THE UNIVERSITY *of* EDINBURGH

Big Data Analysis on Long-span Bridge Structural Health Monitoring Systems

Donghui Xu

Doctor of Philosophy

The University of Edinburgh

2023

Abstract

Structural Health Monitoring (SHM) systems are being installed on long-span bridges worldwide to facilitate condition assessment and damage detection. The vast volume of data generated by these SHM systems presents significant challenges for analysis. This thesis introduces innovative research into big data analysis, drawing on SHM data from two long-span bridges in Scotland: the Queensferry Crossing (QC) and the Forth Road Bridge (FRB).

Initially, data integrity was addressed, as raw data from both bridges exhibited errors. The errors were detected by visual analysis and statistical analysis, e.g. z-score. The data available for this project related to the first year of collection from the QC - before the final snagging of the system was undertaken by the contractor. Ensuring data integrity is crucial for subsequent analysis. A novel Random Forest model based on relavant sensor data was proposed to impute substantial segments of missing data. The z-score filter and Exponentially Weighted Moving Average (EWMA) were utilised to identify and rectify outliers. The proposed methods were found to effectively reconstruct SHM data, interpolating missing segments and identifying and substituting outliers.

Subsequently, to investigate the vehicle-induced structural response, a new traffic load simulation model was developed. This simulation utilised a car-following model combined with a lane-changing model to replicate the trajectory of vehicles on the Queensferry Crossing. This was based on vehicle attribute distributions captured by the dynamic weigh-in-motion system positioned at the bridge entrance. The deflection influence line at the south midspan was then derived from the bridge's finite element

model. Ultimately, the deflection at the south midspan was determined using both the simulated vehicle data and the influence line. The findings revealed a strong correlation between the actual monitored and simulated deflections in terms of both magnitude and variations.

Once the simulation model was validated, a recurrent neural networks training, validation, and testing strategy was proposed to predict deflections based on the vehicle information at the bridge entrance using the simulated data. Parameter analysis was performed to investigate the impact of hyper-parameters, e.g. neuron numbers and number of layers, on the prediction performance. It was discovered that multi-layer Long-short Term Memory (LSTM) networks outperform single-layer models with a large number of neurons. The performance of LSTM was evaluated by the mean square root between the observed values and predicted values. The results showed accurate generalisation of deflection data based on traffic information.

The study also delved into the exploration of thermal-induced deflections. The initial step involved identifying the QC's critical sections using its finite element model. The analysis revealed that significant bending moments were observed at the north and south midspans, as well as the deck over the southern and northern auxiliary piers. Considering the placement of SHM sensors, we chose south midspan as the research target. Then, a temperature field analysis was performed to investigate temperature differences at different locations on the bridge. Next, Generalised Pareto Distrbution (GPD) was employed to estimate the maximum vertical temperature difference and tower temperature difference in 120 years based on SHM data.

Further, finite element analysis was performed to compare south midspan deflections when thermal loads were applied to different parts of the QC. It was observed that thermal loads on the stay cables have the most substantial impact on deck deformation. Then, correlations between temperatures and deflections were explored by using wavelet transform. The study culminated with the application of LSTM to predict deflections based on temperature data sourced from various bridge locations.

This research explores the feasibility of machine learnring applications to bridge SHM data analysis, mapping complex relationships between environmental & operational conditions and bridge structural responses. The outcome of this research provides researchers and bridge engineers with new perspectives on how to utilise SHM data for bridge maintenance.

Acknowledgements

I would like to extend my deepest gratitude to Professor Michael C. Forde for his unwavering support and guidance throughout the duration of this project as my principal supervisor.

I would like to say thank you to Professor Yong Lu for his valuable suggestions in the annual review and support as the second supervisor.

I am profoundly grateful for the invaluable support provided by Transport Scotland, James Fisher Testing Services, and Bear Scotland, who have generously supplied essential data and documentation for this research.

Special thanks are due to Dr Antonio Caballero from Screening Eagle Technologies for the financial support and the wealth of ideas generated during our numerous insightful discussions.

I would also like to express my appreciation to Dr Xiang Xu for his inspirational suggestions that have greatly enriched this work.

This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF) (http://www.ecdf.ed.ac.uk/).

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Donghui Xu

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Backgrounds

The term "span" refers to the length of a bridge deck between two ground supports. "Long-span" is a relative term that has evolved alongside advancements in materials and bridge construction technology, leading to a significant increase in bridge span lengths throughout the twentieth century. Long-span bridges have played an indispensable role in human society, facilitating crucial connections across rivers, valleys, bays, and other natural formations. De Miranda (2016) delineated two typical types of long-span bridges, namely suspension bridges and cable-stayed bridges.

The construction of long-span bridges experienced a significant surge in the United States during the 1930s as a measure to counteract the Great Depression (Administration, 1939). Subsequently, World War II impacted construction efforts, which were later revitalised in both North America and Europe following the war's conclusion. A second wave of long-span bridge construction emerged in Asia, driven by economic prosperity; this began in Japan in the 1970s and in China in the 2000s. Meanwhile, North America experienced a decline in construction since its peak in the 1960s.

The design life varies in different standards. For example, the American Association of State Highway and Transportation Officials Load Resistance Factor Design (AASHTO LRFD) 2012 specifies the minimum design life for a permanent bridge is 75 years in the United States and North America. The Eurocode (EN 1990.2002) requires a design life span of 120 years for bridges. Therefore, it is evident that post-war bridges constructed in North America will approach their minimum design life span within the next two decades.

The American Society of Civil Engineers publishes an Infrastructure Report Card annually to assess the condition and performance of all American infrastructure. In the 2021 report, it was revealed that 42% of all bridges are at least 50 years old, and 7.5% of the nation's bridges are deemed structurally deficient, carrying over 178 million trips each year. A similar situation exists in Europe as well. For instance, in the UK, the Forth Road Bridge, which spans the Firth of Forth and connects Edinburgh in the south to Fife in the north, was discovered to have truss end link failures. Consequently, the Forth Estuary Transport Authority (FETA) recommended its full replacement in December 2015. At present, the Forth Road Bridge only accommodates public transport, pedestrians, and cyclists. All other traffic utilises the replacement bridge, the Queensferry Crossing, which incurred a construction cost of approximately 1.35 billion pounds sterling.

Hence, regular inspections and assessments are vital to avert premature structural deficiencies in these long-span bridges. Traditionally, assessments of in-service bridges have relied on periodic visual inspections. However, it has been found that these assessment methods can be inaccurate due to the subjectivity of inspectors and the inaccessibility of critical components (Phares, Graybeal, Rolander, Moore, & Washer, 2001). Moreover, these visual inspections can prove to be both expensive and time-

consuming. For instance, the biennial visual inspection of the Brooklyn Bridge in New York incurred a cost of $1 million and required 3 months for completion (Dubin & Yanev, 2001). As a result, there is a pressing need for a continuous and automated monitoring technique for bridges.

Motivated by the limitations of traditional visual inspection methods and the advancements in statistical analysis techniques, research and implementation of structural health monitoring (SHM) systems have gained popularity in recent years. SHM can be defined as a comprehensive system encompassing sensory, data manipulation, damage detection, and modelling components to continuously acquire information about the integrity of in-service structures (Baidar & Aftab, 2015).

For a continuous monitoring system such as SHM, vast amounts of data are generated daily. To fully understand the potential of these data, there are generally two methods that can be implemented. The model-based methods require a finite element model of the actual structure to represent the structural responses of real bridges. The advantages of model-based methods include the potential accuracy of damage detection, including their locations and severity (Neves, González, Leander, & Karoumi, 2017). However, developing these finite element models (FEM) can be challenging, and the computations associated with these models can be quite costly.

Data-based methods allow damage detection without a digital replica/twin of real bridges. Statistical models are developed purely from the SHM data, and patterns and trends can be identified through statistical analysis. However, due to the complexity of the operational environment of in-service bridges, interpreting the monitoring data can pose a challenge. The large volume of data often contains errors, including missing parts, drifted sensors, and incorrect units, among others. Therefore, data preprocessing is essential before any analysis. On the other hand, statistical model results are not as intuitive as the FEM results.

1.2 Thesis Aims & Objectives

Given the concerns surrounding premature structural deficiencies in long-span bridges and the rapid implementation of complex SHM systems, research on SHM data has garnered increased interest in recent years. The aims of this thesis are to gain an understanding of raw long-term SHM data on long span bridges and to investigate the environmental and operational impacts on bridge responses. The following objectives are proposed to achieve the aims:

- Inspect raw SHM data from long span bridges. Propose new preprocessing techniques to enhance the overall data quality, thereby rendering it suitable for subsequent analysis.
- Develop a novel empirical model capable of generating stochastic traffic data on the bridge. Compare the simulated structural response with monitored data. The proposed simulation methodology offers the advantage of enabling traffic analysis even when monitored data are incomplete or unreliable.
- Examine the impact of traffic on the bridge deck through the application of machine learning techniques to predict deck deflections at various locations of interest.
- Conduct a comprehensive temperature field analysis on different components of the bridge. This analysis provides insights into thermal load distributions on different parts of the bridge.
- Perform extreme temperature estimations based on one-year monitoring data for several bridge components and compare the results with design codes.
- Separate dynamic load effects from the monitored structural responses using a wavelet transform filter. Use a neural network model to perform the thermal-induced structural response prediction.

Chapter 2

Review on Structural Health Monitoring & Machine Learning

This chapter introduces the historical development of the Structural Health Monitoring system and the challenges to the SHM study. Furthermore, insights of two approaches, model-based and data-based methods, to the SHM research are also provided. Also, a brief introduction of machine learning and its applications is also presented.

2.1 Premauture failures on bridges

Bridge safety and reliability are the common interests of engineers across the world. The failure of bridge components can lead to serious consequences including the collapse of the bridge, which not only can cause casualties but also compromise public confidence in civil infrastructures in the long term. Mahmoud (2009) provides a detailed review of bridge failure case studies. Most bridge failures happen during the construction stage as the structure is incomplete, therefore, vulnerable. In-service bridge collapse is rare. Several common causes of bridge failure nowadays are listed below:

- **Unforeseen natural disasters** For example, the partial collapse of the San Francisco-Oakland Bay Bridge in Oakland, California due to the Loma Prieta earthquake in 1989. Therefore, in active seismic areas, e.g. Japan, bridges are constantly monitored and assessed mainly for verifying the functions of vibration control systems (Fujino, Siringoringo, Ikeda, Nagayama, & Mizutani, 2019).
- **Accidents caused by human errors** The most common cause here is the collision of ships with bridges. For example, the Almö Bridge in Sweden collapsed as a result of a strike to the arch from a bulk carrier. There are cases that the bridge collapsed due to insufficient design. For example, the I-35W Mississippi River bridge in Minnesota, US, suddenly collapsed in 2007 due to undersized gusset plates.
- **Premature failures** This can be interpreted as structural failures that affect the operation of the bridge before the design lifespan of certain components. For example, shear failure of bearings in expansion joints was discovered after only 3-4 years of service on the Jiangyin Bridge in China (T. Guo, Liu, Zhang, & Pan, 2015). Usually this is due to the following causes: lack of inspections and maintenance during the operation stage, underestimated traffic loads during the design stage, insufficient design due to limited knowledge back in time.

Unlike damages caused by accidents, premature failures can only be detected by regular inspections, which makes them hard to discover. Among all the different bridge components in long-span bridges, the expansion joints are one of the most vulnerable parts due to the direct and repetitive influence of vehicle loads (T. Guo et al., 2015). On the other hand, the average design service life span of the joints is between 15 to 20 years, significantly shorter than the life span of the bridge (Dexter, 2002). Therefore, most premature failures happen on the expansion joints. For example, the main expansion joints, the roller shutter joints, on the Forth Road Bridge have been in service for over 50 years since opening. The traffic volume and weight on the bridge

have increased dramatically since then, which is beyond the scope of the design stage. The annual traffic volume was 4 million when the bridge was first opened and then increased to 24 million in 2008. Furthermore, the maximum weight of heavy good vehicles permitted on UK roads increased from 22 tons in 1964 to 44 tonnes today (Colford, Jones, Timby, & Brown, 2009). Therefore, as a result, these joints were considered at the end of their service life in 2009. However, the work required to replace these joints would significantly affect the daily traffic and the high cost of the related temporary work made it difficult for such a maintenance to occur. On the other hand, a new replacing bridge across the Forth, known as the Queensferry Crossing nowadays, was opened in 2017. Therefore, the service life of these joints was extended until the opening of the new bridge after a review with frequent monitoring and maintenance. Premature failures of the expansion joints are also reported on other long-span bridges.

Another bridge component suffering from premature failures is the bridge deck. As an important superstructure, the quality of the bridge deck has a direct influence on the driving quality during the bridge operation stage (L. Deng, Wang, & Cai, 2017). The deterioration of the reinforced concrete deck is a major concern to bridge engineers. Martino, Maser, Birken, and Wang (2016) summarised three most common causes: freeze-thaw action, alkali-silica reaction and reinforcement corrosion, the last being the dominant cause. Due to the nature of the reinforcement, these corrosions are difficult to be identified by visual inspections. As a result, a bridge deck may need to be replaced many times throughout the life span of the bridge, increasing cost and affecting the bridge operation.

Apart from bridge degradations, other factors also affect the operations of bridges, such as ice accretion [1] on cables, endangering the public (Johnson, 2023).

2.2 Structural Health Monitoring Development

The term "damage" can be defined as changes resulting in negative impacts on the performance of a certain system. In the engineering field, damages are often changes in the material or geometries of the subject. Some of these changes can potentially lead to the total loss of functions (C. R. Farrar & Worden, 2007). Therefore, engineers carry out regular inspections of these systems to identify damages. Some of these inspections require the system to be put out of service for an extended period of time, which causes an economic loss to the system owner. Also, these inspections are normally scheduled periodically. Therefore, they may not be able to identify the damages in a timely manner before they escalate to a critical level to the system. As a result of the above concerns, a continuous monitoring method for these systems is always in high demand. Structural Health Monitoring (SHM) has been widely implemented as a continuous damage identification strategy in aeronautical, civil and mechanical engineering fields (Catbas et al., 2017).

One of the most successful applications of the SHM system is in aeronautics. The industry realised the necessity of monitoring aeroplanes after several fatal crashes including the Aloha Airlines flight 243 accident on 28 April 1988. In this accident, the aeroplane suffered extensive damage as a result of multiple-site fatigue cracking (National Transportation Safety Board, 1989). Several types of sensors are frequently implemented in aeronautics, including strain gauges, optical fibre sensors and crack wires, etc. Also, some non-destructive testing (NDT) techniques are used to assess the

[1] Traffic Diversion Strategy, New Civil Engineer, 3 July 2023 https://www.newcivilengineer.com/latest/traffic-diversion-only-remedy-to-queensferry-crossings-falling-ice-woes-after-many-solutions-tested-03-07-2023/

structural integrity, for example, acoustic emission, an NDT method using acoustic waves to "listen" to cracks or delamination (Boller & Meyendorf, 2008). These sensors and NDT methods are widely adopted in SHM systems in other fields as well. For example, SHM systems are also widely installed on rotating machinery, including wind turbines, turbogenerator sets and compressors, etc. The damage identification process is relatively simple on rotating machinery as the damage locations are generally known. Therefore, using a single-channel Fast Fourier Transform is sufficient for periodical monitoring of these machines (C. R. Farrar & Worden, 2007). The vibration signature is an essential feature to determine the damage status in rotating machinery as structural damages will normally lead to pronounced changes of the dynamic behaviour. As a result, damages can be assessed by measuring and comparing the different structural vibrations (Catbas et al., 2017).

In the civil engineering field, structural health monitoring systems are gradually implemented on bridges (Cousins, McAra, & Hill, 2022; D. Xu et al., 2019), high-rise buildings (Rafiei & Adeli, 2017), dams (Oliveira & Alegre, 2019), tunnels (Manuello, Niccolini, & Carpinteri, 2019) and wind turbines (Siringoringo & Fujino, 2006) etc. It can be summarised that the Structural health monitoring system used in the civil engineering field is for the following purposes, confirming design, construction aid, demonstration and life extension (Inaudi, 2010). Furthermore, there is a tendency for more SHM systems on bridges to be designed for life extension purposes, allowing continuous monitoring during the operation stage. Table 2.1 shows several examples of SHM system installation on long-span road bridges worldwide. It can be concluded that a comprehensive SHM system consists of 3 types of sensors:

- **Environment monitoring** Towers, cables and deck temperatures are recorded. Wind speed and air pressure, humidity are also included in the monitoring dataset. Thermal and wind-induced structural responses can be investigated using the environmental data (Rizzo & Enshaeian, 2021; X. Xu, Forde, Ren, & Huang, 2021). Furthermore, temperatures and air pressure can be used to determine cable icing conditions (Cousins et al., 2022). The sampling rate for these sensors is fixed.
- **Traffic monitoring** Traffic crossings are normally monitored by a dynamic weigh-in-motion (DWIM) system. Different from environmental or structural response sensors, DWIM is only triggered when a vehicle passes its sensors. Each data entry of DWIM generally contains the vehicle passing time, number of axes, speed, weight and the system normally assigns a class for each passing vehicle based on its number of axes and weight (e.g. cars, motorcycles or lorries). With the presence of DWIM data, the traffic-induced structural responses can be further studied.
- **Structure monitoring** The structure monitoring system consists of a variety of sensors, including dynamic strain gauges, accelerometers, displacement transducers, cable force sensors, corrosion sensors and inclinometers etc (Sakiyama, Lehmann, & Garrecht, 2021). These sensors are mounted on different structural components. The bridge management team can use these sensors to assess the health conditions of different bridge parts.

Figure 2.1 shows a few examples of sensors installed on the Queensferry Crossing, UK. The Queensferry Crossing is currently monitored by 2184 sensors (Cousins et al., 2022). Apart from bridge degradations, other factors also affect the operations of bridges, such as ice accretion on cables, endangering the public. Therefore, a

continuous monitoring system installed on long-span bridges is needed to identify these hazards. Apart from the sensory system, an SHM system installed on long-span bridges normally also includes local or cloud storage facilities and a real-time assessment process, alerting the bridge management team if anomalies are observed.

With the rapid development in SHM implementations, a large amount of monitoring data is available for analysis and relevant research has become more popular. There are two methods that can be implemented for the research of SHM systems, model-based and data-based. Extensive research has been carried out based on these two methods for extreme load estimation (Getachew & Obrien, 2007), condition assessment (M. Liu, Frangopol, & Kim, 2009) and damage detections (S.-Z. Chen, Wu, & Feng, 2019). Detailed reviews and critiques will be provided in the following sections.

