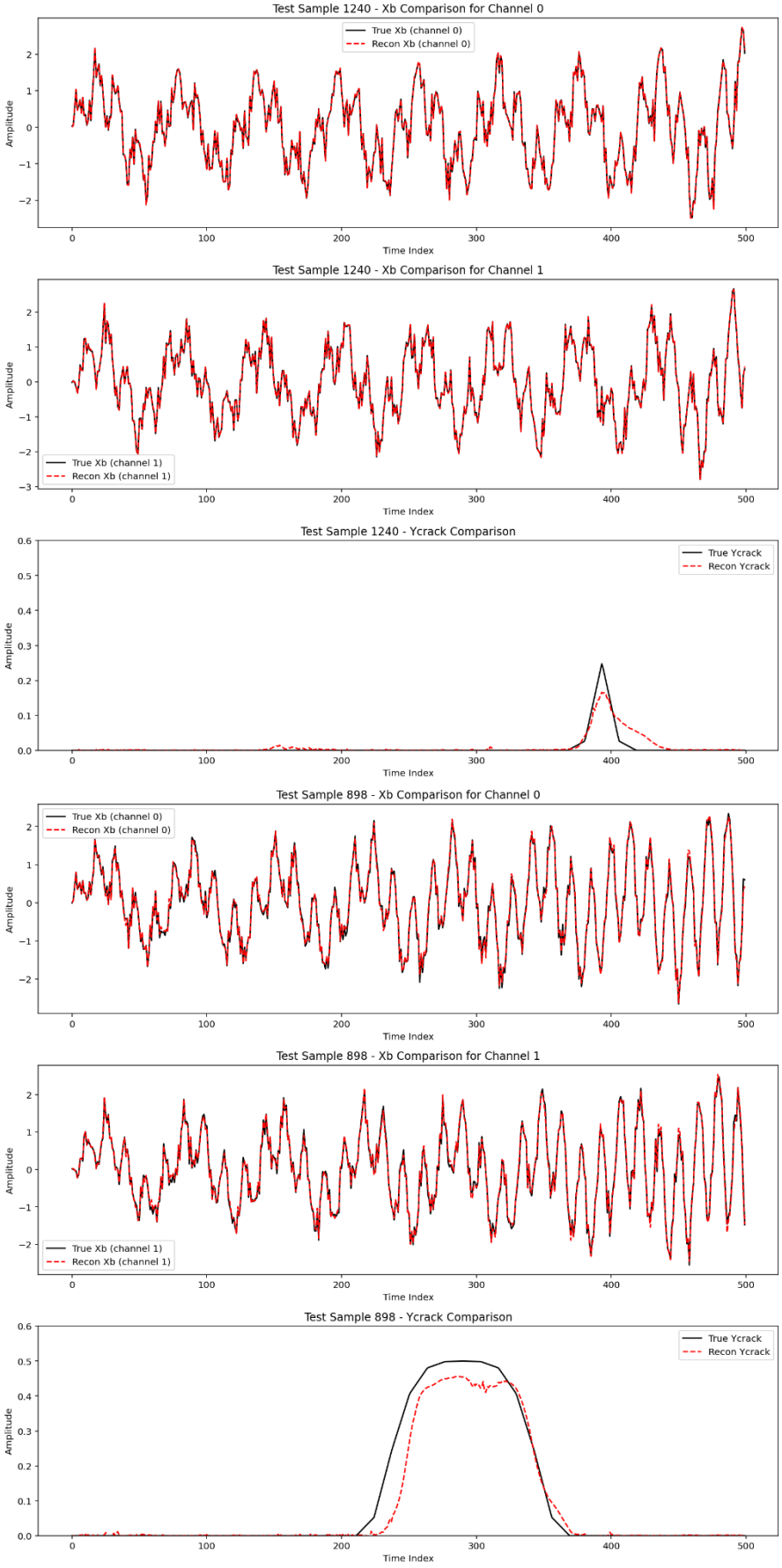


Figure 6.20 Examples of the dual-head AE reconstruction results

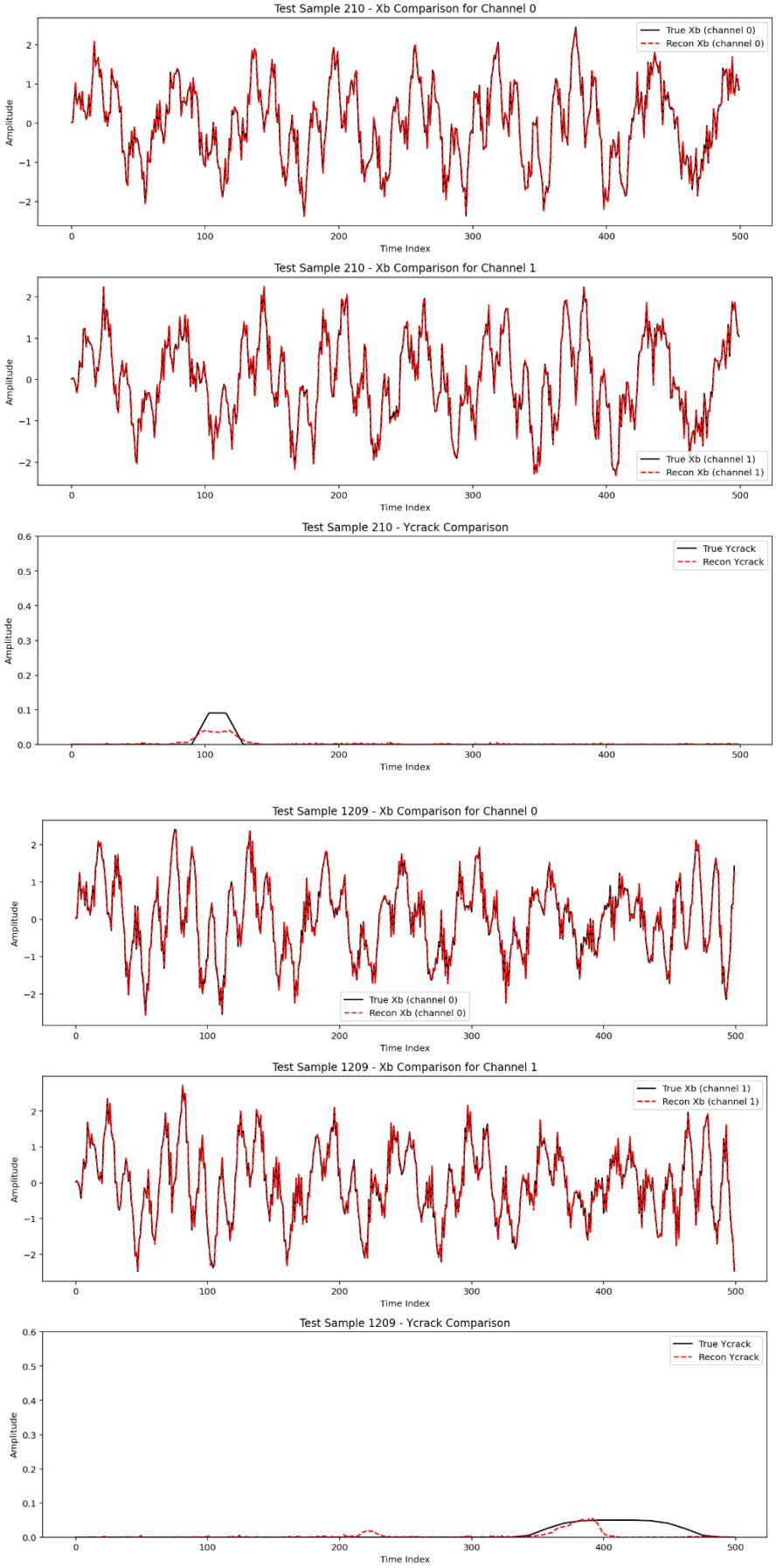


Figure 6.21 Examples of the dual-head AE reconstruction results

6.9 Conclusion and Discussion

The comparison of CNN, LSTM, and CNN+LSTM encoders demonstrated that the hybrid model achieves a better reconstruction accuracy (MSE ~14.3×10^{-3}) with a 23% improvement over CNN, which suggests that CNNs and RNNs capture different aspects of time-series data. The hybrid AE's high fidelity (SSIM ~0.90) shows that VBI acceleration responses can be accurately reconstructed using the CNN+LSTM architecture for the AE. The hybrid loss (time + frequency + phase) played a crucial role in training the autoencoder to achieve high accuracy. Without it, initial models would reconstruct the general trend but miss sharp peaks or be slightly phase-shifted. The hybrid loss compelled the model to align the signals in all aspects, resulting in more accurate reconstructions.

While the autoencoder was trained in an unsupervised manner, its latent features demonstrate a strong correlation with damage severity, indicating that AE-based unsupervised feature extraction from VBI acceleration responses can yield damage-sensitive features. By employing these features, an SVM regressor was able to predict crack depth within an approximate range of ±10%. However, the regression accuracy dropped to ±20% under domain shift configuration, suggesting that while the domain generalisation strategy strengthens feature robustness across many modelling variations, it reduces the damage sensitivity. By training on a range of simulated scenarios (varying bridge and vehicle parameters), the proposed autoencoder achieved robustness to these variations, as evident in the latent features not clustering by bridge stiffness or mass in