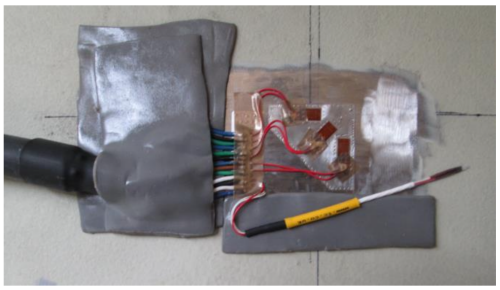
Table 2.1: Examples of SHM implementations on long-span bridges

Bridge	Bridge Type	Total length (m)	SHM installation time	Number of sensors	Sensors	Reference
Golden State Bridge California, USA	Suspension	1966	1993	64	Accelerometers	Çelebi (2012)
Tsing-ma Bridge Hong Kong	Suspension	2160	1997	280	Environmental (wind, temperature) Response (GPS, acceleration, strain)	S. Li, Zhu, Xu, Chen, and Li (2012)
Tamar Suspension Bridge Plymouth, UK	Suspension	642	2001	Around 100	Environmental (wind, temperature) Response (cable tension, deck level)	Koo, Brownjohn, List, and Cole (2013)
Humber Bridge Yorkshire, UK	Suspension	2220	2010-2012	12	Environmental (wind, temperature) Response (GPS, Accelerations)	J. Brownjohn, Koo, Scullion, and List (2015)
Sutong Bridge Jiangsu, China	Cable-stayed	8206	2008	1078	Environmental (wind, temperature) Response (GPS, Accelerations, cable tension, deck level, etc.)	H. Wang, Li, Niu, Zong, and Li (2013)
Forth Road Bridge Edinburgh, UK	Suspension	2517	2015	192	Environmental (wind, temperature) Response (GPS, Accelerations, strain, etc.) Operational (Weigh-in-motion)	D. Xu et al. (2019)
Queensferry Crossing Edinburgh, UK	Cable-stayed	2700	2018	2184	Environmental (wind, temperature) Response (GPS, Accelerations, strain, etc.) Operational (Weigh-in-motion)	Cousins et al. (2022)

(a) Rainfall Senesor(height: 39cm, diameter: 18.5cm)

(b) Deck Thermometer

(c) Displacement Transducer ($6cm \times 4cm$)



(d) Dynamic Strain Gauge ($80cm \times 60cm \times 45cm$ with enclosure)

Figure 2.1: Sensors on Queensferry Crossing

2.3 Model-based Methods

The model-based methods normally first require a finite element (FE) model of the real structure. Then, monitoring data will be used as data input for the FE models. Then, the damaged-induced perturbations in mass or stiffness can be identified (Moore, Nichols, & Murphy, 2012). Görl and Link (2003) compared the changes in stiffness parameters of undamaged and damaged structures using FE models to identify the extent of the damage from an experimental steel frame structure. Moore et al. (2012), on the other hand, successfully identified specific cracks using only three arbitrarily

located strain gauges on a thin, clamped aluminium plate. However, these works only focused on small-scale experimental specimens which are much less complex than in-service structures in load conditions. Recently, a new method called finite element model updating has emerged in the SHM research area (Friswell & Mottershead, 2013). Using the FE model updating method, the deviations in structural parameters, e.g. structural responses and material properties, between an FE model baseline and the updated FE model can be obtained. Then, the locations and severity of damage can be identified by these deviations (Wu & Li, 2006). Reynders, Teughels, and De Roeck (2010) used this general method to identify the loss of stiffness in an in-service bridge. J. M. W. Brownjohn, Moyo, Omenzetter, and Lu (2003) identified the natural frequency increases using FE model updating after some upgrading works on a short-span bridge. These investigations have focused on the dynamic structural responses for their FE model updating, e.g. natural frequencies or mode shapes. Recently, some researchers have been using static strain results from the SHM system to update FE models. Okasha, Frangopol, and Orcesi (2012) used monitored strain data obtained in a crawl test to assess the lifetime reliability of ageing bridges by FE model updating. Sanayei, Phelps, Sipple, Bell, and Brenner (2011) presented the method to use the strain data collected during the load test of a newly constructed bridge to calibrate a baseline FE model for future use in the SHM system.

It can be concluded that most FE model-based methods perform damage detections by identifying the stiffness changes using frequencies and mode shapes. However, high sampling rate frequency data is hardly available for long-term monitoring as data storage is limited. On the other hand, the development of an FE model for long-span bridges can be difficult and the computation cost is much higher than experimental structures. Therefore, data-based methods are gaining popularity in recent years for damage detection and condition assessment on in-service long-span bridges.

2.4 Machine Learning

Machine learning (ML) can be defined as computer algorithms enabling programs to improve with experience automatically using data (T. Mitchell, 1997). ML can be classified as a subset of artificial intelligence (AI). AI is the capability of a computer system to simulate human behaviours like learning and sensing. ML has achieved great success in multiple fields, e.g. computer vision (Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018), natural language processing (Olsson, 2009) and data mining (Dogan & Birant, 2021; T. M. Mitchell, 1999). Machine learning application in the civil engineering field is fairly recent. For instance, C. Farrar and Worden (2012) introduced the theoretical background of different machine learning algorithms and their potential applications in SHM data analysis but lacks case studies in large civil infrastructures.

Machine learning is a type of Artificial intelligence (AI) where models are trained with sample data (input) to make predictions or decisions without human intervention during the training stage. It can be concluded that there are 3 parts in the procedure, input data, ML model training and output data. Depending on different problems, the selection of the machine learning model and the input data preparation can vary.

2.4.1 Input data

Data manipulation

In this research, data manipulation is defined as the process of visualising and reshaping raw data to make it more organised, understandable, or efficient for subsequent data cleaning processes. For example, if the raw data are stored in a less efficient format, such as a string, data manipulation will convert it to a float to further save storage space and increase the subsequent processing speed.

Data cleaning

SHM systems collect a great number of different types of data, e.g. time-series data (temperature, strain, deflection), discrete data (vehicle passing from the weigh-in-motion system) and sometimes imagery data (concrete surface photos). However, the dimension of the input data to an algorithm must maintain consistency. For example, the discrete data need to be converted to time-series format before being concatenated with the time-series dataset. Furthermore, sensors collect data at different sampling rates even under the same SHM system on one bridge. Therefore, the raw data cannot be used as input to machine learning algorithms. For example, in the same input dataset indexed by timestamp, the sampling rate needs to remain consistent for different data types. Downsampling and upsampling techniques may be necessary. Overall, high-frequency data requires more computational resources during training. Furthermore, noises in high-frequency data can impact the accuracy of ML models. Data manipulation that reshapes the dimension of the raw data is essential.

After manipulating the data, the next step is to ensure the data quality which is essential to any machine learning task. Poor quality data can lead to catastrophic consequences and even cause projects to fail (Gudivada, Apon, & Ding, 2017). In the SHM data, several common types of data errors are listed below.

- **Missing data** This is presented as gaps in a continuous dataset. Normally, missing data is caused by malfunctioning sensors caused by power issues or errors when certain data points are recorded in the storage system. In the raw data, missing parts are shown as empty entries or *Nan*, which the algorithms will treat as errors and leads to failed training. Therefore, these missing parts must be imputed.

- **Outliers** Outliers are data that significantly differ from other records. They often indicate a measurement error by sensors affected by environmental factors. These outliers usually exceed the reasonable range of its data type, e.g. $100^{\circ}C$ for air temperature. These data must be excluded from the dataset as it will greatly reduce the accuracy of the algorithms since the variation of the data will be flattened with the existence of extreme values. On the other hand, some outliers are within the reasonable range, indicating abnormal events. These outliers should be preserved. Separate classification algorithms should be used to distinguish outliers from the raw data. For example, the k-nearest neighbours classifier (k-NN) can be used to identify outliers based on distance (Schubert, Zimek, & Kriegel, 2014). AutoRegressive Integrated Moving Average (ARIMA) model (Bianco, Garcia Ben, Martinez, & Yohai, 2001) and Isolation Forest (Cheng, Zou, & Dong, 2019) are also observed to be used to detect outliers for time series data.

Data normalisation

Normalisation is necessary to further improve the data quality after the data cleaning process. Different sensors in an SHM system have different measurement scales, e.g. strain, displacement and temperature. These measurements show significantly different data ranges from each other. Data normalisation is a process to adjust each measurement to the same scale so each feature will have the same numerical contribution to the algorithms (Singh & Singh, 2020). One of the most commonly used normalisation methods is the min-max normalisation, where the data are rescaled to $[0, 1]$. The general formula can be written as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.1}$$

Other data normalisation methods are also used, such as z-score normalisation where the normalised data has a mean of 0 and a standard deviation of 1, shown in Equation 2.2; MaxAbsScaler which scales and translates each feature individually such that the maximal absolute value of each feature in the training set will be 1, shown in Equation 2.3.

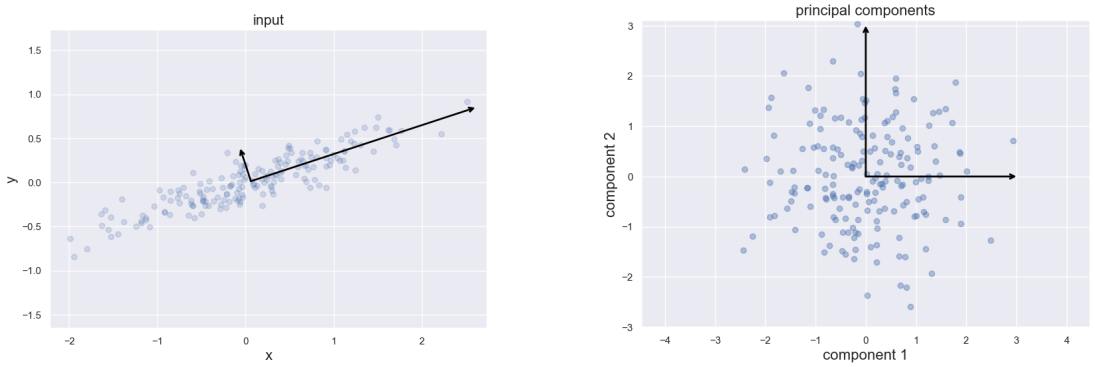
$$X' = \frac{X - \bar{X}}{\sigma} \tag{2.2}$$

where X' is the scaled value, X is the original value, $\bar{X}$ is mean of the dataset and σ is the standard deviation.

$$X' = \frac{X}{\max(|X|)} \tag{2.3}$$

Feature engineering

Finally, a step called feature engineering is applied to the data. Feature engineering can be described as a process to extract features, e.g. characteristics, from the raw data. It will either add extra features based on the raw measurements, for instance, one-hot encoding for categorical data, an example of which is depicted in Table 2.2 with four classes utilising one-hot encoding. One-hot encoding can convert categorical types into numerical values without introducing bias. In One Hot Encoding, the categorical parameters will prepare separate columns for categories and the value will be set to 1 for the corresponding category while other columns are all zeros. Another example is to reduce the dimension of the raw dataset by implementing Principal Component Analysis (PCA), an example of which is illustrated in Figure 2.2a and 2.2b. Dong and Liu (2018) described a variety of feature engineering approaches for different data types. In general, feature engineering strategies depend highly on the data and algorithms. There are no universal procedures. Efficient feature engineering will contribute significantly to the generalisation of the machine learning models but it's also labour-intensive (Bengio, Courville, & Vincent, 2013).



(a) Original data **(b)** Principal components

Figure 2.2: PCA example

In this research, wavelet transform was used to separate high frequencies caused by dynamic loads when investigating temperature-induced deck deflections. Furthermore, discrete traffic data were converted to the time-series format when studying vehicle-induced deck deflections.

Table 2.2: One-hot encoding

Undamaged	1	0	0	0
Minor	0	1	0	0
Major	0	0	1	0
Critical	0	0	0	1

In general, the data preprocessing to prepare the input data for the subsequent machine learning training involves dimension reshaping, missing data imputation, outliers handling, normalisation and feature engineering etc. It can be summarised in Figure 2.3. The quality of raw SHM data varies. Due to their environmental conditions, the stability of sensors is normally severely affected on long-span bridges and their measurements can show inferior quality. Therefore, data preprocessing is an indispensable procedure to increase the accuracy of the algorithm, even though it is generally much more time-consuming than the machine learning training process.

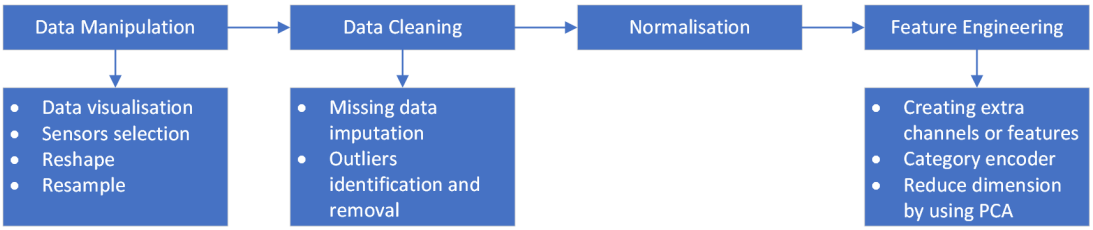


Figure 2.3: Data Preparation

2.4.2 Algorithms

In this section, a brief introduction is given for the fundamental concepts of machine learning and the algorithms of several ML models are presented.

As defined in the previous section, ML models will improve their accuracy by automatically learning from the existing data. Therefore, any ML algorithm has three essential parts: input data, the ML model and the output. The learning process is often referred to as "training". For SHM data analysis and some other applications, the ML approaches can be classified into 3 categories, supervised learning, unsupervised learning and semi-supervised learning:

- **Supervised learning**: The desired outputs are already prepared in the training set, often called the "labelled data". They will "guide" the training process. Therefore, the core of supervised learning algorithms is to minimise the discrepancies between the training output and the labelled data. A supervised learning model can make two kinds of predictions: classification and regression. In an SHM problem, classification can be the determination of the structure status, e.g. normal or critical; regression can be the prediction of the structural responses, e.g. deflections.

- **Unsupervised learning**: In an unsupervised learning model, the input set lacks the "labelled data". This is normally because the labels are unknown at the data collection stage. Therefore, the core of unsupervised learning is to identify the commonalities among the data and put them into different clusters. For example, novelty detection using vibration data is currently a common unsupervised learning method in SHM (Eltouny, Gomaa, & Liang, 2023).
- **Semi-supervised learning**: These models train with both labelled and unlabelled data. Typically, the amount of labelled data is limited, while most are unlabelled. A semi-supervised learning algorithm will be guided with labelled data and the unlabelled are used to uncover hidden structures in the data (Schwenker & Trentin, 2014). Within the context of SHM, a small amount of damage scenarios will be labelled to train a classifier. Then, the classifier is used to predict unlabelled data and create labels. The most confident unlabelled instances and their predicted labels are then added to the training set, and the classifier is retrained.

Some of the commonly used machine learning algorithms in an SHM study are presented below. Support vector machine (SVM) is a supervised learning model for binary classification. SVM was first proposed by Cortes and Vapnik (1995). Apart from linear classification, SVM is also capable of non-linear classification using kernel functions.

For regression problems, the decision tree is one of the popular models. A decision tree is a tree-like structure consisting of decision nodes, chance nodes and end nodes. This model starts with a decision node where a decision node represents a point where the data is split according to a certain condition. This condition is typically based on the values of one of the input features. Then in a chance node, outcomes are determined by chance, reflecting the uncertainty inherent in predictions or decisions. Finally, it

reaches the end nodes, representing the terminal points of a regression tree where no further splitting occurs. Decision tree is relatively easy to understand. Multiple decision trees can be constructed to improve its accuracy to form the random forest model, created by Ho (1995).

Artificial neural networks (ANNs) have received the most attention and development compared to other ML models in recent years. There are many types of ANNs developed by researchers which have shown great success in complicated prediction tasks. Figure 2.4 presents the structure of a very basic ANN model, the multilayer perceptron (MLP). There can be multiple hidden layers in the model and each hidden layer contains multiple neurons which use a non-linear activation function, e.g. sigmoid. Therefore, the function represented by the neuron can be written as:

$$a = f(z) = f(\sum_{j=1}^{m} x_j w_j + w_0) \tag{2.4}$$

where x is the input vector, w_j and w_0 are the weights, z is the pre-activation, linear function, f is the non-linear function and a is the output vector of the neuron, m is the total number of parameters. The output vector of the neuron can also be the input vector for the neurons in the next hidden layer. During training, MLP is repeatedly presented with the training data set. All the weights in the network are adjusted using a technique called *Backpropogation* (Rumelhart, Hinton, & Williams, 1985). The detailed mathematical procedures of the backpropagation is given by Bishop et al. (1995). The basic idea is to calculate the gradient of the loss, discrepancies between the desired outputs and the MLP outputs, with respect to the weights and then to adjust the weights in the direction of the steepest gradient. Backpropagation is the most important concept in ANNs.

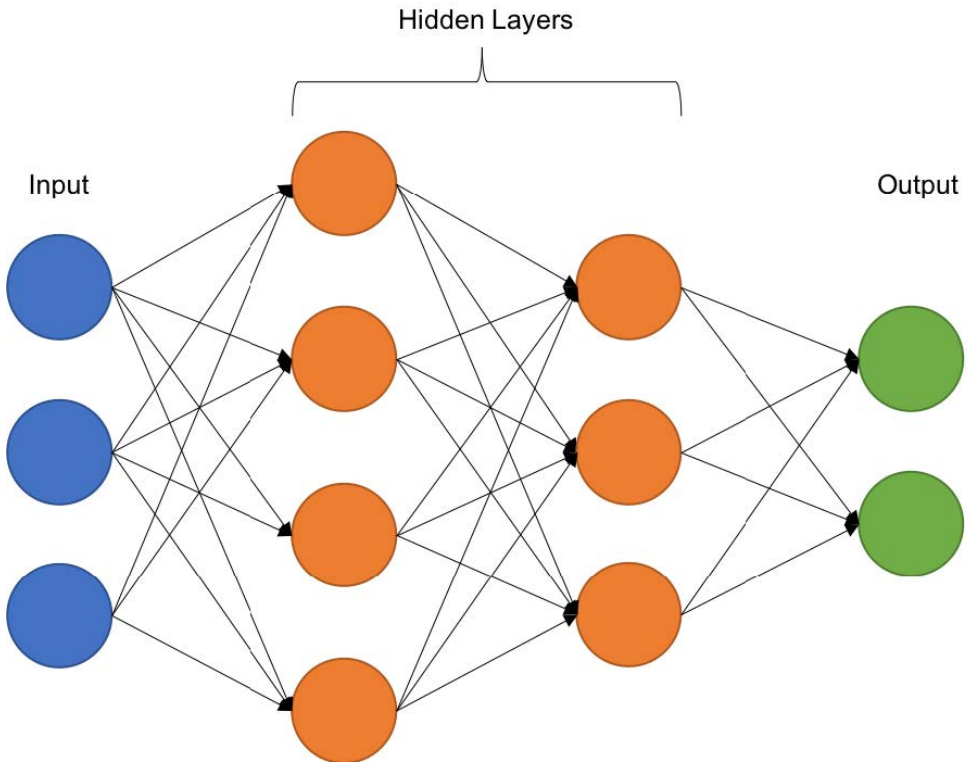


Figure 2.4: Structure of the multilayer perceptron (MLP)

Based on the idea of MLP, many different types of ANNs are proposed and proven feasible and more efficient in a certain specialised field. For example, Convolutional Neural Network (CNN), used in deep learning (DL), is popular in image processing tasks. Unlike MLP, a CNN incorporates additional convolutional layers prior to the fully connected hidden layers. These convolutional layers utilise a set of learnable filters to scan the input data and identify salient features. Deep learning is a variety of ANNs with multiple levels of representation, enabling complex functions and relations to be learnt as each layer in the deep CNN model is capable of identifying a certain type of motif (LeCun, Bengio, & Hinton, 2015). However, the complexity of the deep learning models is a great challenge for computer hardware to process. Therefore, to address this issue, Chellapilla, Puri, and Simard (2006) introduced the use of graphics processing units (GPUs) for deep learning training. Cireșan, Meier, Gambardella, and Schmidhuber (2010) proves GPUs are 40 times faster than standard microprocessors

training multilayer perceptron. As a result, the implementation of complex multi-layer deep neural networks has become possible with the development of GPU-based neural network training. There have been several successful deep neural network (DNN) applications. Krizhevsky, Sutskever, and Hinton (2017) proposed AlexNet achieved high accuracy in the ImageNet Contest 2012 with a proper approach to reduce the overfitting issue. Overfitting is when neural networks are too aligned with training data resulting in poor generalisation. The success of AlexNet is a milestone. Since then, DNN has been applied to various fields. WaveNet (Oord et al., 2016) is able to generate raw audio waveforms with realistic human voices and is used to generate Google Assistant voices in US English.

Recurrent Neural Network (RNN) is a hypothesis that the neural network utilises the previous outputs as part of the input vector to the same neuron. Therefore, RNN can process the input of variable lengths and take into account historical information. Therefore, RNN outperforms other ANNs in tasks like natural language processing and time-series forecasting (Tealab, 2018). Long short-term memory (LSTM) developed by Hochreiter and Schmidhuber (1997) uses a "gate" concept in the neural networks to control the preservation of information so that LSTM can learn from historical information, reducing the gradient vanishing effect in the traditional RNN model (Bengio, Simard, & Frasconi, 1994). Successful attempts have been made in time-series data forecasting using RNN in different fields. Cao, Li, and Li (2019) managed to make one-step ahead forecasting on decomposed financial time series using LSTM. There are similar studies using LSTM to predict petroleum production (Sagheer & Kotb, 2019), gold price (Livieris, Pintelas, & Pintelas, 2020). R. Zhao, Yan, Wang, and Mao (2017) applied a convolutional bi-directional LSTM (CBLSTM) model directly to raw sensory data from a machine health monitoring system of a computer numerical

control (CNC) milling machine. The results show the proposed model can predict the actual tool wear. In civil SHM studies, monitoring data are normally presented in the time-series format. Therefore, the use of the recurrent neural network to analyse SHM data from long-span bridges is feasible.

2.4.3 Output

The model output data for SHM problems are different depending on the research questions. For condition assessments, it can be the pre-defined health states of the structure, e.g. different anomaly patterns (Tang, Chen, Bao, & Li, 2019). For image-based crack detection, it can be binary classifications (crack or not) and some even explore the crack length and width (X. Yang et al., 2018). For regression problems, the output is generally the structure response predictions, e.g., displacement, strain and vibrations.

2.5 Data-based Methods

In this section, the machine learning based data driven SHM applications are reviewed. By comparison with model-based methods, data-driven models do not need the FE model to reflect the changes in the target structure. Condition assessment and damage detection are performed purely based on the monitoring data. As presented in the previous section, machine learning can further increase the scale of automation in data-driven analysis. Furthermore, the recent development in machine learning algorithms also enables ML to be used for more complicated tasks. Therefore, there have been many attempts to use ML for SHM data analysis. Different algorithms are surveyed in this part.

2.5.1 Multilayer Perceptron (MLP)

Min, Park, and Yun (2010) introduces an impedance-based structural damage detection method using MLP. Wireless sensors first collect impedance signals. Then, cross coefficients (CC) are used as the predefined damage index. Finally, an MLP model is trained on CC values to predict damage severity. This method successfully detects loose bolts and cracks on a steel bridge girder. Soyoz and Feng (2009) trained an MLP model based on the modal parameters from vibration measurements to study ageing and long-term bridge structural parameters using data collected from an in-service bridge, Jamboree Road Over-crossing, California, USA. Mehrjoo, Khaji, Moharrami, and Bahreininejad (2008) used MLP to evaluate the damage status of joints on truss bridges. Mode shapes and the natural frequency of the bridge were used as the input to the model. The location and severity of damages can be accurately determined by the MLP. However, due to the limits of computation capabilities at the time of this research, the most complicated structure in this article is a truss bridge with 16 joints. Similar studies on the damage localisation on truss bridges using MLP are also performed by researchers (Choi & Kwon, 2000; Nguyen, Tran, Bui-Tien, De Roeck, & Abdel Wahab, 2020). D. Zhao, Ren, Huang, and Xu (2019) used an MLP model to predict the mid-span girder deflections under the ambient temperature influence. Hakim and Razak (2013) proved the MLP were capable of using the first five natural frequencies as input to determine the severity of the damage on the steel bridge girder. Pawar, Venkatesulu Reddy, and Ganguli (2007) used MLP to detect the damage location and size using a vector of Fourier coefficients of the mode shapes on a fixed-fixed beam. Other uses of MLP damage detection on beam-like structures are also reported (Rosales, Filipich, & Buezas, 2009; Tran-Ngoc, Khatir, De Roeck, Bui-Tien, & Wahab, 2019; J. Zhao, Ivan, & DeWolf, 1998). Several other studies are listed in Table 2.3.

Table 2.3: Other MLP applications on bridges SHM

Reference	Structure	Input	Output	Preprocessing
Taha, Noureldin, Osman, and El-Sheimy (2004)	FE-model of a prestressed concrete bridge	Wavelet Normalised Index	Damage Ocurrance	Wavelet Transform
Concepcion and Ilagan (2019)	Lab test platform	Temperature-compensated frequencies	3 damage classes	PCA
Gordan et al. (2020)	Slab-on-girder bridge	Frequencies of first four modes	Damage Types	Transform Formatting Cleaning, Feature Engineering

2.5.2 Convolutional Neural Network (CNN)

Due to the huge success of CNN in image processing, its application in structural health monitoring has been extremely popular in recent years. Since structural damages can appear as cracks which can spread, enlarge and eventually lead to the collapse of the structure, detection and handling of cracks are vital (Munawar, Hammad, Haddad, Soares, & Waller, 2021). Traditionally, cracks on structures are examined and detected by visual inspections by experts. However, this method is labour-demanding, expensive and subjective to human errors. With the development of image processing algorithms, automatic crack identification is now possible with machine learning methods, especially with CNN. Compared with MLP where each neuron is connected to all neurons in the following layer, as shown in Figure 2.4, (dramatically increasing the trainable parameters and inefficiency) CNN utilises convolutional layers before fully connected layers where different filters slide across the image, allowing patterns to be recognised regardless of their locations in the image. Therefore, the trainable parameters are far less for a CNN model. As a result, a CNN model can have multiple convolutional layers to identify multiple features in an image with a smaller size of parameters to train and also achieve high accuracy. There have been multiple successful attempts using CNN in crack detection. Table 2.4 shows the studies analysed in this section.

Image-based Damage Detections

Cha et al. (2017) used CNN to develop a model to detect concrete cracks from images. In their experiments, 277 images were manually labelled as "cracked" or "intact" to train the CNN model. Figure 2.5(a) shows the original photo of the concrete surface. The trained model achieved 97% accuracy. The shape and location of cracks can be identified using CNN, as shown in Figure 2.5(b). Compared with the traditional Canny edge detection, Figure 2.5(c) and Sobel edge detection, Figure 2.5(d), the proposed CNN detection method can provide clearer crack information. The proposed method also benefits from the simple data preparation needed. Modarres, Astorga, Droguett, and Meruane (2018) applied CNN to concrete surface images collected from a concrete bridge. Two sets of images are used to verify the feasibility of the CNN model, one clean set of well-controlled surfaces with different lighting conditions representing concrete in new bridges and one noisy set of textured concrete surfaces with unclear cracks representing in-service structures in a visual inspection. The results show that the proposed CNN outperforms other machine learning models, e.g., MLP, random forest, SVM with a non-linear kernel in crack detection for both image sets. Zhang, Yang, Zhang, and Zhu (2016) built a deep CNN model enabling the automatic detection of pavement cracks. Due to the low contrast between cracks, surroundings, and shadows, such a task can be challenging for traditional methods. The proposed deep CNN model achieved satisfactory performance compared with the SVM and Boosting method. Similar studies are also performed by Alipour, Harris, and Miller (2019); F. Chen and Jahanshahi (2017); Dung et al. (2019); H. Xu et al. (2019). Attempts have been made by using other popular deep CNN models for concrete crack detection, e.g. ResNet-50 (Kruachottikul et al., 2021), AlexNet (X. Zhao, Li, Su, Zhou, & Loh, 2018), VGG-16, GoogLeNet (Zhu & Song, 2020).

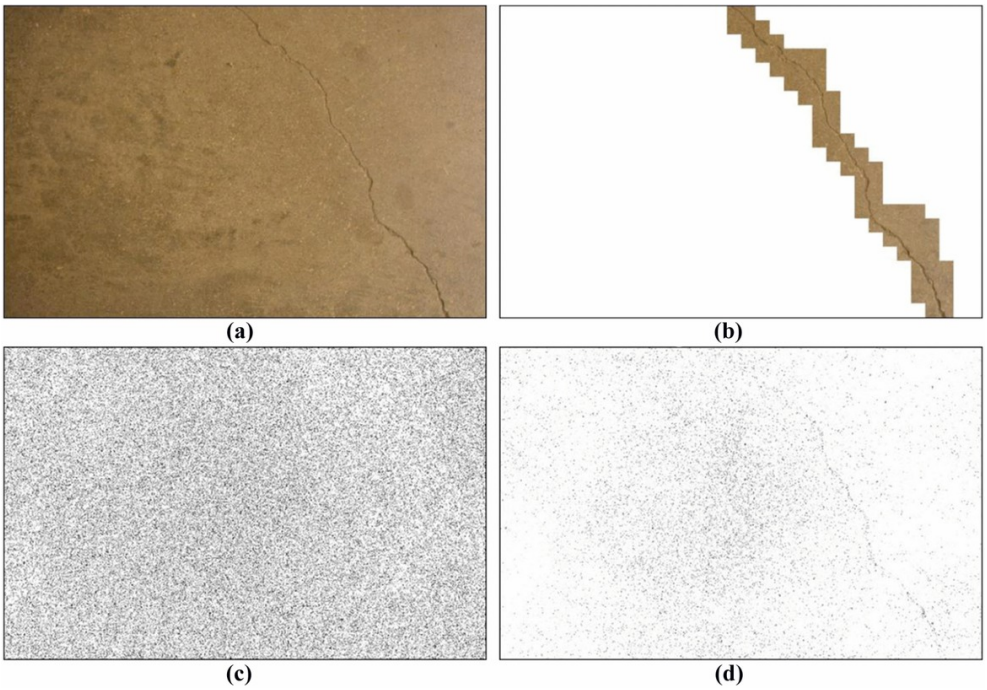


Figure 2.5: Proposed CNN model compared with other detection methods (Cha et al., 2017)

Apart from directly using the CNN or deep CNN models, researchers incorporate different algorithms in the CNN layers to further improve accuracy or capability in complex crack detection problems. H. Xu et al. (2019) introduced the Atrous Spatial Pyramid Pooling (ASPP) module in the CNN model. ASPP can avoid loss of information when the image is downsampled by traditional convolutional layers. This ASPP CNN model was trained on 4856 concrete surface images and the results outperform the conventional CNN model. Several studies have used Region-based CNN (RCNN) for detection. In this model, the output contains the classifications and bounding boxes showing the region of interest (ROI), e.g., the locations of cracks. In this way, the size of cracks can be quantified by width and height. Suh and Cha (2018) successfully used a faster R-CNN model to detect steel corrosion and concrete cracks automatically and achieved real-time monitoring. Similarly, X. Zhao et al. (2018) proposed a comprehensive inspection method for bridge maintenance. First, an

AlexNet was used to classify different types of bridges, e.g., suspension, cable-stayed bridges. Second, a Faster-RCNN is trained to identify different bridge components (e.g. tower, deck). Finally, concrete cracks are detected by a GoogLeNet. However, this research lacked discussions on the link among the three parts.

Time-series Classification

Even though most studies on CNN application are image-based crack detection, a few researchers applied CNN to time-series SHM monitoring data. Bao et al. (2019) transformed time series signals from a long-span bridge SHM system into image vectors and manually labelled these images for different anomaly patterns, as shown in Figure 2.6. Each subfigure represents data from a different channel. Then, these images served as input to CNN to simulate the manual inspection process. The trained CNN can detect potential anomalies in large volumes of SHM data. Similarly, Tang et al. (2019) split and visualised the raw time-series monitoring data in the time and frequency domain. Then, CNN is used to detect anomalies in these visualised data. The proposed methods are validated using acceleration data from an in-service long-span cable-stay bridge. Different strategies regarding time-series data visualisation are also proposed by others (G. Liu, Niu, Zhao, Duan, & Shu, 2022; Shajihan, Wang, Zhai, & Spencer Jr, 2022). The implementation of 1D-CNN in some research (Abdeljaber, Avci, Kiranyaz, Gabbouj, & Inman, 2017; Ni et al., 2019) enables the direct use of time-series data without visualisation. Traditional CNN models take imagery data as input, so the input shape is *(height, width, depth)*, where *depth* is the colour channels in an image. However, the time-series SHM monitoring data is one-dimensional. Using 1D-CNN allows the convolutional kernel to slide in one direction only. Therefore, 1D-CNN can be used on SHM data without data visualisation and the 1D convolutional layer manages to extract features from the time-series data.

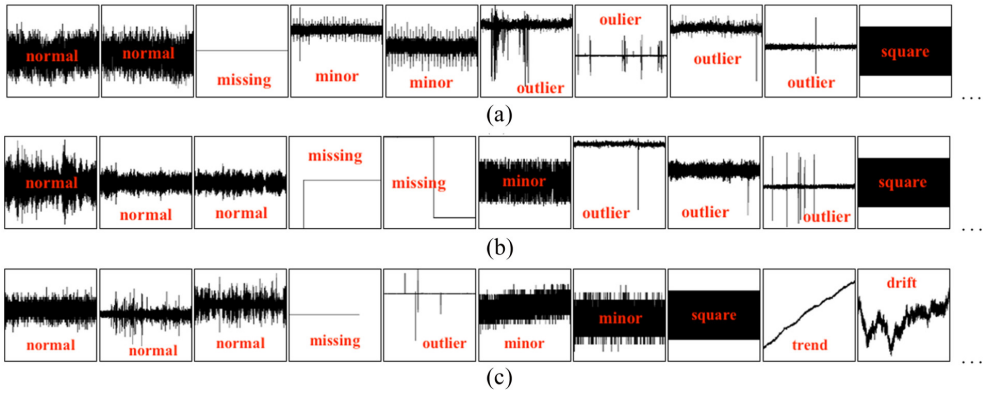


Figure 2.6: Examples of data visualisation with anomalies labelled (Bao et al., 2019)

Table 2.4: CNN applications on bridges SHM

Algorithms	Usual Applications	Structure	Preprocessing	Input	Output	Results	Reference
CNN	Canny and Sobel edge detection	Concrete surface	Labelling	Images of concrete surface	Crack existence, locations	Accuracy: 97%	Cha et al. (2017)
CNN	PCA, SVM, MLP, Random Forest	Concrete surface on bridges	Labelling, Add noise	Images of concrete surface	Damage types	Accuracy: 99.6%	Modarres et al. (2018)
Deep ConvNet	SVM, Boosting	Pavement surface	Normalisation	Images of pavement surface	Crack existence	Precision: 0.8696 Recall: 0.9251	Zhang et al. (2016)
ASPP-CNN	Resnet, Vgg	Bridge cracks	Reshape, labelling	Images of bridge cracks	Crack existence	Accuracy: 96.37%	H. Xu et al. (2019)
FCN	(No discussed)	Concrete surface	Reshape, labelling	Images of concrete surface	Crack existence, locations	Precision: 90%	Dung et al. (2019)
NB-CNN	LBP-SVM	Steel surface	Reshape, labelling	Video frames with noise (non-crack marks)	Crack bounding boxes	Hit rate: 98.3%	F. Chen and Jahanshahi (2017)
CrackPix (CNN)	Edge detection	Concrete surface	Upsampling, resize, labelling	Images of concrete surface	Crack existence	Crack: 92% Non-crack: 99.9%	Alipour et al. (2019)
Modified ResNet-50	ResNet-50, GoogLeNet	Concrete surface	Image processing (enhance, split, expand)	Accelerations	Defect existence, classification, Severity prediction	Defect: 90.4% Classification: 81% Severity: 78%	Kruachottikul et al. (2021)
AlexNet (CNN), Fast R-CNN, GoogLeNet	(Not discussed)	Bridges	Labelling	Images of bridges	Bridge types, bridge components, Cracks existence & locations	AlexNet: 96% Fast RCNN: 90.45% GoogLeNet: 99%	X. Zhao et al. (2018)
CNN	Deep MLP	A long-span bridge	Data visualisation,	Visualised time series data	7 types of anomalies	Overall accuracy: 93.5%	Tang et al. (2019)
CNN	Deep MLP	A long-span cable-stayed bridge, China	Data visualisation, decomposed into 3 channels	Visualised & decomposed time series data	7 types of anomalies	Overall accuracy: 95.07%	Shajihan et al. (2022)
GANomaly-CNN	(Not discussed)	A long-span cable-stayed bridge, China	Data visualisation, decomposed into 3 channels using FFT and GAF	Visualised & decomposed time series data	7 types of anomalies	Global accuracy: 98.2%	G. Liu et al. (2022)
1D-CNN	1D-DCT	A suspension bridge, China	Labelling	Time series data	Binary classification	F-measure: 97.53%	Ni et al. (2019)

2.5.3 Recurrent Neural Network (RNN)

For its ability to retain historical information from the input sequence, RNN is capable of tasks regarding natural language processing and time-series data analysis. There have been many attempts to use RNN for time-series forecasting. However, the application of RNN in civil SHM problems is rare compared to CNN. In an SHM system, data corruption is inevitable despite the advances in sensor hardware in recent years. There have been a few attempts using RNN to correct data errors. Jeong et al. (2019) proposed a method using bidirectional recurrent neural networks (BRNN) for sensor data reconstruction. The assumption is that errors can be corrected by data sampled by other sensors nearby. The bidirectional feature used in this article allows the RNN to learn correlations between different sensor data from two directions, past to future and future to past. The results show the proposed BRNN method can reach more accurate estimations of the missing parts, compared to SVM, PCA, and RNN. One test example is given in Figure 2.7 where missing data from 2 sensors are predicted using nearby sensors. However, one drawback is observed. The training time of BRNN is significantly higher than other models, in the meantime, the accuracy improvement is relatively low.