t-SNE plots. Nevertheless, simulations can never capture all real-world effects; for example, environmental factors like temperature were not included in the generalised dataset. In real bridges, these could act as new domain shifts. Therefore, while our model shows robustness to many modelled uncertainties, further generalisation might be needed for damage detection in real scenarios.

The introduction of a dual-head autoencoder that outputs a crack signature function is a step beyond traditional damage detection, which typically provides the crack severity as the output of a regression task and the crack location as the output of a classification task. The proposed model learns to map the VBI acceleration data to the crack profile. The accurate reconstruction of crack shapes (SSIM ≈ 0.83) even for low-severity damage is a remarkable result, especially when the model’s performance held across different domain

configurations, meaning it could identify a crack on a slightly shorter or longer or heavier bridge just as well. This robustness is attributed to the domain-generalised training, which is critical for real-world applications where exact bridge parameters are uncertain.

In conclusion, the proposed approach integrates several concepts, including drive-by monitoring, deep autoencoders, and domain generalisation, to enable *damage identification directly from VBI acceleration data*. It shows that with sufficient training diversity, a deep learning model can generalise to structural modelling errors and operating conditions, addressing a significant gap in structural damage detection. For real-world implementation, the proposed approach should be tested and validated with experimental data. Furthermore, applying the approach to various damage scenarios (multiple cracks) or different damage types would enhance its applicability.