Regression Analysis with RNN

For an in-service bridge, structural responses are affected by environmental and operational conditions. Therefore, the thermal-induced, traffic-induced and wind-induced bridge responses are popular research topics in the SHM studies. There have been several papers investigating the recurrent neural networks applications in the above topics. Yue et al. (2021) successfully used LSTM to predict thermal-induced deflections of a long-span bridge based on temperature data. The prediction results are shown in Figure 2.8. LSTM is also used for the non-linear mapping between the

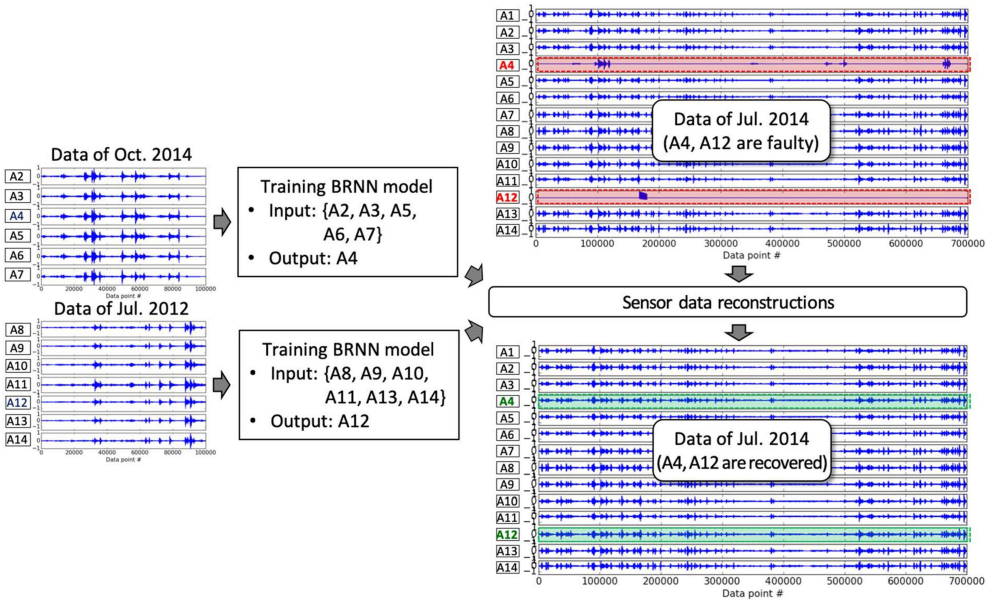


Figure 2.7: Missing data recovery with BRNN (Jeong et al., 2019)

dynamic displacement and the vehicle-induced strain (H. Zhao, Ding, Li, Sheng, & Geng, 2020). S. Li, Li, Laima, and Li (2021) investigated buffeting responses induced by wind effect using LSTM. In this research, the wind speed, direction and attack angle are preprocessed and used as input data. The vertical acceleration of the deck is used as the output. The prediction results are then compared with the buffeting responses calculated based on the Davenport-Scanlan's model and finite element method (FEM). LSTM predictions are more accurate than the conventional FEM model in both time and frequency domain. Similar research shows that the wind-induced bridge action can also be accurately predicted by LSTM (T. Li, Wu, & Liu, 2020). A. Guo, Jiang, Lin, and Li (2020) predicts girder deflections based on temperature, crack, humidity, strain and historic deflection data with a LSTM network. Despite the high accuracy recorded in this research, the proposed method only manage to predict deflections 3 steps further using a dataset of 8735 time steps for training and testing. The model is highly converged to the given dataset. Therefore, the generalisation on other data is questionable. There is one research on outlier detections using RNN. Mousavi and Gandomi (2021) proposed a damage detection scheme using

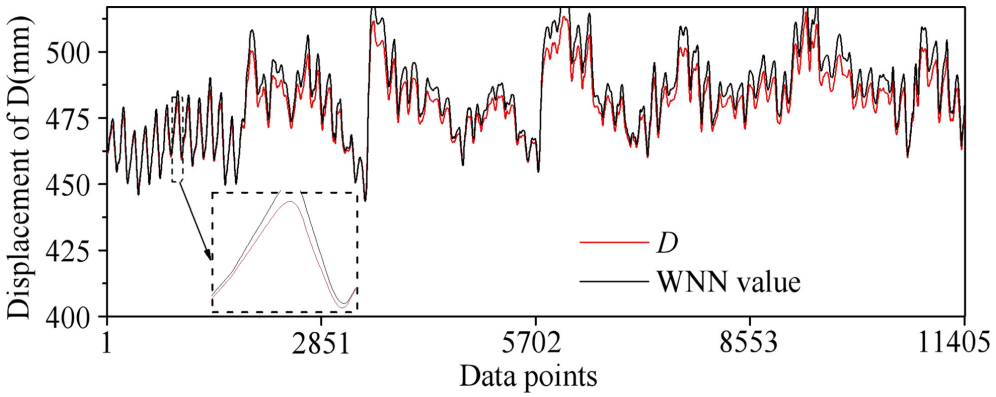


Figure 2.8: Displacement predictions with LSTM (Yue et al., 2021)

bidirectional LSTM model to learn the Mahalanobis distance, the distance between a data point to distribution, of the natural frequency of the bridge when the bridge is under a healthy state. Later, the model was verified by monitoring data with damage occurrence, where large errors were observed, indicating outliers existence. Different from other research where raw time-series data are used for training, the input data was preprocessed with Variational Mode Decomposition (VMD) to denoise and remove any seasonal patterns in the signals. Then, a Minimum Covariance Determinant (MCD) is used to obtain the Mahalanobis distance for training.

Damage classification

Damage classification using RNN is also proposed by different researchers. J. Yang et al. (2020) used a Hierarchical CNN and gate-recurrent unit (GRU) method (HCG), an algorithm combining the feature of both convolution and recurrent layers, to perform damage detection on acceleration data from a bridge scale model. Concentrated force was applied to the main span creating cracks to represent structural damage. The prediction accuracy was compared with MLP, CNN and RNN. The proposed HCG model not only outperformed all other neural network methods on this damage detection task but also showed similar training time compared with other models. Sony, Gamage, Sadhu, and Samarabandu (2022) divided acceleration data into multiple

fixed-length windows to reduce the data dimensions and increase the size of training data sets. Each window is labelled with damage status. Then an LSTM model is used to classify acceleration responses. The model successfully predicted the damage class and probabilities of damage over the target bridge (damage localisation).

Table 2.5: RNN applications on bridges SHM

Algorithms	Usual Application	Structure	Preprocessing	Input	Output	Reference
BRNN	PCA, MLP	Telegraph Road Bridge, Michigan, US	Normalisation, Scaling	Vibration responses from nearby sensors of the target	Missing Vibration responses of the target	Jeong et al. (2019)
LSTM	Liner regression	A steel truss girder cable-stayed bridge over the Yangtze River, China	Normalisation	Temperature or Deflection+ Temperature	Displacement	Yue et al. (2021)
LSTM	SVM with Gaussian kernel	A 25m box-girder bridge, China	Normalisation	Temperature	Strain or Displacement	H. Zhao et al. (2020)
LSTM	Davenport–Scanlan's buffeting force model	A suspension bridge, China	Normalisation, Feature extraction	Wind speed, Wind direction	Buffeting responses	S. Li et al. (2021)
LSTM	Quasi-steady theory	A unit-length section of a bridge deck	Normalisation	Deck displacement	Aerodynamics forces and responses	T. Li et al. (2020)
LSTM	(Not discussed)	A bridge in Hubei Province, China	Normalisation	Humidity, Temperature, Crack, Strain, Deflection	Deflection	A. Guo et al. (2020)
BiLSTM	PCA-based method	Z24 Bridge, Switzerland	VMD, FastMCD	Mahalanobis distances of vibration data	Mahalanobis distances of vibration data	Mousavi and Gandomi (2021)
CNN-GRU	CNN or RNN only	A scale model of a bridge	Normalisation, Labelling	Vibration data	4 damage states	J. Yang et al. (2020)
LSTM	1D-CNN	Z24 Bridge, Switzerland	Normalisation, Labelling	Vibration data	3-4 damage classes	Sony et al. (2022)

2.5.4 Other ML methods

Apart from the huge number of attempts at using neural networks for damage detection and condition assessment on bridges - other machine learning models, e.g. SVM and random forest, are evaluated for various topics. The studies in this section are summarised in Table 2.6.

Support Vector Machine

SVM is used to determine damage location and severity. For example, He and Yan (2007) introduced the wavelet basis as the kernel function of SVM. The proposed wavelet support vector machine was then deployed to detect damages under ambient excitation. H. Pan, Azimi, Yan, and Lin (2018) used a radial basis function (RBF) kernel SVM to perform classification on damaged and undamaged states of a cable-stayed bridge. Compared with other SVM applications, different data preprocessing methods are investigated in this research. It was found that wavelet transform has a higher accuracy in noisy data, whilst Teager-Huang transform (THT) performs better on regular signals with limited noise. B. Li, Ushiroda, Yang, Song, and Xiao (2017) used SVM on the impact-echo data, an acoustic inspection method, collected from a road bridge in New York to classify deck damage states. In this research, the data are preprocessed with Fast Fourier Transforms (FFT) and Wavelet Transforms (WT). An attempt was made to investigate the relationship between fatigue damage of hangers on a suspension bridge and live traffic loads monitored by the weigh-in-motion (WIM) system (Y. Deng, Zhang, Feng, & Li, 2021). The proposed SVM model accurately predicted the fatigue damage on the target hanger using traffic volumes and equivalent traffic loads. Furthermore, it was found that the increasing traffic loads can greatly accelerate the fatigue damage on hangers. Image-based cable defect identification using SVM was also reported (Bu et al., 2015).

Random Forest

Random forest is another machine learning algorithm that is widely used in SHM studies. The Random Forest method is a type of regression tree approach that employs bootstrap aggregation and the random selection of predictor variables to attain a high level of predictive accuracy (Rigatti, 2017). The popularity of random forests is attributed to attributed to its versatility in addressing various prediction challenges and their minimal requirement for parameter tuning. Furthermore, the algorithm can achieve high accuracy on high-dimensional feature spaces (Sagheer & Kotb, 2019).

Lei, Sun, Xia, and He (2020) proposed the random forest method to predict the damage state for regional bridges under seismic conditions. The model achieved over 90% accuracy on damage classification. Random forest is also used for imputation to reconstruct the missing monitoring data from the Forth Road Bridge (D. Xu et al., 2019).

Clustering

Unsupervised learning methods are also employed in damage detection studies due to the large amount of raw SHM monitoring data and labour-intensive labelling process required for any supervised learning methods. Diez et al. (2016) used K-means based cluster on joints vibration data from the famous Sydney Harbour Bridge. The proposed approach managed to group vibration data showing similar behaviour. Therefore, the vibrations under undamaged and damaged states are grouped into different clusters, allowing damaged joints to be detected. A similar research used the K-medoids clustering method to perform damage detection on the Z24 Bridge vibration data (Sarmadi, Entezami, Salar, & De Michele, 2021), on the PI-57 Bridge, France (de Almeida Cardoso, Cury, & Barbosa, 2019).

In this section, several other machine learning algorithms used in SHM studies are discussed. SVM is very popular for its non-linear kernel function on complex classifications.

Table 2.6: Other MLP applications on bridges SHM

Algorithms	Usual Application	Structure	Preprocessing	Input	Output	Reference
SVM	FE model-based fatigue analysis	Nanxi Suspension Bridge, China	Input: traffic volumes and traffic weights Output: fatigue damage generated by FEM using WIM data	Bridge-site traffic loading parameters	Fatigue damage of hangers	Y. Deng et al. (2021)
PSO-SVM	Zernike moments, Gabor filter, Wavelet	Cables on long-span bridges	Texture extracted by GLCM, Manual labelling	Grey-scale cable images	4 damage classes	Bu et al. (2015)
SVM	Visual inspections, GPR, PSA	Flushing 149st Bridge, New York, US	FFT, WT	Impact-echo data	5 damage types	B. Li et al. (2017)
SVM	Physics-based methods	Manavgat Cable-stayed Bridge, Turkey	WT, HHT, THT	Vibration data	15 damage types	H. Pan et al. (2018)
Random Forest	Traditional methods using material stiffness and boundary conditions	Nanli River Bridge, China	Labelling	Ground motion data, Bridge dynamic parameters, Bridge design parameters	Seismic damage states	Lei et al. (2020)
Random Forest	Linear regression	Forth Road Bridge, UK	Outlier removal	Strain, Temperature	Missing strain data	D. Xu et al. (2019)
K-means Clustering	Manual labelling	Sydney Harbour Bridge, Australia	k-Nearest Neighbours(kNN), FFT	Vibration data	Clusters	Diez et al. (2016)
K-medoids Clustering	Manual labelling	Z24 Bridge, Switzerland		Vibration data	Clusters	Sarmadi et al. (2021)
K-medoids Clustering	Manual labelling	PI-57 Bridge, France	Discrete Fourier Transform	Vibration data	Clusters	de Almeida Cardoso et al. (2019)

2.6 Summary

The development of SHM on long-span bridges has grown rapidly in recent years. Therefore, the demand for studying the data collected from these SHM systems is high. In this section, a detailed review is presented on the following topics: premature failures on bridges and their causes, development of the SHM systems, model-based finite element methods applications, machine learning backgrounds and structures, data-driven methods applications in bridge SHM.

Model-based methods for long-term bridge SHM data analysis have limitations, primarily due to the need for high-frequency data and a full-scale FE model. In contrast, data-driven methods, especially machine learning, are becoming prevalent due to their ability to handle complex datasets without a full structural replica. The advancement of machine learning has enabled its application in bridge SHM, offering enhanced generalisation over traditional statistical models. This review highlights various machine learning models in bridge SHM, detailed in Tables 2.3, 2.4, 2.5, and 2.6. Typically, machine learning in SHM focuses on condition assessment, such as damage state classifications, and specific damage detections like image-based concrete crack detection.

The basic ANNs, like MLP, excel in areas such as mode shape analysis and offer better prediction accuracy than some methods. However, its simplicity makes it best suited for classifying damage in simple laboratory structures. MLP's reliance on fully connected layers means that adding more neurons or layers significantly increases trainable parameters, hindering its deep learning potential. While modern SHM systems on bridges capture both structural responses and environmental data, like temperature and traffic, MLP struggles to recognise trends from such complex multi-channel data.

Convolutional Neural Network (CNN) is the most popular neural network model in SHM studies in recent years. As a deep learning model, it is capable of processing complicated multi-channel training data. The convolution layers not only extract features from the input dataset but also reduce its dimension. As a result, the scale of the trainable parameters is controllable and the development of a deep neural network with convolutional layers is possible. In the SHM field, the most successful application is image-based crack detection. The automation of concrete crack detection becomes possible with this ML method. One other feature of image-based damage detection methods is that it is a universal solution not only applicable to bridges but also other civil infrastructures, e.g. dams, high-rise buildings. Furthermore, from Table 2.4, the data preprocessing is relatively straightforward, normally only involving labelling and reshaping. For its exceptional performance in image processing, some researchers attempted to visualise the time-series SHM data which then can serve as the input data to CNN. The results also indicate successful detections of structural damages. Therefore, in the data pre-processing stage, the manual labelling of the training data is mandatory. As a result, one notable drawback of CNN is the requirement for data labelling which can be both time-consuming and monotonous.

RNN's application in SHM studies, particularly for regression analyses, is emerging. While many studies explore the correlation between environmental and operational conditions, like thermal effects and structural responses, RNNs can predict future structural behaviours, such as deflections. Predicting exact structural responses is challenging, and LSTM, a type of RNN, is prone to overfitting. However, RNN's strength lies in providing direct insights into structural behaviour, allowing engineers to apply further probability models to assess potential damage. Notably, RNNs in bridge SHM typically require less data preprocessing than CNN or MLP models, making them adaptable to various datasets.

As for other machine learning methods, the applications of SVM and random forests are limited by the requirements of complicated data preprocessing steps, as shown in Table 2.6. K-means clustering is the most popular unsupervised method to group unlabelled data. It is possible to replace manual labelling required for supervised and semi-supervised learning with K-means clustering.

Considering that the data are sourced from two in-service bridges in this project, data-based approaches, e.g. RNNs, Random Forest, will be used as the main research methods. FE model of the QC will also be employed to complement and support the data-driven research approaches.

2.7 Thesis Structure

Chapter 3 introduces the research targets of this thesis: the Forth Road Bridge and the Queensferry Crossing, both spanning the Firth of Forth in Edinburgh, UK. The chapter provides an overview of the structural health monitoring (SHM) systems implementation on these bridges and discusses the preprocessing techniques applied to the SHM data, thereby facilitating subsequent research. Additionally, the chapter addresses issues related to data integrity and presents preprocessing methods that have been verified using raw SHM data.

Chapter 4 proposes a traffic load simulation method based on traffic flows and vehicle attributes on the Queensferry Crossing. The process initiates with the creation of a vehicle fleet. Subsequently, vehicle trajectories are determined using a car-following model combined with a lane-changing model. The deck deflection at specific points is then obtained by integrating these trajectories with the bridge's influence line. The chapter also uses structural health monitoring (SHM) data from the Queensferry Crossing to validate the precision of the proposed simulation approach.

Chapter 5 delves into the utilisation of a neural network model for examining vehicle-induced deflections. The simulation outcomes from Chapter 4 serve as the training, validation, and testing datasets for the model. A subsequent parameter analysis is conducted to scrutinise the impact of various hyper-parameters on the model's generalisation accuracy.

Chapter 6 commences with a finite element analysis aimed at identifying critical sections on the bridge where substantial bending moments are observed. This is followed by a detailed temperature field analysis focusing on key areas of the QC, including the south midspan deck section, the central tower, and the longest stay cable. The chapter finishes with the estimation of extreme temperature differences for both the bridge deck and the central tower, employing a Generalised Pareto Distribution based on one-year SHM data from the QC. These estimations are subsequently compared with existing design codes.

Chapter 7 initiates with the utilisation of the QC's finite element model to assess the impact of thermal loads on the bridge deck when uniform temperature changes are applied to various structural components, such as the deck, towers, and cables. The vertical displacement of the deck is compared under these three load cases. Subsequently, wavelet transform techniques are employed to explore the correlation between temperature variations and midspan deflections. The chapter concludes with the application of a neural network model to predict thermal-induced deflections.

Chapter 3

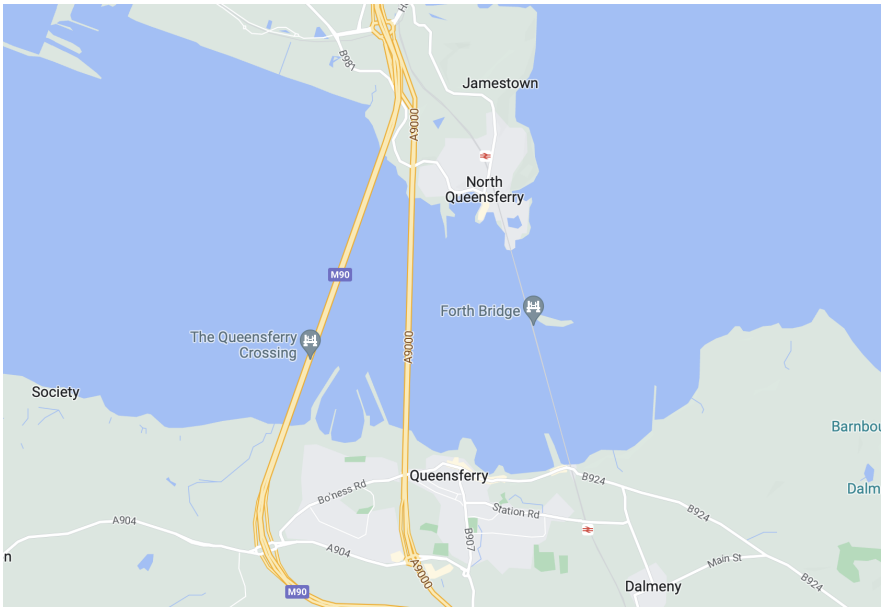
Data Preparations

3.1 The Bridges

Currently, there are 3 bridges, as shown in Figure 3.1 crossing the Firth of Forth, an estuary located at the north of the city of Edinburgh, United Kingdom. One of them is the Forth Bridge, a cantilever rail bridge completed in 1890 and now a UNESCO World Heritage Site. The other two bridges are road bridges, the Forth Road Bridge and the Queensferry Crossing, our research targets.