6.10 Summary

This chapter presented a domain-generalised deep AE-based framework for bridge damage detection using VBI acceleration responses. The main innovation is a dual-output autoencoder that reconstructs the bridge acceleration response and the crack profile. The findings of this chapter are summarised below:

- A hybrid AE model with parallel CNN and LSTM encoders accurately reconstructs bridge responses from the vehicle acceleration data, capturing both local frequency content and global temporal patterns.
- Latin Hypercube Sampling is used for creating a domain-generalised dataset encompassing varying bridge length, stiffness, mass, vehicle speed, and road roughness. Training on this dataset enabled the autoencoder to extract domain-invariant, damage-sensitive features.
- **A** hybrid loss (combining time-domain MSE, frequency-domain error, and phase error) is employed for better signal reconstruction with an SSIM of more than 0.9.
- **T**he proposed dual-head autoencoder is able to predict the crack signature function along with the bridge vibration signals. The optimised model achieved an SSIM of 0.9949 for bridge acceleration reconstruction and 0.8290 for crack signature reconstruction on the test set.
- Importantly, the proposed approach successfully maintained its damage detection performance in the presence of various domain shifts (e.g., ±5% changes in the

bridge structural parameters, vehicle speed, and different road roughness classes). This robustness to domain shifts can eliminate the need for model updating and calibrations.

CHAPTER 7 - CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

This thesis aimed to develop data efficient, transfer-learning (TL) approaches for AI-driven bridge condition assessment using vehicle-bridge-interaction (VBI) responses. A variety of bridge condition assessment tasks from added mass detection, prestressed force identification and crack location, depth and shape estimation is covered in this research.

The domain shifts caused by modelling errors are almost inevitable. However, The transfer learning techniques were adopted to overcome the domain shift challenge in the condition assessment in the presence of simulation-to-real world domain-shift.

The proposed approaches were introduced based on the level of data scarcity in the target domain:

1- **With limited labelled data for each class:** A fine-tuning method as a parameter-based TL technique was proposed and validated to transfer the knowledge between the source and target domains on a classification task for added mass location detection. A CNN model was first trained on a source FE model for added mass detection, and fine-tuned for the target FE model with one available data point in each specific label. The classification results show that the model can learn the damage-sensitive feature patterns even with a small number of available data in the target domain.

2- **With limited labelled data:** A Hybrid Domain Adaptation approach (combined maximum-mean-discrepancy and adversarial learning) was proposed for the prediction of the remaining prestressed force in bridge structures, while the only available data from the target domain corresponds to the fully prestressed condition (PF=100%). The method was applied in several domain shift scenarios, including the variations in the bridge and vehicle parameters and road roughness.

The proposed approach successfully mitigates domain shifts caused by modelling errors, even in unseen domains where further domain shift occurs on the data after domain adaptation.

3- **Without labelled data from the target domain:** A domain generalisation approach was proposed and adopted for crack detection in concrete bridges. A dual-head autoencoder was employed to reconstruct the crack shape signature function from the bridge acceleration responses in the time domain. The domain generalised dataset was generated using the Latin Hypercube Sampling method to cover a range of parameter variations that can cause domain shifts due to the modelling error. The target domain parameters are assumed not to be fixed and can vary within the defined ranges. The crack shape reconstruction results show a promising accuracy in detecting the crack location, depth, length and shape, eliminating the domain shift effect.

Two main feature extraction methods were proposed and applied in the current study to extract damage-sensitive features from the VBI responses:

1) Pretrained CNN networks such as CNNs (AlexNet, ResNet-50, VGG-16, and GoogLeNet) proved to be powerful in feature extraction from time-frequency domain representation of the signals. AlexNet showed superior performance compared to the other networks under various signal noise levels, sensor locations, road roughness, and vehicle speeds.
2) Coupled CNN+LSTM autoencoders were utilised in the last chapter to extract damage-sensitive features from the VBI acceleration responses in an unsupervised learning approach. The extracted features demonstrated an acceptable level of robustness in the presence of a range of domain shifts.

7.2 Recommendations for Future Works

Although the proposed approach demonstrated a promising outcome in bridging the gap between the source and target domains, certain limitations persist in the current work, and consequently, a number of recommendations are presented for future works:

1- **Experimental validation:** The proposed methods aimed to eliminate the domain shift between FE models and real structures. However, due to limitations during the completion of this thesis, the proposed approaches were not validated using experimental data, which is an essential step before making any further claims about the robustness of the proposed approaches.

2- **Online learning:** Since the proposed methods can leverage any recorded data in the target domain to improve their performance, any online recording of the bridge acceleration can be utilised not only for the online health monitoring but to improve the model accuracy and adaptation measurement noise, temperature variation, slow creep-shrinkage and gradual road surface wear.

3- **Physics-informed regularisation:** Embedding mode-shape orthogonality or equilibrium constraints into the loss terms could further improve the model performance.

4- **Large language models / Multimodal models fine-tuning:** leveraging the large language models and multimodal models (e.g. Chat-GPT and Gemini) with billions of learnt parameters can transfer their advanced representation-learning and cross-modal reasoning capabilities to the bridge condition assessment domain.

5- **Network-level decision support:** Integrating PF estimates and crack-shape signatures into reliability-based maintenance planning allows road agencies to prioritise interventions and lower bridge-maintenance costs.

REFERENCES

Adhikary, B. B., & Mutsuyoshi, H. (2004). Artificial neural networks for the prediction of shear capacity of steel plate strengthened RC beams. *Construction and Building Materials*, *18*(6), 409-417. https://doi.org/10.1016/j.conbuildmat.2004.03.002

Akbulut, Y., Sengur, A., Guo, Y. H., & Smarandache, F. (2017). NS-k-NN: Neutrosophic Set-Based k-Nearest Neighbors Classifier. *Symmetry-Basel*, *9*(9), Article 179. https://doi.org/10.3390/sym9090179

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining,

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., & Inman, D. J. (2021). A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mechanical Systems and Signal Processing*, *147*, 107077.

Azimi, M., & Pekcan, G. (2020). Structural health monitoring using extremely compressed data through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *35*(6), 597-614. https://doi.org/10.1111/mice.12517

Bacsa, K., Liu, W., Abdallah, I., & Chatzi, E. (2025). Structural Dynamics Feature Learning Using a Supervised Variational Autoencoder. *Journal of Engineering Mechanics*, *151*(2), 04024106.

Balafas, K., & Kiremidjian, A. S. (2015). Development and validation of a novel earthquake damage estimation scheme based on the continuous wavelet transform of input and output acceleration measurements. *Earthquake Engineering & Structural Dynamics*, *44*(4), 501-522.

Betti, M., Facchini, L., & Biagini, P. (2015). Damage detection on a three-storey steel frame using artificial neural networks and genetic algorithms. *Meccanica*, *50*(3), 875-886. https://doi.org/10.1007/s11012-014-0085-9

Bo, L. F., & Sminchisescu, C. (2010). Twin Gaussian Processes for Structured Prediction. *International Journal of Computer Vision*, *87*(1-2), 28-52. https://doi.org/10.1007/s11263-008-0204-y

Boonyakitanont, P., Lek-uthai, A., Chomtho, K., & Songsiri, J. (2020). A review of feature extraction and performance evaluation in epileptic seizure detection using EEG. *Biomedical Signal Processing and Control*, *57*, Article 101702. https://doi.org/10.1016/j.bspc.2019.101702

Bornn, L., Farrar, C. R., Park, G., & Farinholt, K. (2009). Structural Health Monitoring With Autoregressive Support Vector Machines. *Journal of Vibration and Acoustics-Transactions of the Asme*, *131*(2), Article 021004. https://doi.org/10.1115/1.3025827

Bu, J., Law, S., & Zhu, X. (2006). Innovative bridge condition assessment from dynamic response of a passing vehicle. *Journal of Engineering Mechanics*, *132*(12), 1372-1379.

Bull, L. A., Gardner, P. A., Dervilis, N., Papatheou, E., Haywood-Alexander, M., Mills, R. S., & Worden, K. (2021). On the transfer of damage detectors between structures: An experimental case study. *Journal of Sound and Vibration*, *501*, Article 116072. https://doi.org/10.1016/j.jsv.2021.116072

Bull, L. A., Gardner, P. A., Gosliga, J., Rogers, T. J., Dervilis, N., Cross, E. J., Papatheou, E., Maguire, A. E., Campos, C., & Worden, K. (2021). Foundations of population-based SHM, Part I: Homogeneous populations and forms. *Mechanical Systems and Signal Processing*, *148*, Article 107141. https://doi.org/10.1016/j.ymssp.2020.107141

Cantero, D., Sarwar, Z., Malekjafarian, A., Corbally, R., Alamdari, M. M., Cheema, P., Aggarwal, J., Noh, H. Y., & Liu, J. (2024). Numerical benchmark for road bridge damage detection from passing vehicles responses applied to four data-driven methods. *Archives of Civil and Mechanical Engineering*, *24*(3), 190.

Cexus, J.-C., Boudraa, A.-O., & Bouchikhi, A. (2010). A combined Teager-Huang and Hough transforms for LFM signals detection. 2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP),

Cha, Y. J., & Wang, Z. L. (2018). Unsupervised novelty detection-based structural damage localization using a density peaks-based fast clustering algorithm. *Structural Health Monitoring-an International Journal*, *17*(2), 313-324. https://doi.org/10.1177/1475921717691260

Chen, Y., Feng, M. Q., & Tan, C.-A. (2009). Bridge structural condition assessment based on vibration and traffic monitoring. *Journal of Engineering Mechanics*, *135*(8), 747-758.

Chen, Z. P., Pan, C. D., & Yu, L. (2018). Structural damage detection via adaptive dictionary learning and sparse representation of measured acceleration responses. *Measurement*, *128*, 377-387. https://doi.org/10.1016/j.measurement.2018.06.046

Chong, J. W., Kim, Y., & Chon, K. H. (2014). Nonlinear multiclass support vector machine-based health monitoring system for buildings employing magnetorheological dampers. *Journal of Intelligent Material Systems and Structures*, *25*(12), 1456-1468. https://doi.org/10.1177/1045389x13507343

Cortes, C., & Vapnik, V. (1995). SUPPORT-VECTOR NETWORKS. *Machine Learning*, *20*(3), 273-297. https://doi.org/10.1023/a:1022627411411

Crivelli, D., Guagliano, M., & Monici, A. (2014). Development of an artificial neural network processing technique for the analysis of damage evolution in pultruded composites with acoustic emission. *Composites Part B-Engineering*, *56*, 948-959. https://doi.org/10.1016/j.compositesb.2013.09.005

Cury, A., & Cremona, C. (2012). Pattern recognition of structural behaviors based on learning algorithms and symbolic data concepts. *Structural Control & Health Monitoring*, *19*(2), 161-186. https://doi.org/10.1002/stc.412

de Lautour, O. R., & Omenzetter, P. (2009). Prediction of seismic-induced structural damage using artificial neural networks. *Engineering Structures*, *31*(2), 600-606. https://doi.org/10.1016/j.engstruct.2008.11.010

Demirlioglu, K., & Erduran, E. (2024). Drive-by bridge damage detection using continuous wavelet transform. *Applied Sciences*, *14*(7), 2969.

Etem, T. (2025). Advancing non-destructive weld assessment using autoencoder-based feature extraction. *Nondestructive Testing and Evaluation*, 1-21.