The Forth Road Bridge (FRB), a suspension bridge constructed in the 1960s, connects Edinburgh at South Queensferry and Fife at North Queensferry. FRB was designed to replace the ferry service to carry motor vehicles across the Firth of Forth. The bridge has a total length of 2512m and the main span is 1006m and at the time of opening was the longest suspension bridge outside the United States. The FRB has been a major link in the Scottish road networks since then. There have been several inspections since the early 21st century, conducted by the Forth Estuary Transport Authority (FETA). For example, in 2003, an internal inspection was carried out on the cables. Even though it was confirmed the cables were suffering 8% - 10% loss of strength at the time, it was predicted that traffic restrictions would be required in 2014 and the bridge would face premature closure if no actions were taken (Colford & Cocksedge, 2008). Acoustic monitoring and dehumidification of cables were also suggested.

(a) The Forth Bridges (www.theforthbridges.org)



(b) The Forth Bridges location (maps.google.com)

Figure 3.1: The Forth Bridges

In December 2015, several truss end links which connects the bridge deck and the tower were discovered with major cracks, as shown in Figure 3.2. Then, the bridge was forced to close for three weeks for the repair work to be carried out. In the meantime, a structural health monitoring was planned and installed on the FRB to provide live monitoring data for engineers to assess the condition of the bridge.

A second Forth crossing bridge was proposed when several structural performance issues were discovered on the FRB. On the other hand, the current traffic volume on the FRB can reach 65000 per day at its peak time, far exceeding the design traffic load which can result in accelerated structural degradations (Colford & Cocksedge, 2008). The new bridge, the Queensferry Crossing (QC), formerly called the Forth Replacement Crossing, located at the west of the existing FRB, is a three-tower cable-stayed bridge, opened in August 2017. At the same time, the original FRB remains open with restricted access to public transport, cyclists and pedestrians only. The Queensferry Crossing has an overall length of 2.7 kilometres and its tallest tower reaches 207m. Currently, the QC is the longest three-tower cable-stayed bridge in the world. Different from the FRB, the QC was designed with a comprehensive SHM in place at its construction stage. Therefore, the QC can provide more detailed SHM data from its opening.

The data available for this project related to the first year of collection from the QC - before the final snagging of the system was undertaken by the contractor. The timing of this research project meant the data available was from July 2020 to July 2021. Only limited data was available from the FRB.

3.2 SHM data

As discussed in the previous sections, the SHM systems were installed on both bridges to provide live information for engineers to assess the bridge conditions. Therefore, a great number of sensors were fitted on all bridge components, including the towers, decks and cables. Figure 3.3 shows the layout of all the sensors on the QC. These sensors monitor not only structural responses, e.g. deck deflections but also their environmental and operational conditions, e.g. deck temperature, rainfall and traffic. Table 3.1 shows the types and number of sensors installed on the FRB and the QC.

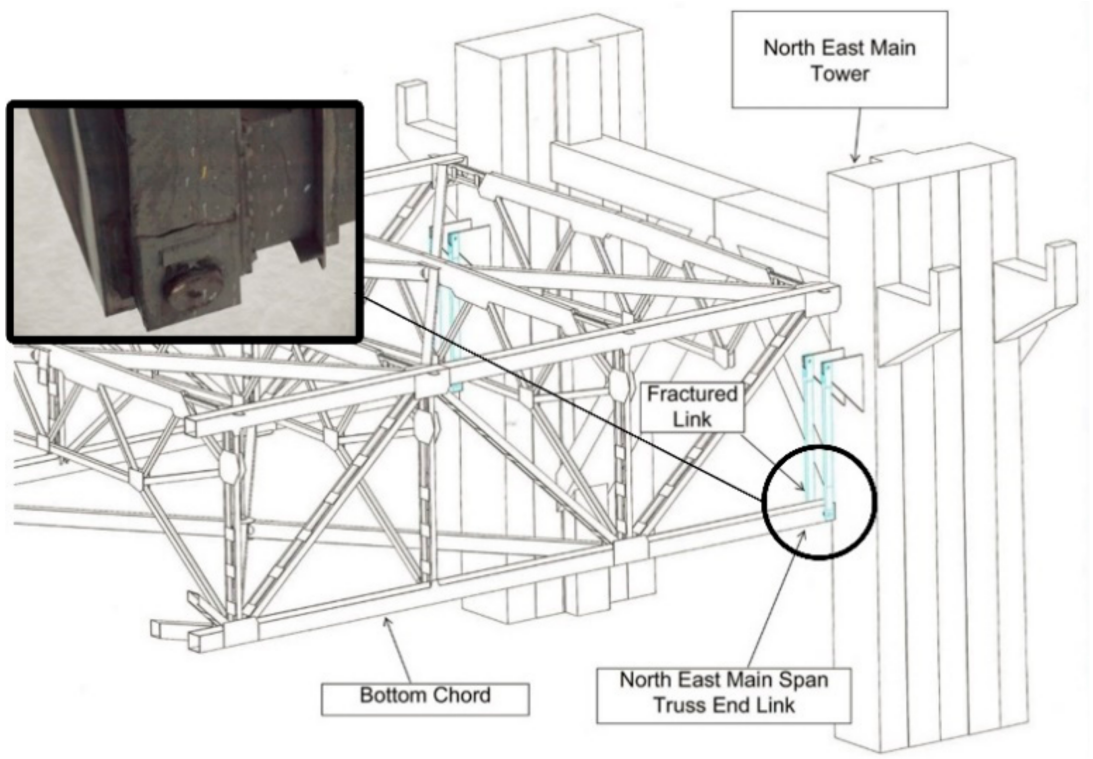


Figure 3.2: Cracked north east truss end link (Ni et al., 2019)

Currently, there are 192 sensors on the FRB and 2184 sensors on the QC. Since the QC is a newly built bridge, the SHM was well designed with much more sensors and new types of sensors, e.g. corrosion sensors for long-term reinforced concrete deck monitoring. Various types of sensors monitored different data on the bridges. For example, GPS stations record the displacement of the deck in three directions. Strain gauges are directly installed on deck girders. Dynamic weigh-in-motion systems record the attributes of each passing vehicle. On the QC, 45 data acquisition units (DAUs) are installed on-site for primary data processing and temporary storage. Two main servers are fitted in the south abutment and bridge management office at South Queensferry, UK. The server has a capacity of 185-day data. Furthermore, all data are archived on the Microsoft Azure server. Figure 3.4 describes the layout of a DAU and its communications with sensors and different systems.

Table 3.1: Sensor types

SHM Sensor Type	Queensferry Crossing	Forth Road Bridge
Accelerometers	102	-
Air temperature	13	2
Anemometer	11	2
Asphalt temperature	40	6
Barometers	2	1
Bearing gauges	16	8
Bearing pressure sensors	-	8
Concrete deck temperature	70	-
Concrete tower temperature	46	-
Corrosion sensors	360	-
Displacement transducers	32	-
Dynamic Weigh-in-motion	96	64
GPS location	21	10
Rainfall gauges	2	1
Relative humidity sensors	12	34
Strain gauges	887	128
Stay cable temperature	56	-
Steel surface temperature	158	32
Cable acoustic monitoring	-	116
Tiltmeters	48	16

For both SHM systems, the monitoring data are downsampled to 1Hz to store in the cloud server even though the sensors' original frequency could be higher (e.g. 20Hz, 50Hz, 100Hz) by taking the mean of original data in 1 second. When analysing SHM data, a low sampling rate (e.g. 1Hz) is sufficient for static loads (e.g. temperature) while a high sampling rate is normally needed for dynamic analysis using accelerations. On the QC, the storing sampling rate will be automatically adjusted to the sensor's original frequency under extreme environmental conditions. The lower sampling rate is designed to save storage space and maintenance costs. Despite this data storage strategy, the SHM system on the QC can still produce over 202GB of data in just one day.

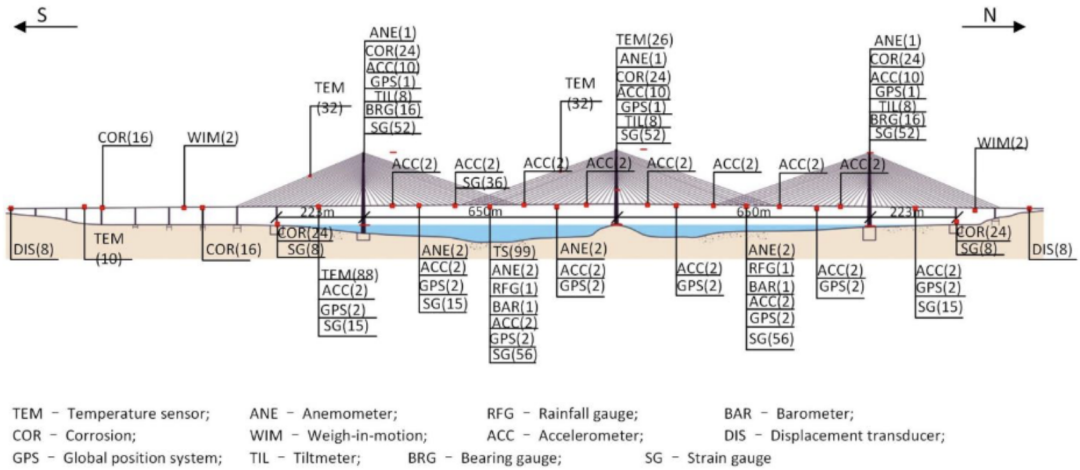


Figure 3.3: Sensor layout on the Queensferry Crossing

Table 3.2 shows examples of raw data from the FRB DWIM system. A weigh-in-motion system is designed to measure vehicle axle number and gross weight when the vehicle passes the system without stopping. The DWIM data are different from other SHM sensors as the DWIM system is triggered by each passing vehicle. Therefore the data are discrete rather than time series. The DWIM sensors are fitted on the pavement at both the entrance and exit of each carriageway. Therefore, there are four sets of sensors for each bridge. The DWIM system can record various passing vehicle information. For example, in Table 3.2, we can observe the speed, number of axles, wheelbase and gross weight of each vehicle, as well as the lane it used and the time it passed the system. From the raw DWIM data, we can acquire the traffic volume distribution and traffic load distribution in the time domain.

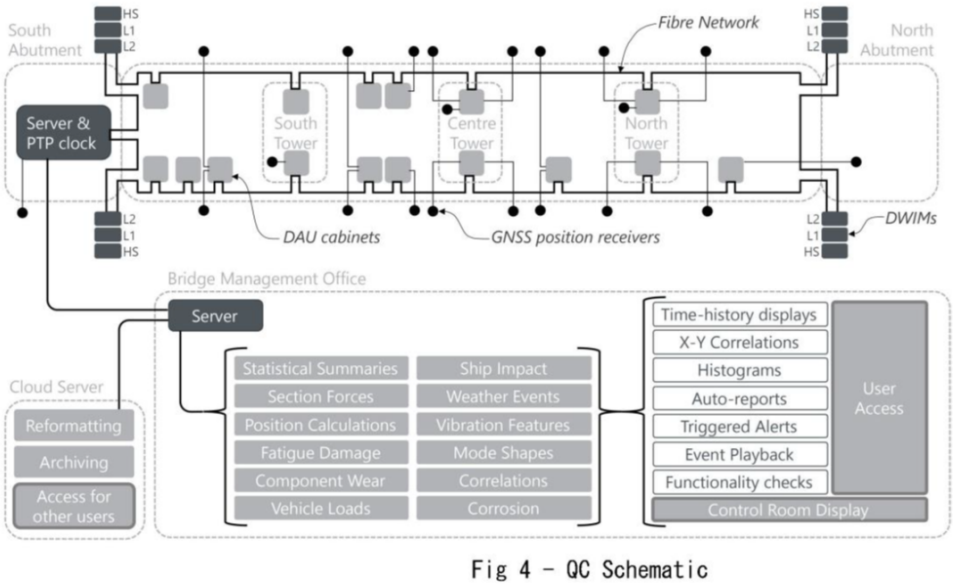


Figure 3.4: Data acquisition unit on Queensferry Crossing

Table 3.2: DWIM raw data

DATE	TimeString	LANE	LANE_DETAILS	DIRECTION	DIRECTION_NUMBER	SPEED	SPEED_MPH	CLASS_INDEX	CLASS	LENGTH	AXLES	WHEELBASE	VALIDITY	STRADDLE	OVERLOADED	GROSS
12/16/16 00:00:00	00:00:00:900	1	NORTH	SOUTHBOUND	1	69	43	1	2	550	2	290	0	0	Y	2270
12/16/16 00:00:00	00:00:09:400	4	NORTH	NORTHBOUND	2	59	37	0	1	390	2	270	64	0	Y	1850
12/16/16 00:00:00	00:00:11:400	3	NORTH	NORTHBOUND	2	60	38	33	1	570	2	290	0	0	Y	2010
12/16/16 00:00:00	00:00:11:900	1	NORTH	SOUTHBOUND	1	69	43	0	1	550	2	275	0	0	Y	1420
12/16/16 00:00:00	00:00:12:000	4	NORTH	NORTHBOUND	2	62	39	12	52	1770	4	1230	0	0	Y	36300
12/16/16 00:00:00	00:00:22:200	4	NORTH	NORTHBOUND	2	64	40	0	1	550	2	265	0	0	Y	1460
12/16/16 00:00:00	00:00:25:900	3	NORTH	NORTHBOUND	2	59	37	0	1	520	2	260	0	0	Y	1650
12/16/16 00:00:00	00:00:30:600	4	NORTH	NORTHBOUND	2	59	37	22	55	1780	5	1205	0	0	Y	31700
12/16/16 00:00:00	00:00:36:500	4	NORTH	NORTHBOUND	2	55	34	0	1	480	2	255	0	0	Y	1430
12/16/16 00:00:00	00:00:37:600	4	NORTH	NORTHBOUND	2	61	38	0	1	470	2	275	0	0	Y	1210
12/16/16 00:00:00	00:00:55:700	4	NORTH	NORTHBOUND	2	61	38	0	1	530	2	265	0	0	Y	940
12/16/16 00:00:00	00:01:07:400	4	NORTH	NORTHBOUND	2	60	38	1	2	590	2	285	0	0	Y	1540
12/16/16 00:00:00	00:01:30:300	4	NORTH	NORTHBOUND	2	68	42	0	1	540	2	265	0	0	Y	1690
12/16/16 00:00:00	00:01:35:500	4	NORTH	NORTHBOUND	2	68	42	0	1	480	2	245	0	0	Y	970
12/16/16 00:00:00	00:01:37:100	1	NORTH	SOUTHBOUND	1	59	37	33	1	560	2	285	0	0	Y	2070

3.3 Data preprocessing

As discussed in Chapter 2, data quality is important for any analysis. Both the FRB and QC data show a certain degree of abnormal data. Furthermore, from Table **??**, we can observe that data from different sensors are included in the same file and not properly sorted. Therefore, in this section, some common data preprocessing procedures used in this research are discussed.

3.3.1 Data cleaning

Missing data

Figure 3.9 shows some examples of missing data in the SHM time sequence. The causes of missing values on the FRB and QC vary, e.g. sensor system power issues. Therefore, missing data occur randomly. The scale of the missing values determines the usability of the sensor data. If only several data points are lost over a relatively long time span, the missing value can be imputed by its nearby data, e.g. linear interpolation:

$$y = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0} \tag{3.1}$$

where (x_0, y_0) and (x_1, y_1) coordinates are known and x is in the interval (x_0, x_1).

It is noteworthy that segmental data loss is not an uncommon occurrence. As illustrated in Figure 3.5, there are discernible intervals during which data are conspicuously absent. Given the complexity of the data structure, linear interpolation is insufficient for accurately imputing these missing segments. Consequently, this research proposes a machine learning-based approach for the task of segment imputation.

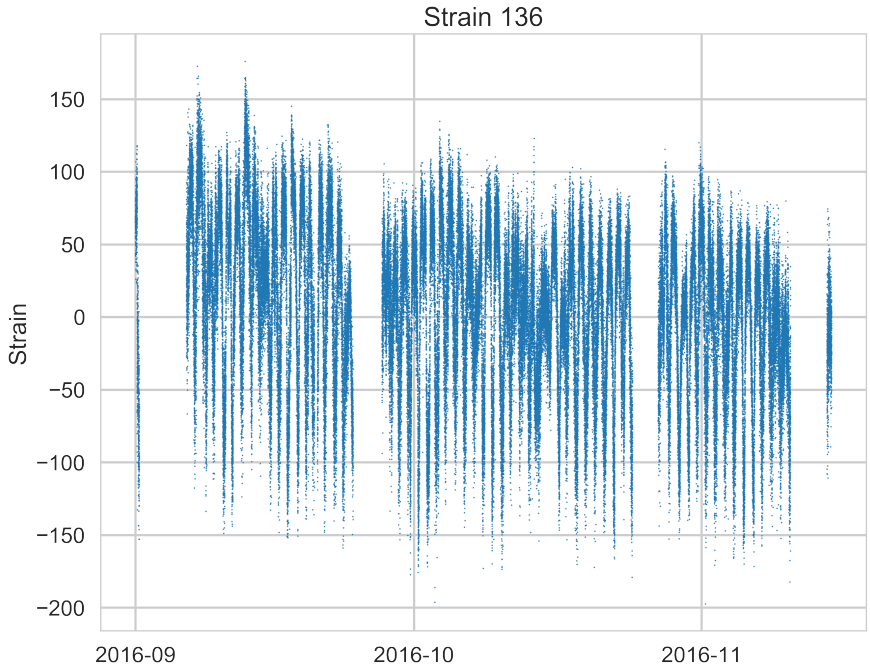


Figure 3.5: Data loss for one strain gauge on FRB

To evaluate the machine learning algorithms, one target sensor and a correlated group with 3 sensors plus traffic data are selected. Detailed information regarding the specifications and locations of these sensors is provided in Table 3.3. The sensors are installed on the FRB.

Table 3.3: Sensor specifications for missing data interpolation

	Sensor ID	Sensor Type	Location
Target Sensor	136	Dynamic Strain	Main SpanInner Link (bottom pin)
Correlated Group	60	Dynamic Strain	South West Side SpanInner Link (top pin)
	93	Displacement	South West Side SpanTruss End Post
	266	Thermometer	South East Side SpanTruss End Post

The correlation coefficients between all sensors are given in Table 3.4, where 1 represents total positive linear correlation, -1 for total negative linear correlation and 0 for no correlation. It can be discovered that correlations are generally small among sensors, ranging from -0.2 to 0.2. However, the data of displacement sensor 93 and thermometer 266 have a relatively high correlation, nearly -0.9. This can be explained by the static deformation caused by the temperature effect. Researchers have reported that a linear relationship exists between temperature and deck displacement on the Tsing Ma Bridge in Hong Kong (Y. Xu, Chen, Ng, Wong, & Chan, 2010). As a group, the correlation between the predicting group and the target sensor is higher.