Fallahian, M., Khoshnoudian, F., & Talaei, S. (2018). Application of couple sparse coding ensemble on structural damage detection. *Smart Structures and Systems*, *21*(1), 1-14. https://doi.org/10.12989/sss.2018.21.1.001

Figueiredo, E., Park, G., Farrar, C. R., Worden, K., & Figueiras, J. (2011). Machine learning algorithms for damage detection under operational and environmental variability. *Structural Health Monitoring-an International Journal*, *10*(6), 559-572. https://doi.org/10.1177/1475921710388971

Fugate, M. L., Sohn, H., Farrar, C. R., Sem, & Sem. (2000, Feb 07-10). Unsupervised learning methods for vibration-based damage detection.*Proceedings of the Society of Photo-Optical Instrumentation Engineers (Spie)* [Imac-xviii: A conference on structural dynamics, vols 1 and 2, proceedings]. 18th IMAC Conference on Computational Challenges in Structural Dynamics, San Antonio, Tx.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. International conference on machine learning,

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748-768.
Gardner, P., Bull, L. A., Gosliga, J., Dervilis, N., & Worden, K. (2021). Foundations of population-based SHM, Part III: Heterogeneous populations - Mapping and transfer. *Mechanical Systems and Signal Processing*, *149*, Article 107142. https://doi.org/10.1016/j.ymssp.2020.107142
Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., & Li, W. (2016). Deep reconstruction-classification networks for unsupervised domain adaptation. European conference on computer vision,
Gonzalez, M. P., & Zapico, J. L. (2008). Seismic damage identification in buildings using neural networks and modal data. *Computers & Structures*, *86*(3-5), 416-426. https://doi.org/10.1016/j.compstruc.2007.02.021
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.
Gosliga, J., Gardner, P. A., Bull, L. A., Dervilis, N., & Worden, K. (2021). Foundations of Population-based SHM, Part II: Heterogeneous populations - Graphs, networks, and communities. *Mechanical Systems and Signal Processing*, *148*, Article 107144. https://doi.org/10.1016/j.ymssp.2020.107144
Gu, X. Q., Zhang, C., & Ni, T. G. (2021). A Hierarchical Discriminative Sparse Representation Classifier for EEG Signal Detection. *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, *18*(5), 1679-1687. https://doi.org/10.1109/tcbb.2020.3006699
Guan, H., & Liu, M. (2021). Domain adaptation for medical image analysis: a survey. *IEEE Transactions on Biomedical Engineering*, *69*(3), 1173-1185.
Hansen, L. K., & Salamon, P. (1990). NEURAL NETWORK ENSEMBLES. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993-1001. https://doi.org/10.1109/34.58871
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition,
Hermans, M., & Schrauwen, B. (2013). Training and analysing deep recurrent neural networks. *Advances in neural information processing systems*, *26*.
Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504-507. https://doi.org/10.1126/science.1127647
HoThu, H., & Mita, A. (2013). Damage detection method using support vector machine and first three natural frequencies for shear structures.
Huang, L., Hong, X., Yang, Z., Liu, Y., & Zhang, B. (2022). CNN-LSTM network-based damage detection approach for copper pipeline using laser ultrasonic scanning. *Ultrasonics*, *121*, 106685.
Jiang, K., Han, Q., Du, X., & Ni, P. (2021). A decentralized unsupervised structural condition diagnosis approach using deep auto-encoders. *Computer-Aided Civil and Infrastructure Engineering*, *36*(6), 711-732.
Khazaeli, S., Nguyen, L. H., & Goulet, J. A. (2021). Anomaly detection using state-space models and reinforcement learning. *Structural Control and Health Monitoring*, *28*(6), e2720.
Khoshnoudian, F., & Talaei, S. (2017). A New Damage Index Using FRF Data, 2D-PCA Method and Pattern Recognition Techniques. *International Journal of Structural Stability and Dynamics*, *17*(8), Article 1750090. https://doi.org/10.1142/s0219455417500900
Kim, J., Lynch, J. P., Lee, J.-J., & Lee, C.-G. (2011). Truck-based mobile wireless sensor networks for the experimental observation of vehicle-bridge interaction. *Smart Materials and Structures*, *20*(6), 065009.
Kislov, K., & Gravirov, V. (2017). Use of artificial neural networks for classification of noisy seismic signals. *seismic Instruments*, *53*, 87-101.
Koune, I., & Cicirello, A. (2024). Disentangled representation learning with physics-informed variational autoencoder for structural health monitoring.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the Acm*, *60*(6), 84-90. https://doi.org/10.1145/3065386

Law, S. S., & Zhu, X. Q. (2004). Dynamic behavior of damaged concrete bridge structures under moving vehicular loads. *Engineering Structures*, *26*(9), 1279-1293.

Law, S. S., & Zhu, X. Q. (2011). Moving Loads-Dynamic Analysis and Identification Techniques Introduction. *Structures and Infrastructures Book Series*, *8*, 1-24. https://doi.org/https://doi.org/10.1201/b10561

Lee, K., Chung, M., & Kim, S. (2021). Damage detection of catenary mooring line based on recurrent neural networks. *Ocean Engineering*, *227*, 108898.

Li, B., Wang, Y., Zhang, S., Li, D., Keutzer, K., Darrell, T., & Zhao, H. (2021). Learning invariant representations and risks for semi-supervised domain adaptation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Li, J., Guo, J., & Zhu, X. (2021). Time-varying parameter identification of bridges subject to moving vehicles using ridge extraction based on empirical wavelet transform. *International Journal of Structural Stability and Dynamics*, *21*(04), 2150046.

Li, J., Law, S., & Hao, H. (2013). Improved damage identification in bridge structures subject to moving loads: numerical and experimental studies. *International Journal of Mechanical Sciences*, *74*, 99-111.

Li, J., Li, Z., & Lü, S. (2021). Feature concatenation for adversarial domain adaptation. *Expert Systems with Applications*, *169*, 114490.