Table 3.4: Correlations

	str 60	**str 136**	**disp 93**	**tmp 266**	**traffic**
str 60	1.000	0.190	0.199	0.011	-0.333
str 136	0.190	1.000	-0.119	0.054	-0.464
disp 93	0.199	-0.119	1.000	-0.899	-0.291
tmp 266	0.011	0.054	-0.899	1.000	0.214
traffic	-0.333	-0.464	-0.291	0.214	1.000

Then, a machine learning model, namely, random forest is used to predict the loss segment. Random forest works as an ensemble learning method for classification, regression or other problems. At the training stage, a random forest contains various decision trees. A decision tree is a tree-like or flow-chart like structure of decisions and represents possible consequences. In summary, the tree model determines the importance (sensitivity) ranking of the features, then uses a probabilistic approach to generate a predictive outcome. Random Forest is an ensemble learning method that uses multiple decision trees together, increasing accuracy.

The predictions of a random forest model can be written as:

$$\hat{f}_{rf} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \tag{3.2}$$

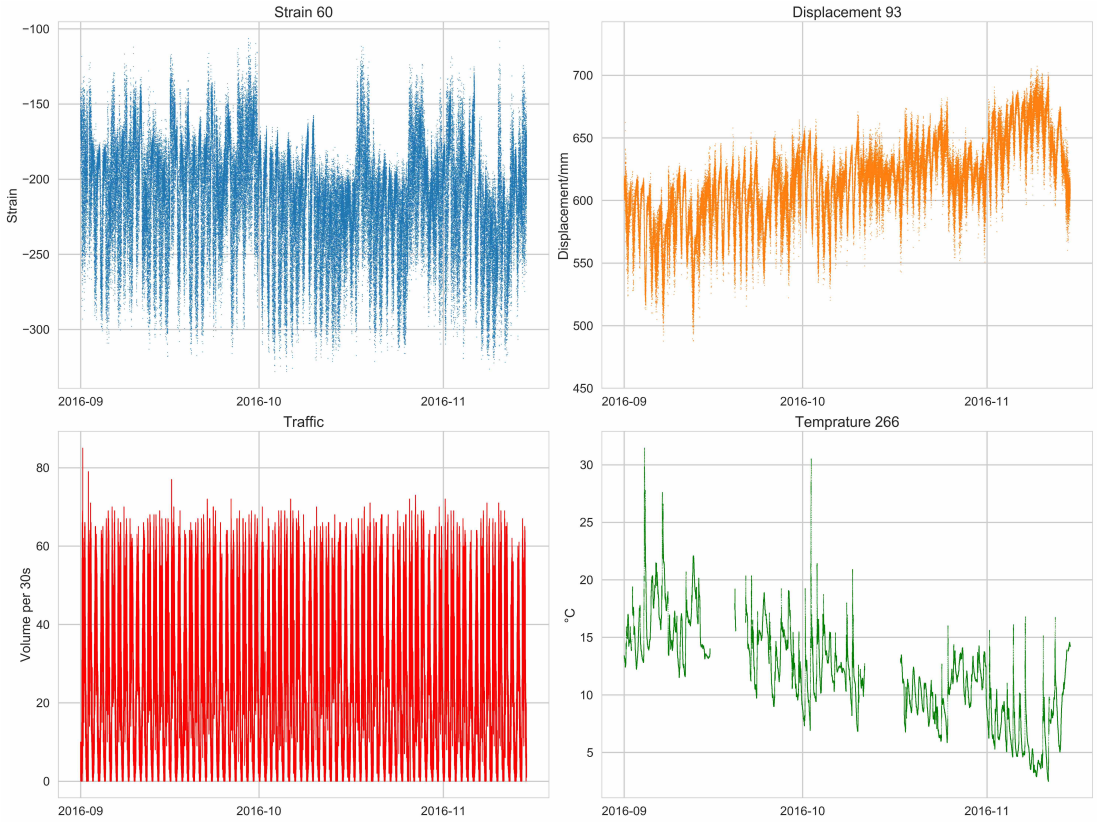


Figure 3.6: Data loss for one strain gauge on FRB

where x' is the feature vector on which the random forest trains. B is the number of trees, and $f_b(x')$ is determined by individual b-th decision tree for x' within the forest making predictions based on their learned structures. The prediction results are shown in Figure 3.8. It can be observed that the model accurately imputed the missing segment by learning correlations between the target sensor and other relevant sensors.

However, if a large segment of data is lost, such as Figure 3.9, this part of the data will not be used for data analysis or alternate sensors will be used to replace the chosen sensor. Otherwise, the imputation of a long-term missing part can introduce bias and damage the accuracy of the data. The detection of missing values for time-series SHM data is straightforward. Since the SHM for both systems has a fixed sampling rate of 1Hz for most sensors, therefore, the number of total data points for a given period

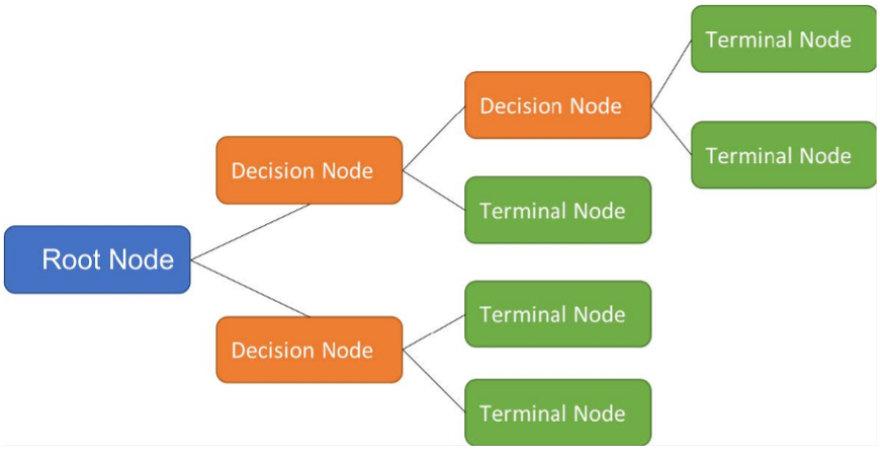
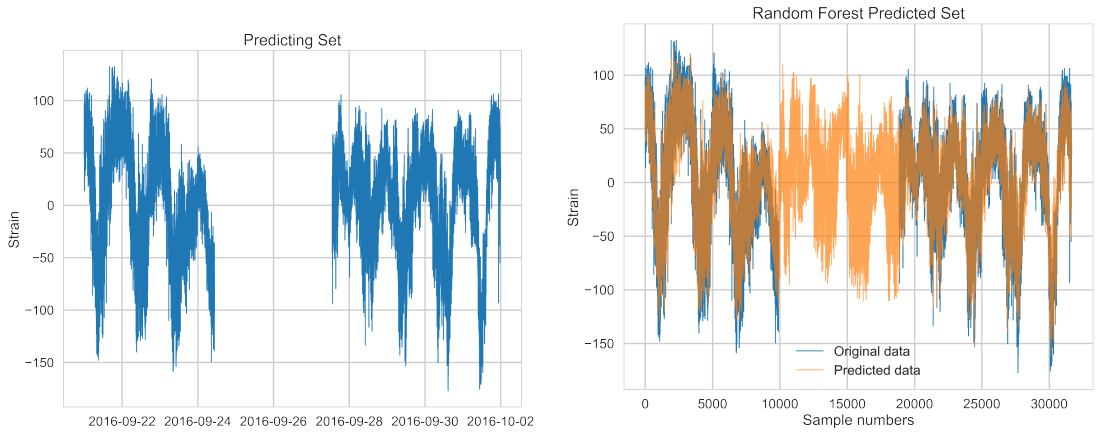


Figure 3.7: Decision Tree



(a) Prediction set **(b)** Prediction results

Figure 3.8: Random forest predictions

is known. If the actual number of data points is smaller, missing parts exist. Next, every timestamp will be added to the raw dataset for the given period, if there is a missing datum, the value for its corresponding timestamp will show as *Nan*. This can be detected and interpolated.

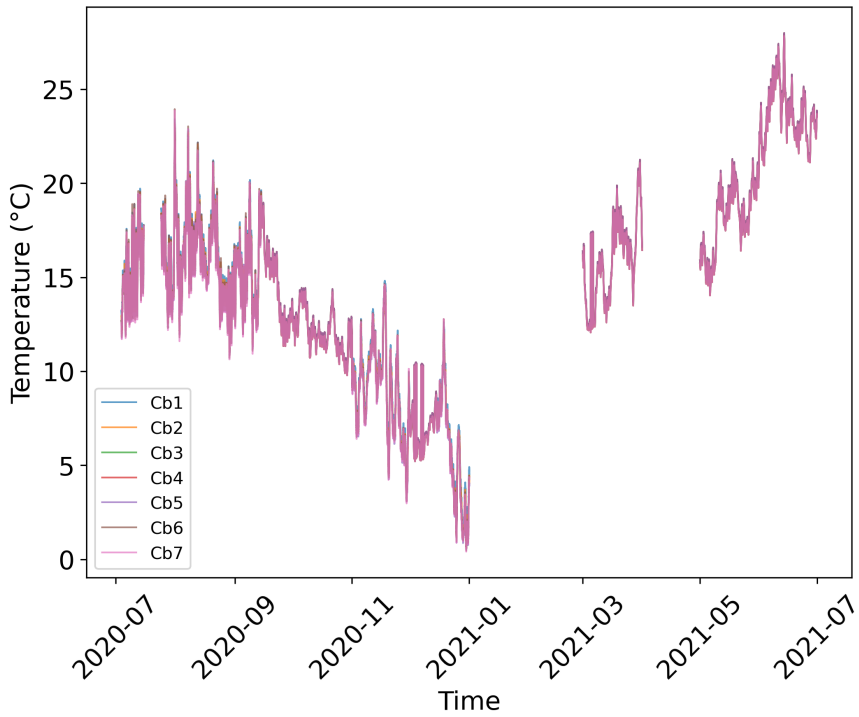
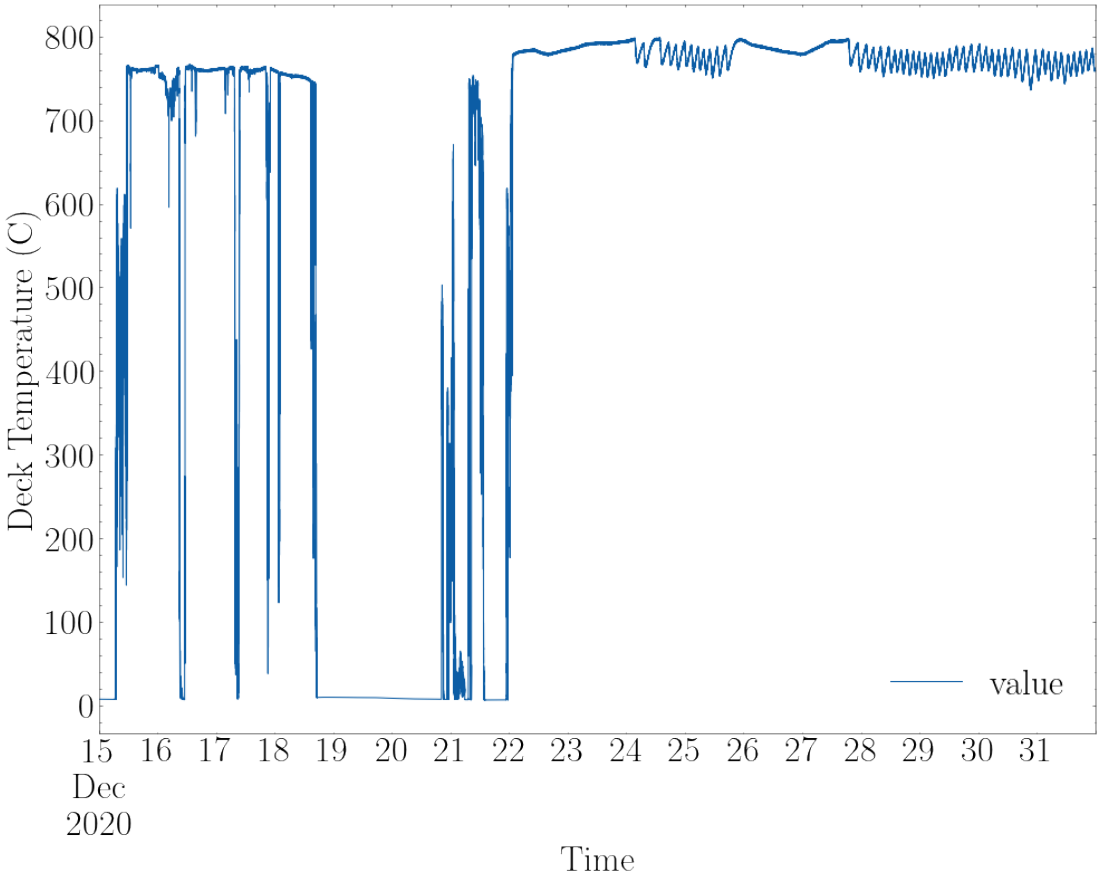


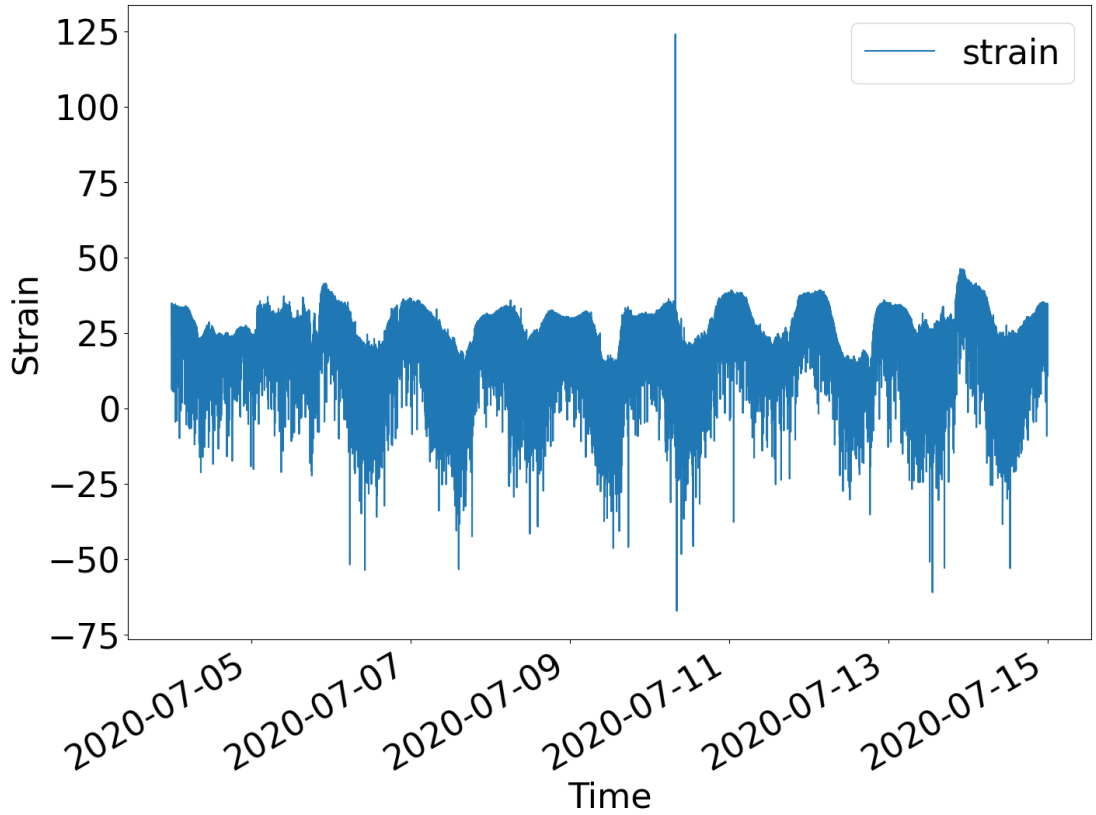
Figure 3.9: Missing data examples

Outliers

Outliers, or drifted values, are values that exceed the normal pattern of a given dataset. For an SHM system, an outlier could be caused by extreme events, e.g. a large load, and could also be caused by sensor faults, e.g. a value exceeding the sensible range of the measurements, as shown in Figure 3.10. Outliers show extreme events of great interest for researchers as an indication of potential abnormal behaviours of the bridge. Outliers caused by sensor faults are unwanted in any data analysis as they will introduce bias. Therefore, they should be deleted for a cleaner data set for further data mining. Regardless of the causes of outliers, they should be identified first before proper actions are taken.



(a) Deck temperature with sequence outliers



(b) Deck strain with point outliers

Figure 3.10: Outliers examples

The essence of any outlier detection methods is similar:

1. Create the normal pattern of the given dataset
2. Calculate the deviations between each datum and the pattern
3. Make assumptions about the normal behaviour of the data
4. Detect any datum that does not comply with the assumptions

There are two types of outliers which can be observed in the SHM data:

- **Point outliers** A datum behaves unusually in a specific time instance, for example, several unusual strains in Figure 3.10b
- **Sequence outliers** Consecutive data points in a time span are abnormal compared with the global pattern. Although, individual datum is not an outlier in that time span.

For point outliers detections in this research, z-score method is used. Z-score measures the relationship of an individual value to the mean of the dataset where the value belongs:

$$Z = \frac{x - \mu}{\sigma} \tag{3.3}$$

where x is the value of interest, μ is the mean of the dataset, σ is the standard deviation of the dataset. It can be observed from Equation 3.3 that z-score is limited to normally distributed datasets. Figure 3.11 presents the distribution analysis for the 15-day deck strain data examples. Figure 3.11(a) shows the histogram of the strain data. The distribution is narrow, indicating a small variance. The scatter plot in Figure 3.11(b) demonstrates that there are some outliers existing in the dataset. These can be seen in the whiskers in the boxplot Figure 3.11(c) and violin plot Figure 3.11(d). We can see that the deck strain data show the standard distribution for this period with several extreme values. Then the z-score for each datum is calculated and plotted. Different thresholds, 3 to 6, are set for z-scores to examine the detection of outliers on the given dataset, as shown in Figure 3.12. From this figure, we can see

that when $z = 6$, most data which are distant from the distribution can be identified as outliers without including much data in the normal pattern for this sensor. Figure 3.13 shows the results from the z-score outlier detector with the threshold set to 6. An in-depth analysis of these outliers is unnecessary for data preprocessing at the current stage. Instead, these drifted values will be removed from the dataset and interpolated.