Li, J. T., Zhu, X. Q., Law, S. S., & Samali, B. (2020). Time-varying characteristics of bridges under the passage of vehicles using synchroextracting transform. *Mechanical Systems and Signal Processing*, *140*, Article 106727. https://doi.org/10.1016/j.ymssp.2020.106727

Li, S., Cao, Y., Gdoutos, E. E., Tao, M., Alkayem, N. F., Avci, O., & Cao, M. (2024). Intelligent framework for unsupervised damage detection in bridges using deep convolutional autoencoder with wavelet transmissibility pattern spectra. *Mechanical Systems and Signal Processing*, *220*, 111653.

Li, X., Bolandi, H., Masmoudi, M., Salem, T., Jha, A., Lajnef, N., & Boddeti, V. N. (2024). Mechanics-informed autoencoder enables automated detection and localization of unforeseen structural damage. *Nature Communications*, *15*(1), 9229.

Li, Z., Lin, W., & Zhang, Y. (2023). Real-time drive-by bridge damage detection using deep auto-encoder. Structures,

Lin, Y. z., Nie, Z. h., & Ma, H. w. (2017). Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *32*(12), 1025-1046.

Liu, C., Chen, Y., Xu, X., & Che, W. (2024). Domain generalization-based damage detection of composite structures powered by structural digital twin. *Composites Science and Technology*, *258*, 110908.

Liu, T., Xu, H., Ragulskis, M., Cao, M., & Ostachowicz, W. (2020). A data-driven damage identification framework based on transmissibility function datasets and one-dimensional convolutional neural networks: Verification on a structural health monitoring benchmark structure. *Sensors*, *20*(4), 1059.

Mandic, D. P., Ur Rehman, N., Wu, Z., & Huang, N. E. (2013). Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis. *IEEE signal processing magazine*, *30*(6), 74-86.

Mariniello, G., Pastore, T., Menna, C., Festa, P., & Asprone, D. (2021). Structural damage detection and localization using decision tree ensemble and vibration data. *Computer-Aided Civil and Infrastructure Engineering*, *36*(9), 1129-1149. https://doi.org/10.1111/mice.12633

McGetrick, P. J., Kim, C.-W., González, A., & Brien, E. J. (2015). Experimental validation of a drive-by stiffness identification method for bridge monitoring. *Structural Health Monitoring*, *14*(4), 317-331.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*(1), 55-61.

Mehrjoo, M., Khaji, N., Moharrmi, H., & Bahreininejad, A. (2008). Damage detection of truss bridge joints using Artificial Neural Networks. *Expert Systems with Applications*, *35*(3), 1122-1131. https://doi.org/10.1016/j.eswa.2007.08.008

Mihalkova, L., & Mooney, R. J. (2008). Transfer learning by mapping with minimal target data. Proceedings of the AAAI-08 workshop on transfer learning for complex tasks,

Mojaddady, M., Nabi, M., & Khadivi, S. (2011). Stock market prediction using twin Gaussian process regression. *International Journal for Advances in Computer Research (JACR) preprint*.

Mousavi, M., Holloway, D., Olivier, J., & Gandomi, A. H. (2021). A baseline-free damage detection method using VBI incomplete measurement data. *Measurement*, *174*, 108957.

Noi, P. T., & Kappas, M. (2018). Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. *Sensors*, *18*(1), Article 18. https://doi.org/10.3390/s18010018

Normung, I. O. f. (1995). *ISO 8608 : Mechanical Vibration, Road Surface Profiles, Reporting of Measured Data: International Standard*. International Organization for Standardization, ISO. https://books.google.com.au/books?id=1kyioAEACAAJ

Opitz, D. W., & Shavlik, J. W. (1996). Actively searching for an effective neural network ensemble. *Connection Science*, *8*(3-4), 337-354.

Pan, H., Azimi, M., Yan, F., & Lin, Z. (2018). Time-frequency-based data-driven structural diagnosis and damage detection for cable-stayed bridges. *Journal of Bridge Engineering*, *23*(6), 04018033.

Pan, Q., Zhang, L., Dai, G., & Zhang, H. (1999). Two denoising methods by wavelet transform. *IEEE transactions on signal processing*, *47*(12), 3401-3406.

Pan, S. J., & Yang, Q. A. (2010). A Survey on Transfer Learning. *Ieee Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359. https://doi.org/10.1109/tkde.2009.191

Panareda Busto, P., & Gall, J. (2017). Open set domain adaptation. Proceedings of the IEEE international conference on computer vision,

Pathirage, C. S. N., Li, J., Li, L., Hao, H., Liu, W. Q., & Ni, P. H. (2018). Structural damage identification based on autoencoder neural networks and deep learning. *Engineering Structures*, *172*, 13-28. https://doi.org/10.1016/j.engstruct.2018.05.109

Patle, A., & Chouhan, D. S. (2013). SVM kernel functions for classification. 2013 International conference on advances in technology and engineering (ICATE),

Patle, A., Chouhan, D. S., & Ieee. (2013, Jan 23-25). SVM Kernel Functions for Classification. [2013 international conference on advances in technology and engineering (icate)]. International Conference on Advances in Technology and Engineering (ICATE), Mumbai, INDIA.

Perrone, M. P., & Cooper, L. N. (1992). *When networks disagree: Ensemble methods for hybrid neural networks*.

Poole, J., Gardner, P., Dervilis, N., Bull, L., & Worden, K. (2022). On statistic alignment for domain adaptation in structural health monitoring. *Structural Health Monitoring-an International Journal*. https://doi.org/10.1177/14759217221110441

Radhika, K., & Varadarajan, S. (2018). Ensemble subspace discriminant classification of satellite images. *Journal of Scientific & Industrial Research*, *77*(1), 633-638.