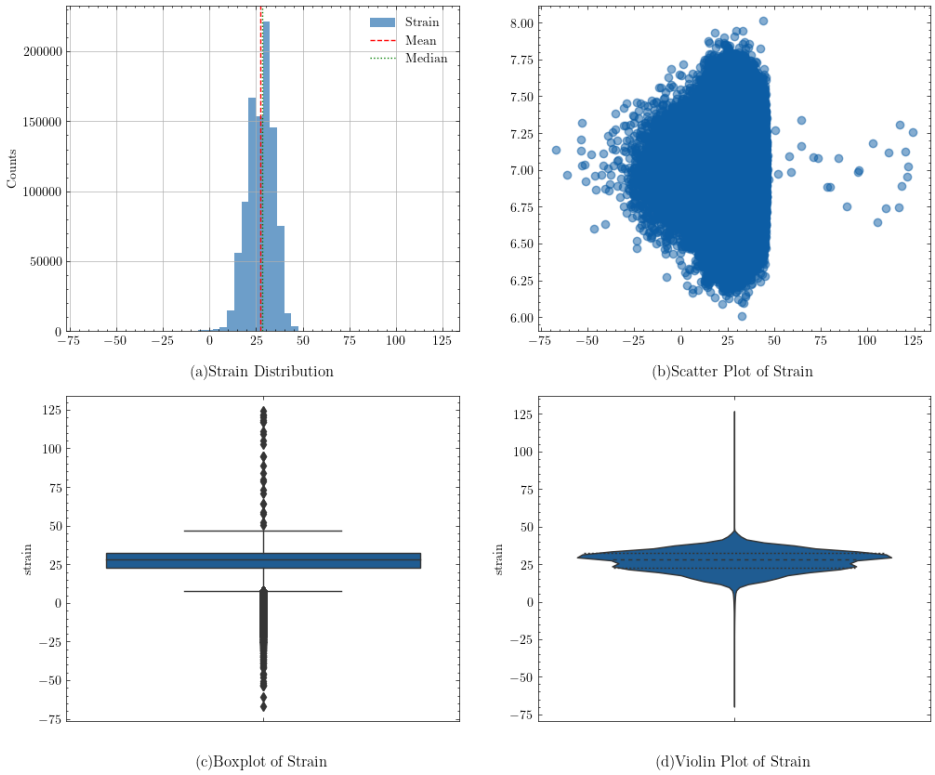


Figure 3.11: Distribution analysis

Next, the moving average is used to interpolate the outliers. From the raw data Figure 3.10b, we can see that the original strain data show seasonality since the strain is affected by temperature and load changes. Therefore, the static mean doesn't represent the variation of the dataset. Exponentially weighted moving average (EWMA) is used as interpolations for outliers in this research. EWMA can be written as:

$$\begin{aligned} \bar{x}_0 &= x_0 \\ \bar{x}_k &= \alpha \bar{x}_{k-1} + (1-\alpha) x_k \end{aligned} \tag{3.4}$$

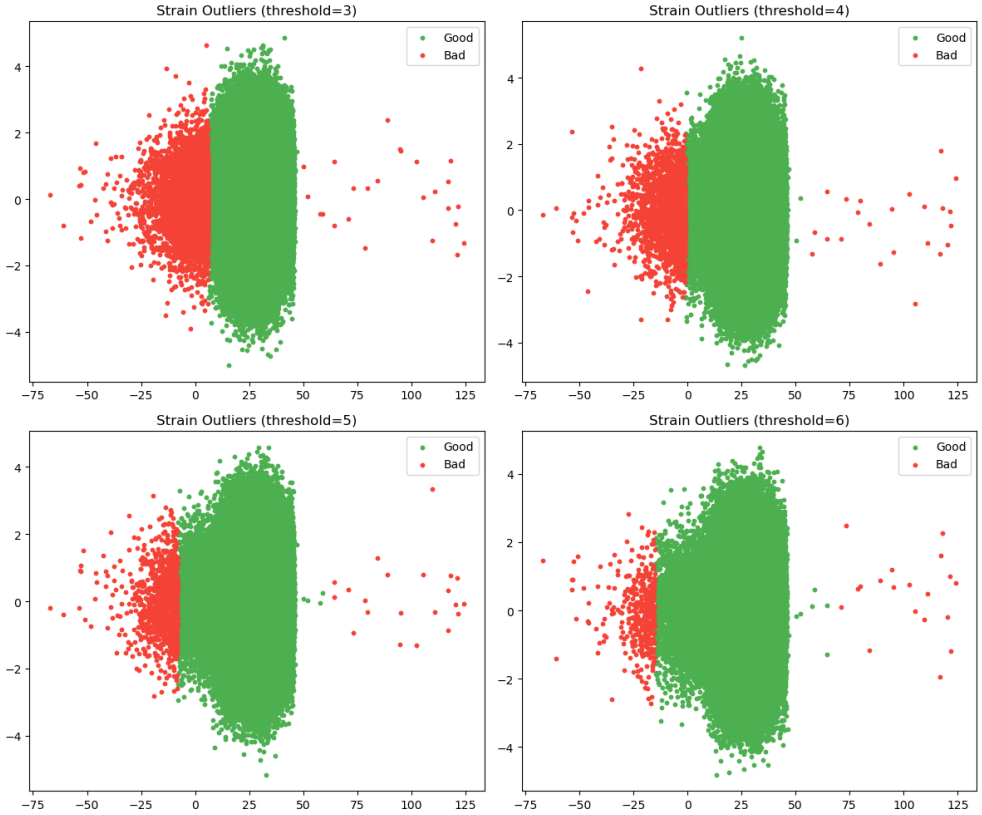


Figure 3.12: Z-Score

where $\bar{x}_k$ is the EWMA value at time k, $\bar{x}_{k-1}$ is the previous EWMA value, α dictates the degree of filtering, adjusting the influence on the current value at step k and the previous EWMA value, $0 \leqslant \alpha < 1$. Figure 3.14 presents the moving average for the 15-day deck strain data where outliers are detected with a z-score filter. In this Figure, it shows that the EWMA is a smoother curve, therefore, can be used as interpolation to the outliers detected by the z-score filter.

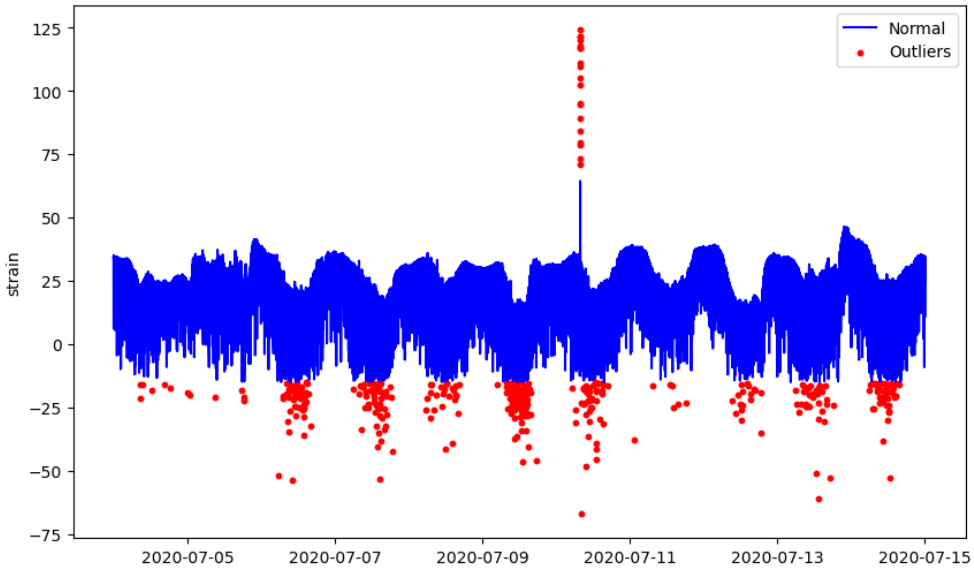


Figure 3.13: Strain outlier detections

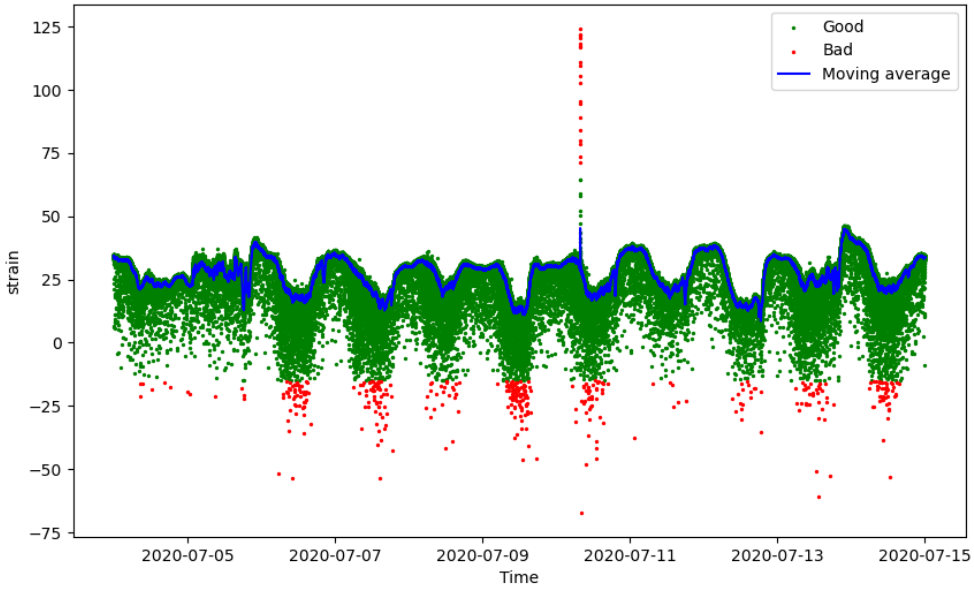


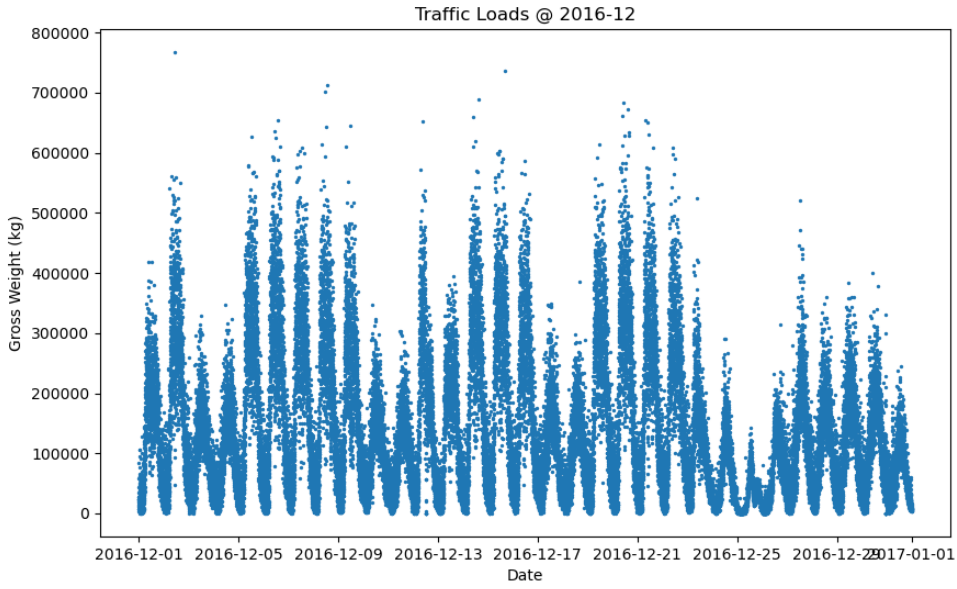
Figure 3.14: Deck strain with EWMA filter

3.3.2 Feature engineering

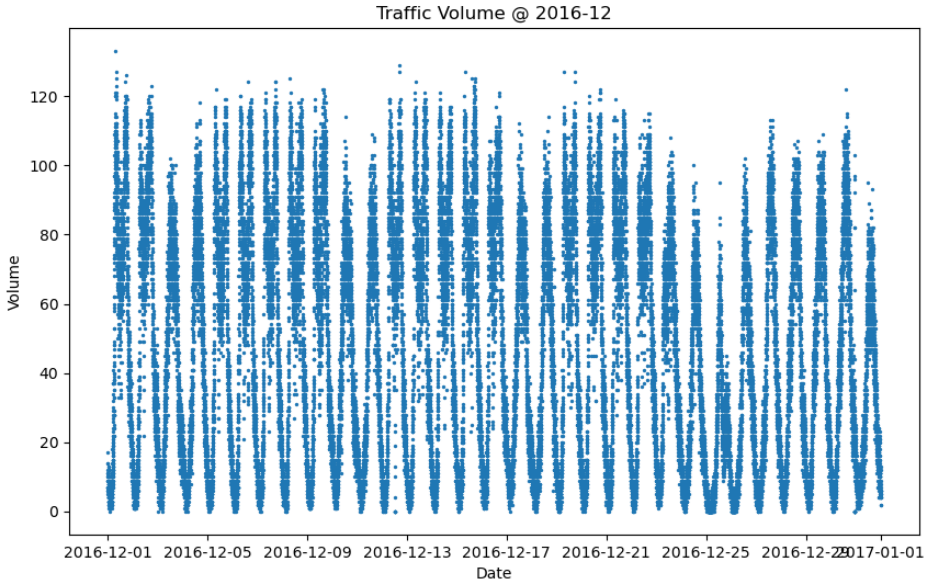
The final step in data preprocessing is feature engineering. A combined multivariate dataset, which includes input and output features, is created for the deep learning model. Given that bridge SHM data primarily consist of time-series data, they remain unlabelled. First, a label must be created for supervised learning models. This research chooses structural responses, such as strain and deflections, as the label or output feature. In some scenarios, lag features are introduced to investigate the models' ability for future forecasting. For instance, the forecasting model is set to predict the SHM data in the $n + k$ step based on the n step.

Data derived from the DWIM system must be converted to a time-series format for deep learning models to align with the format of structural response data. In this research, traffic volume and general gross weight over time are used to construct the multivariate dataset. Figure 3.15 shows examples of the traffic volume and general gross weight from the Forth Road Bridge in December 2016. The general gross weight depicted in Figure 3.15a represents the aggregate weight of each passing vehicle, contingent upon the sampling rate, which in this instance is one minute.

At this stage, the raw data from the bridge SHM system have been selected, cleaned and reconstructed. The final step in data preprocessing is normalisation. As elaborated in Chapter 2, different sensors have different natural ranges. Therefore, all features in the training dataset must be scaled to a unified range to ensure all features contribute equally to the model and prevent features with a more extensive range from dominating the training process. Table 3.5 provides several examples of concatenated datasets for assessing thermally-induced structural responses. Additionally, Figure 3.16 illustrates the normalised concatenated dataset within the range of $[0, 1]$.



(a) FRB traffic gross weight December 2016



(b) FRB traffic volume December 2016

Figure 3.15: Traffic data reconstruction

Table 3.5: Preprocessed dataset example

timestamp	temperature1	temperature2	temperature3	temperature4	temperature5	GPS	strain
2020-09-01 00:00:00	15.24105	15.99695	15.75862	15.19273	13.89298	9.101194	22.09501
2020-09-01 00:00:01	15.24010	15.99680	15.75736	15.19163	13.89313	8.101194	22.26755
2020-09-01 00:00:02	15.24073	15.99727	15.75705	15.19194	13.89250	5.601194	22.20468
2020-09-01 00:00:03	15.23995	15.99774	15.75815	15.19116	13.89219	-4.298806	22.88679
2020-09-01 00:00:04	15.24057	15.99695	15.75658	15.19210	13.89235	-6.398806	23.21715
2020-09-01 00:00:05	15.23979	15.99664	15.75783	15.19194	13.89266	-13.198806	25.39455
2020-09-01 00:00:06	15.24026	15.99790	15.75767	15.19241	13.89282	-13.298806	23.54884
2020-09-01 00:00:07	15.23932	15.99648	15.75767	15.19194	13.89329	-5.698806	23.48732
2020-09-01 00:00:08	15.23901	15.99633	15.75736	15.19163	13.89219	-6.898806	24.47705
2020-09-01 00:00:09	15.23916	15.99680	15.75736	15.19069	13.89345	0.801194	24.19217
2020-09-01 00:00:10	15.23932	15.99680	15.75767	15.19179	13.89298	1.601194	23.31211
2020-09-01 00:00:11	15.23995	15.99601	15.75673	15.19163	13.89298	5.701194	22.46817
2020-09-01 00:00:12	15.23932	15.99476	15.75720	15.19053	13.89219	15.401194	22.39728
2020-09-01 00:00:13	15.23885	15.99711	15.75752	15.19116	13.89266	16.001194	21.68441
2020-09-01 00:00:14	15.23963	15.99601	15.75767	15.19147	13.89282	14.801194	21.69110

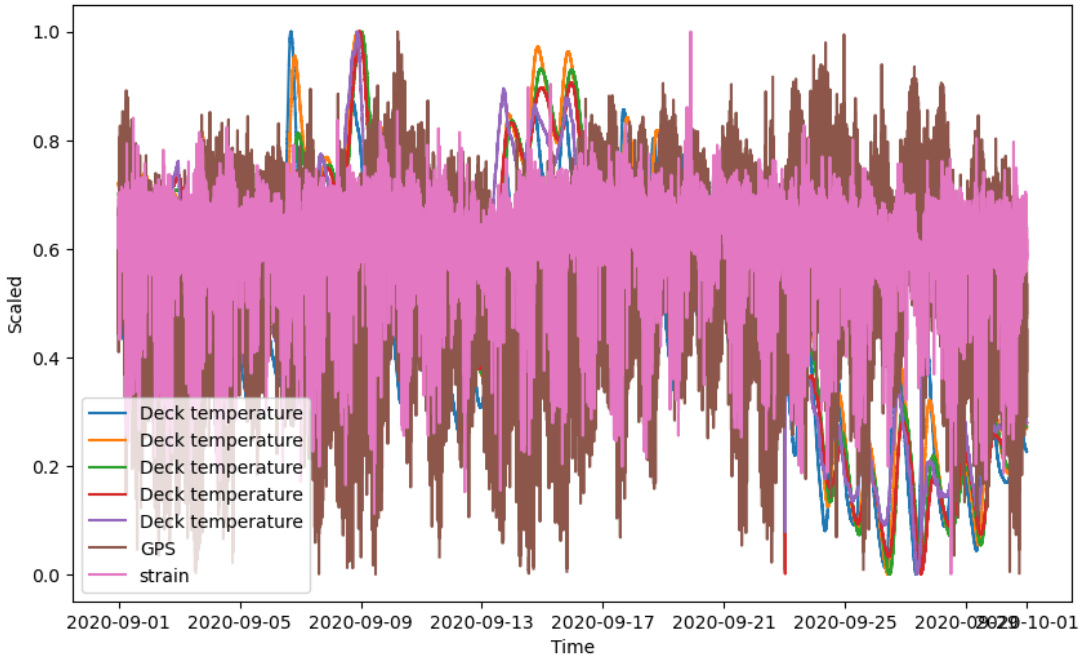


Figure 3.16: Normalised dataset

3.4 Summary

This chapter presented the data preprocessing techniques employed in this study. Both the Forth Road Bridge and the Queensferry Crossing are extensively monitored by a range of sensors, providing a vast amount of data from the past several years. Nevertheless, the harsh operating conditions of SHM systems on long-span bridges result in data containing various errors, such as missing data points and outliers. Furthermore, different analyses necessitate distinct datasets, thereby requiring careful sensor selections.