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598-607. https://doi.org/10.1016/j.engstruct.2017.10.070

Rasmussen, C. E. (2004). Gaussian processes in machine learning. In O. Bousquet, U. VonLuxburg, & G. Ratsch (Eds.), *Advanced Lectures on Machine Learning* (Vol. 3176, pp. 63-71). https://doi.org/10.1007/978-3-540-28650-9_4

Rastin, Z., Ghodrati Amiri, G., & Darvishan, E. (2021). Unsupervised structural damage detection technique based on a deep convolutional autoencoder. *Shock and Vibration*, *2021*(1), 6658575.

Römgens, N., Abbassi, A., Jonscher, C., Grießmann, T., & Rolfes, R. (2024). On using autoencoders with non-standardized time series data for damage localization. *Engineering Structures*, *303*, 117570.

Saito, K., Yamamoto, S., Ushiku, Y., & Harada, T. (2018). Open set domain adaptation by backpropagation. Proceedings of the European Conference on Computer Vision (ECCV),

Salkhordeh, M., Mirtaheri, M., & Soroushian, S. (2021). A decision-tree-based algorithm for identifying the extent of structural damage in braced-frame buildings. *Structural Control and Health Monitoring*, *28*(11), e2825.

Santos, A., Figueiredo, E., Silva, M., Santos, R., Sales, C., & Costa, J. (2017). Genetic-based EM algorithm to improve the robustness of Gaussian mixture models for damage detection in bridges. *Structural Control & Health Monitoring*, *24*(3), Article e1886. https://doi.org/10.1002/stc.1886

Sarmadi, H., Entezami, A., & Ghalehnovi, M. (2022). On model-based damage detection by an enhanced sensitivity function of modal flexibility and LSMR-Tikhonov method under incomplete noisy modal data. *Engineering with Computers*, *38*(1), 111-127. https://doi.org/10.1007/s00366-020-01041-8

Sen, D., & Nagarajaiah, S. (2018). Data-driven approach to structural health monitoring using statistical learning algorithms. In *Mechatronics for cultural heritage and civil engineering* (pp. 295-305). Springer.

Shadan, F., Khoshnoudian, F., & Esfandiari, A. (2016). A frequency response-based structural damage identification using model updating method. *Structural Control & Health Monitoring*, *23*(2), 286-302. https://doi.org/10.1002/stc.1768

Shin, H. C., Roth, H. R., Gao, M. C., Lu, L., Xu, Z. Y., Nogues, I., Yao, J. H., Mollura, D., & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *Ieee Transactions on Medical Imaging*, *35*(5), 1285-1298. https://doi.org/10.1109/tmi.2016.2528162

Shokravi, H., Shokravi, H., Bakhary, N., Heidarrezaei, M., Rahimian Koloor, S. S., & Petrů, M. (2020). Vehicle-assisted techniques for health monitoring of bridges. *Sensors*, *20*(12), 3460.

Silik, A., Noori, M., Altabey, W. A., Ghiasi, R., & Wu, Z. (2021). Comparative analysis of wavelet transform for time-frequency analysis and transient localization in structural health monitoring. *Structural Durability & Health Monitoring*, *15*(1), 1.

Silva, M., Santos, A., Figueiredo, E., Santos, R., Sales, C., & Costa, J. (2016). A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges. *Engineering Applications of Artificial Intelligence*, *52*, 168-180. https://doi.org/10.1016/j.engappai.2016.03.002

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sohn, H., Kim, S. D., & Harries, K. (2008). Reference-free damage classification based on cluster analysis. *Computer-Aided Civil and Infrastructure Engineering*, *23*(5), 324-338. https://doi.org/10.1111/j.1467-8667.2008.00541.x

Sohn, H., Worden, K., & Farrar, C. R. (2002). Statistical damage classification under changing environmental and operational conditions. *Journal of Intelligent Material Systems and Structures*, *13*(9), 561-574. https://doi.org/10.1106/104538902030904

Sun, L., Shang, Z., Xia, Y., Bhowmick, S., & Nagarajaiah, S. (2020). Review of Bridge Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection. *Journal of Structural Engineering*, *146*(5), Article 04020073. https://doi.org/10.1061/(asce)st.1943-541x.0002535

Taha, M. R., Noureldin, A., Lucero, J., & Baca, T. (2006). Wavelet transform for structural health monitoring: a compendium of uses and features. *Structural Health Monitoring*, *5*(3), 267-295.

Talaei, S., Beitollahi, A., Moshirabadi, S., & Fallahian, M. (2018). Vibration-based Structural Damage Detection Using Twin Gaussian Process (TGP). *Structures*, *16*, 10-19. https://doi.org/10.1016/j.istruc.2018.08.006

Tan, C. Q., Sun, F. C., Kong, T., Zhang, W. C., Yang, C., & Liu, C. F. (2018, Oct 04-07). A Survey on Deep Transfer Learning.*Lecture Notes in Computer Science* [Artificial neural networks and machine learning - icann 2018, pt iii]. 27th International Conference on Artificial Neural Networks (ICANN), Rhodes, GREECE.

Tang, Q., Xin, J., Jiang, Y., Zhou, J., Li, S., & Chen, Z. (2022). Novel identification technique of moving loads using the random response power spectral density and deep transfer learning. *Measurement*, *195*, 111120.

Tsialiamanis, G., Mylonas, C., Chatzi, E., Dervilis, N., Wagg, D. J., & Worden, K. (2021). Foundations of population-based SHM, Part IV: The geometry of spaces of structures and their feature spaces. *Mechanical Systems and Signal Processing*, *157*, Article 107692. https://doi.org/10.1016/j.ymssp.2021.107692

Turner, J. D., & Pretlove, A. J. (1991). *Acoustics for engineers*. Macmillan.

Vitola, J., Pozo, F., Tibaduiza, D. A., & Anaya, M. (2017). A Sensor Data Fusion System Based on k-Nearest Neighbor Pattern Classification for Structural Health Monitoring Applications. *Sensors*, *17*(2), Article 417. https://doi.org/10.3390/s17020417

Wahab, M. A., De Roeck, G., & Peeters, B. (1999). Parameterization of damage in reinforced concrete structures using model updating. *Journal of Sound and Vibration*, *228*(4), 717-730.

Wang, W. T., Bao, Y. Q., Zhou, W. S., & Li, H. (2018). Sparse representation for Lamb-wave-based damage detection using a dictionary algorithm. *Ultrasonics*, *87*, 48-58. https://doi.org/10.1016/j.ultras.2018.02.011

Wei, C. W., Huang, J. F., Mansaray, L. R., Li, Z. H., Liu, W. W., & Han, J. H. (2017). Estimation and Mapping of Winter Oilseed Rape LAI from High Spatial Resolution Satellite Data Based on a Hybrid Method. *Remote Sensing*, *9*(5), Article 488. https://doi.org/10.3390/rs9050488

Weinberger, K. Q., & Saul, L. K. (2009). Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, *10*, 207-244. <Go to ISI>://WOS:000270824200005

Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, *1*(1), 1-34.

Wolpert, D. H. (2001, Sep). The supervised learning no-free-lunch theorems. [Soft computing and industry: Recent applications]. 6th On-Line World Conference on Soft Computing in Industrial Applications (WSC6), Electr Network.

Worden, K., & Manson, G. (2003). Experimental validation of a structural health monitoring methodology: Part I. Novelty detection on a laboratory structure. *Journal of Sound and Vibration*, *259*(2), 323-343. https://doi.org/10.1006/jsvi.2002.5168

Worden, K., Manson, G., & Fieller, N. R. J. (2000). Damage detection using outlier analysis. *Journal of Sound and Vibration*, *229*(3), 647-667. https://doi.org/10.1006/jsvi.1999.2514

Xiao, F., Chen, G. S., Zatar, W., & Hulsey, J. L. (2021). Signature extraction from the dynamic responses of a bridge subjected to a moving vehicle using complete ensemble empirical mode decomposition. *Journal of Low Frequency Noise, Vibration and Active Control*, *40*(1), 278-294.

Xu, H., Chen, X., Chen, J., Shi, L., Yang, D., Wang, Z., & Yang, Y. (2024). Review of vehicle scanning method for bridges from 2004 to 2024. *International Journal of Structural Stability and Dynamics*, 2530003.

Yan, Y. J., Cheng, L., Wu, Z. Y., & Yam, L. H. (2007). Development in vibration-based structural damage detection technique. *Mechanical Systems and Signal Processing*, *21*(5), 2198-2211. https://doi.org/10.1016/j.ymssp.2006.10.002

Yessoufou, F., & Zhu, J. (2023). Classification and regression-based convolutional neural network and long short-term memory configuration for bridge damage identification using long-term monitoring vibration data. *Structural Health Monitoring*, *22*(6), 4027-4054.

Yeung, W. T., & Smith, J. W. (2005). Damage detection in bridges using neural networks for pattern recognition of vibration signatures. *Engineering Structures*, *27*(5), 685-698. https://doi.org/10.1016/j.engstruct.2004.12.006

You, K., Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2019). Universal domain adaptation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Yu, Y., Wang, C. Y., Gu, X. Y., & Li, J. C. (2019). A novel deep learning-based method for damage identification of smart building structures. *Structural Health Monitoring-an International Journal*, *18*(1), 143-163. https://doi.org/10.1177/1475921718804132

Zhan, J., Zhang, F., Siahkouhi, M., Kong, X., & Xia, H. (2021). A damage identification method for connections of adjacent box-beam bridges using vehicle-bridge interaction analysis and model updating. *Engineering Structures*, *228*, 111551.

Zhang, W., Li, J., Hao, H., & Ma, H. (2017). Damage detection in bridge structures under moving loads with phase trajectory change of multi-type vibration measurements. *Mechanical Systems and Signal Processing*, *87*, 410-425.

Zhang, Y., Miyamori, Y., Mikami, S., & Saito, T. (2019). Vibration-based structural state identification by a 1-dimensional convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(9), 822-839.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., & Loy, C. C. (2022). Domain generalization: A survey. *Ieee Transactions on Pattern Analysis and Machine Intelligence*.

Zhou, Q. F., Ning, Y. P., Zhou, Q. Q., Luo, L. K., & Lei, J. Y. (2013). Structural damage detection method based on random forests and data fusion. *Structural Health Monitoring-an International Journal*, *12*(1), 48-58. https://doi.org/10.1177/1475921712464572

Zhu, X. Q., & Law, S. S. (2006). Damage Detection in Simply Supported Concrete Bridge Structure Under Moving Vehicular Loads. *Journal of Vibration and Acoustics*, *129*(1), 58-65. https://doi.org/10.1115/1.2202150

Zhu, X. Q., & Law, S. S. (2015). Structural health monitoring based on vehicle-bridge interaction: accomplishments and challenges. *Advances in Structural Engineering*, *18*(12), 1999-2015.

Zhuang, F. Z., Qi, Z. Y., Duan, K. Y., Xi, D. B., Zhu, Y. C., Zhu, H. S., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the Ieee*, *109*(1), 43-76. https://doi.org/10.1109/jproc.2020.3004555

Zolfaghari, M., Jourabloo, A., Gozlou, S. G., Pedrood, B., & Manzuri-Shalmani, M. T. (2014). 3D human pose estimation from image using couple sparse coding. *Machine Vision and Applications*, *25*(6), 1489-1499. https://doi.org/10.1007/s00138-014-0613-6