The data preprocessing workflow can be summarised in Figure 3.17. Initially, appropriate sensors are selected based on the research problem to construct a dataset. Subsequently, the quality of raw data is assessed. Missing data points are imputed via linear interpolation. Random Forest can interpolate shorter segment missing data based on data from correlated sensors. Thereafter, outliers are identified using a z-score filter and subsequently replaced with moving average values. As needed, specific data types, such as traffic loads and volume, are converted to time series format. The cleaned sensor data, referred to as features, are then concatenated into a single dataset, and the output feature is established, e.g., strain or deflections. Ultimately, the dataset is normalised to a unified range, e.g. $[0, 1]$.

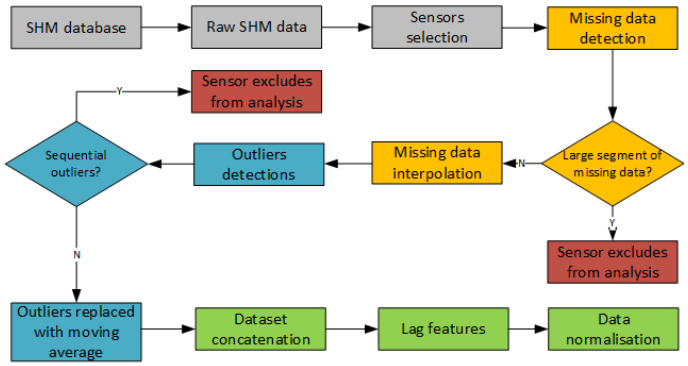


Figure 3.17: Data preprocessing work flow

Chapter 4

Traffic Load Simulations

4.1 Backgrounds

In this chapter, the vehicle-induced structural responses are investigated. As discussed in Chapter 2, both environmental and operational loads influence the bridge responses. Therefore, the traffic effect is a primary factor in measured structural responses. In this research, where two road bridges are involved, the vehicle-induced structural responses contain damage signals, thereby increasing the difficulties of anomaly detection or condition assessment. To enhance the efficacy of anomaly detection or condition assessment, it is crucial to investigate methodologies for modelling vehicle-induced responses.

One distinct feature of traffic loads is the high randomness in spatiotemporal distributions, even though the general trend presents a high seasonality, as shown in Figure 3.15b, the individual vehicle attributes are highly random (e.g., type, speed, gross weight, number of axles). Currently, vehicle information is monitored by the DWIM systems which are normally fitted at the entrances and exits of the bridge. Therefore, the spatial distribution of the traffic across the bridge is unavailable to researchers. Thanks to the advancement in sensing technology and image processing, there are several applications for combining DWIM measurements and footage from the on-site CCTV to reconstruct the spatiotemporal distributions of vehicle loads on bridges (Z. Chen, Li, Bao, Li, & Jin, 2016; Y. Pan, Wang, Dong, & Peng, 2021). As vision-

based methods are significantly influenced by image quality, the effectiveness of these methodologies may become unacceptable during nighttime, foggy weather, or when others obscure the target vehicle. Furthermore, image processing can be both hardware and time-intensive. Matarazzo et al. (2022) proposed using smartphone-vehicle trip data to determine the modal frequencies of bridges. In this research, GPS data were provided from the smartphones. Therefore, it is possible that the spatial information of the passing vehicle can be provided by the passengers' smartphones. However, this method was only implemented under the experimental settings. Therefore, the long-term application is questionable due to privacy concerns.

The in-situ measurements for both traffic and structural responses occasionally generate data of poor quality. Therefore, simulations can be used to generate high-quality samples for further analysis.

In this Chapter, traffic load simulations are performed based on the bridge DWIM datasets. Simulation models based on historical vehicle attributes data are developed. The results are compared with the actual deflections from the bridge SHM system.

4.2 Methodology

Figure 4.1 shows the general flowchart of the proposed strategy of estimating traffic-induced midspan deflections based on simulation techniques. In the simulation, the traffic volume and simulation time window are pre-defined to compute the total number of simulated vehicles. Next, the Poisson process is employed to determine the arrival instant of each vehicle, generating a random traffic sequence. Then, attributes (e.g., speed, gross weight) are assigned to vehicles based on probabilistic models, creating an initial vehicle load sequence at the bridge entrances. This vehicle sequence with attribute information corresponds to the DWIM data measured at the

bridge entrances in reality. Subsequently, utilising the agent-based micro-simulation (i.e., car following and lane changing models), vehicle behaviours (e.g., acceleration) are simulated on the bridge to derive their spatiotemporal distributions of vehicle loads. Additionally, based on FE models, the influence line can be calculated by defining a unit load on the bridge. Finally, using the vehicle loads and locations from the simulation and deck displacement influence line under a unit load, the midspan deflection can be calculated by simple multiplications.

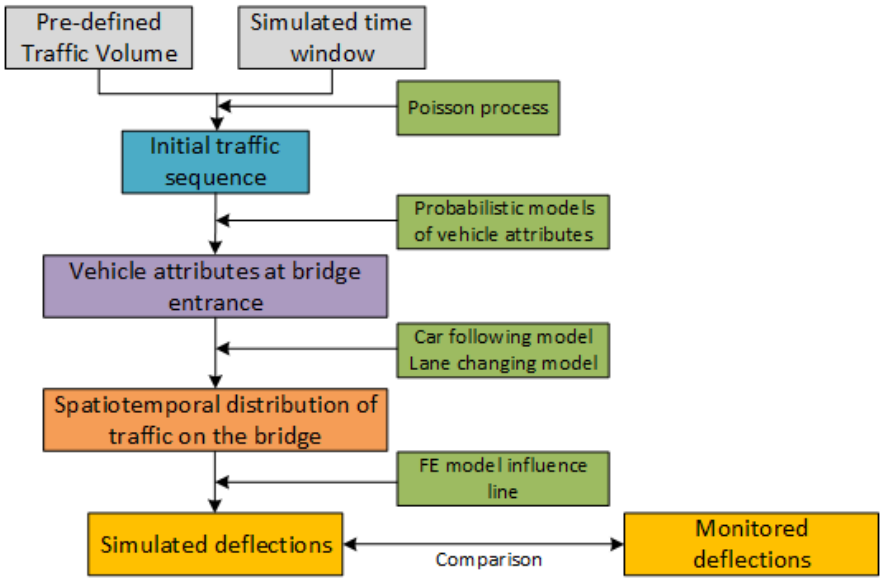


Figure 4.1: Queensferry Crossing FE model

4.2.1 Vehicle attributes generation

For simulating stochastic traffic flows, the initial step involves determining the number of vehicles in the simulation based on the traffic volumes and the chosen time windows. Following this, the arrival time of each vehicle is generated randomly to establish a traffic sequence. Generally, the time intervals between successive vehicles are considered as a variable conforming to a specific distribution. Under normal traffic conditions, the vehicle arrival process is modelled using the Poisson process, signifying that the intervals between two successive vehicles adhere to the probability density

function of an exponential distribution as:

$$f(\Delta T|\lambda) = \lambda e^{-\lambda \Delta T} \tag{4.1}$$

where ΔT is the time interval, and λ is the rate parameter. The parameter λ indicates the total number of arrived vehicles within a unit time. Once the daily mean traffic volume Q is determined, the rate parameter is calculated by using the following equation:

$$\lambda = \frac{Q}{3600 \times 24} \tag{4.2}$$

Besides the vehicle arrival time, various other vehicle attributes, such as vehicle type, speed, and gross vehicle weight, are crucial for generating a vehicle load sequence. The vehicle type is initially created based on historical DWIM records. Once the vehicle type is determined, the attributes are assigned accordingly, utilising their corresponding probabilistic models associated with the specified vehicle type.

4.2.2 Car following model

Vehicle behaviours on the main bridge are represented with the agent-based micro-simulation, which encompasses car following and lane changing models. This approach not only characterises individual vehicle movements but also captures drivers' behaviours in accordance with established rules and regulations. To account for the variations among different types of vehicles, distinct agents are devised based on their unique parameters, such as maximum vehicle speed.

The car following model encompasses acceleration, deceleration, and collision avoidance sequences. The car following logic is influenced by the space gap S_{gap} and the speeds of both the leading, V_L, and following vehicles, V_F. In a car following scenario, a follower aims to maintain a desired space gap S_{de} behind the leader, which is

calculated as the product of the follower's speed and preferred time headway:

$$S_{de} = V_f \cdot TP \tag{4.3}$$

As outlined in Table 4.1, vehicle behaviours are determined based on their speed and gap conditions when travelling across the main bridge (Bham & Benekohal, 2004).

Table 4.1: Actions taken by the following vehicle (Bham & Benekohal, 2004)

	Speeds		
Gaps	$V_L > V_F$	$V_L = V_F$	$V_L < V_F$
$S_{gap} > 3V_F$	Accelerate	Accelerate	Accelerate
$3V_F \geqslant S_{gap} > 2V_F$	Accelerate	Accelerate	Coast
$2V_F \geqslant S_{gap} > 2V_F$	Accelerate	Accelerate	Decelerate
$S_{gap} = S_{de}$	Coast	Coast	Decelerate
$S_{gap} < S_{de}$	Coast	Decelerate	Decelerate

Subsequently, the acceleration and deceleration rates must be established. Taking into account that the vehicle is progressively accelerating through the gears, the acceleration rate of the follower can be described as follows:

$$a_{ac} = \begin{cases} 1.1 m/s^2 & \text{if } 0 \leq v < 12.19 m/s \\ 0.37 m/s^2 & \text{if } v > 12.19 m/s \end{cases} \tag{4.4}$$

The motivation for deceleration is to increase the desired space gap for the follower, or to prevent collisions. Therefore, the deceleration rate will be calculated in three scenarios.

In the context of free-flowing traffic, when the speed of the follower exceeds its desired level, the driver decelerates at a specified deceleration rate as follows:

$$a_{de} = \min(\frac{v - v_{de}}{t}, a_0) \tag{4.5}$$

where v_{de} is the desired speed of the given vehicle type, t is a constant and given as 3, and a_0 is the normal deceleration rate, which equals to $3.05m/s^2$.

In instances where stopping behind the leader with a buffer space is necessary, the deceleration rate must exceed a predetermined target value, which is given by:

$$a_{de} = \frac{V_F^2 - v_L^2}{2S_0} \tag{4.6}$$

where S_0 is the gap value by subtracting the buffer space from the space gap.

In scenarios involving collision avoidance, with the assumption that the deceleration rate of the leading vehicle remains constant, the minimum deceleration rate required for the follower is:

$$a_{de} = \begin{cases} \min(a_L + \frac{V_F^2 - v_L^2}{2S_{gap}}, a_{max}) & \text{if } V_F > V_L \\ a_L - 0.25a_0 & \text{if } V_F \leq V_L \end{cases} \tag{4.7}$$

As mentioned earlier, the deceleration rate is determined according to the three defined cases. During the simulation process, the appropriate case is identified based on the driving conditions of the target vehicle before calculating the deceleration rate.

4.2.3 Lane changing model

In addition to the car following model, lane changing plays a crucial role in traffic flow simulation. A driver tends to change lanes if a slow car exists in the front and there is sufficient space in front of and behind the car. Therefore, two criteria, namely incentive and safety, are considered in the lane changing model presented here (F.-Y. Wang

& Xu, 2019). Specific lane changing cases are defined in Table 4.2 by Kurata and Nagatani (2003). When both incentive and safety criteria are met, the target vehicle is expected to change to the adjacent lane within the simulation program. The lane changing model is depicted in Figure 4.2.

Table 4.2: Lane-changing rules (Kurata & Nagatani, 2003)

Scenarios	Incentive criterion	Security criterion
1	$v_i > 1.02V_{i+1}$, and $\Delta x_i < 4x_C$	$\Delta x_{fi} > 2x_C$, and $\Delta x_{bi} > x_C$
2	$v_i > 1.02v_{i+1}$, and $\Delta x_i < 4x_C$	$v_j > v_i$, $\Delta x_{fi} > x_C$, and $\Delta x_{bi} > x_C$
3	$\Delta x_i < 2x_C$	$\Delta x_{fi} > x_C$, and $\Delta x_{bi} > x_C$

v_i is the speed of the target car; v_{i+1} is the speed of the leading car in the same lane; Δx_i is the gap between the target and leading cars; x_C is the safety distance; Δx_{fi} is the headway between the target car and the front car in the adjacent lane; and Δx_{bi} is the headway between the target car and the back car in the adjacent lane.

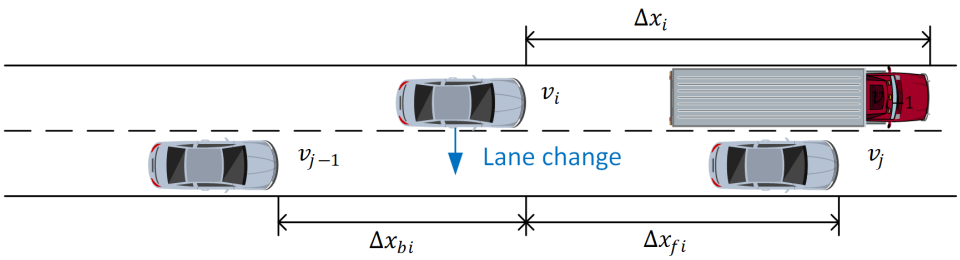


Figure 4.2: Lane-changing model

In the UK, the Highway Code specifies that vehicles should keep to the left lane whenever possible. On the other hand, both the Queensferry Crossing and the Forth Road Bridge are dual two-lane carriageways, excluding the hard shoulders and pedestrians. Therefore, the lane-changing model in this study considers only one scenario, a vehicle changes lanes from the left lane to the right lane. After the lane changing, the vehicle will follow the car following model.

4.2.4 Vehicle-induced structural response simulation

The next step of the traffic load estimation model is to determine the vehicle-induced structural deflections based on the simulated traffic flow. At a specific time t, the spatial distribution of vehicles is determined based on the spatiotemporal distributions of the vehicles. Taking into account the spatial distribution of vehicles, the ones present on the main bridge are selected, as they contribute to the vehicle-induced structural response at instant t.

Considering the weights and locations of vehicles on the main bridge, the most direct approach to calculate the structural response is to utilise a finite element (FE) model. In order to guarantee the reliability of the FE model, it is necessary to perform model updating based on load testing data. To circumvent complex software interactions, the influence line of a given structural response is employed in this study for computation purposes. An influence line graph depicts the variation of deflections, shear force or bending moment at specific nodes on a structure when a unit load is applied. In this research, the Queensferry Crossing FE model developed by ARUP, as shown in Figure 4.3, is used to derive the deflections influence line. There was a total of 1793 nodes and 1405 elements in the entire model. The decks, towers and peers were defined as beam elements while the cables are tension-only elements in MIDAS. As for materials, C55 concrete was used. The steel for cables is specified under standard EN05(S) with the elasticity being $1.95 \times 10^2 kN/mm^2$. The steel for reinforcement is defined with the elasticity being $2.10 \times 10^2 kN/mm^2$. The deck is fixed connected to piers and 3 towers.

As mentioned earlier, the structural response at time t is obtained. By following the same procedure, a structural response sequence is generated with a frequency of f, where the value of f can be determined according to the research objectives.

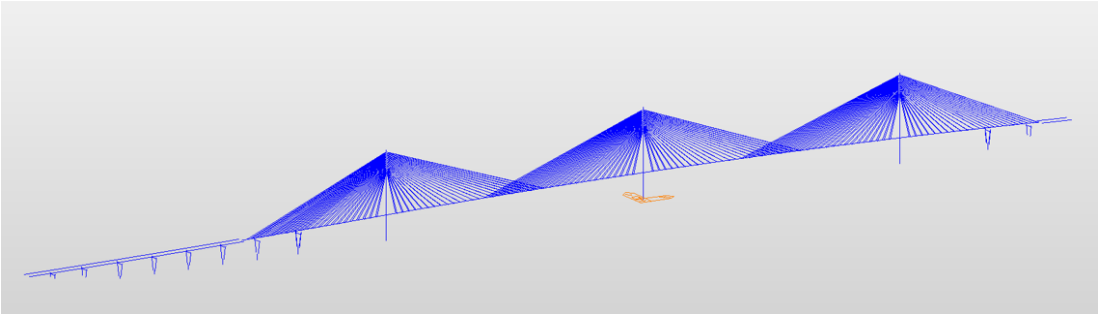


Figure 4.3: Queensferry Crossing FE model

To confirm the efficacy of the simulation program, the simulated structural response is compared with the observed one. If both the simulated and measured signals demonstrate fluctuations at nearly the same level, it implies that the simulated traffic load is consistent with the actual one.

4.3 Case study

In this section, the simulation strategy described in the previous section is applied to the traffic data from the Queensferry Crossing. The DWIM system on the QC consists of 12 sensors at each entrance and exit of the two carriageways, with one sensor on each lane. Figure 4.4 shows the DWIM sensors layout. Based on the report published by the Scottish Government, the average daily traffic volume is 80,000 for the QC, as shown in Figure 4.5, which is later used to determine the traffic sequence according to the methodology. Figure 4.6 plots the average daily traffic variation based on Figure 4.5. The average daily traffic shows different patterns for weekdays and weekends. The weekday traffic volume has two peaks around the morning and evening rush hours, while the weekend traffic volume has one peak around the early afternoon.

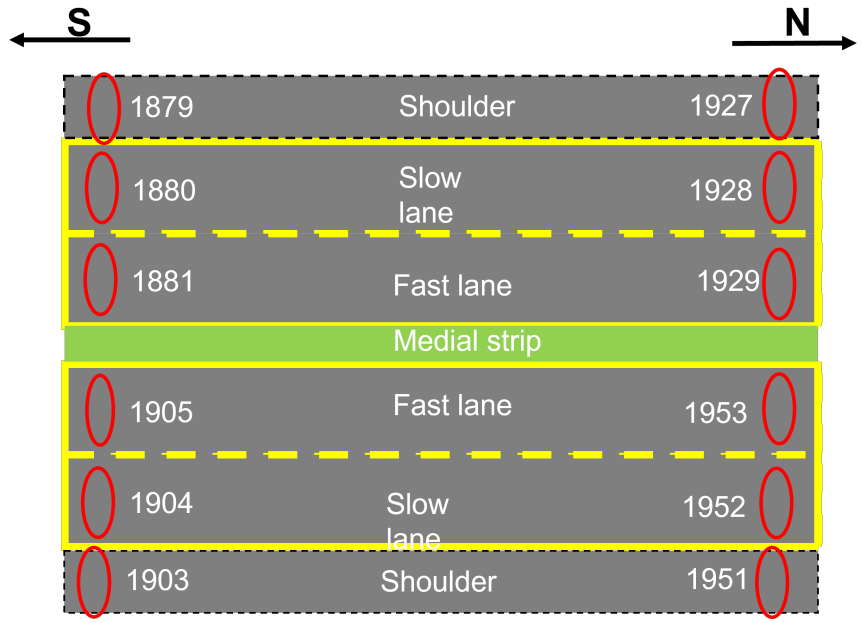
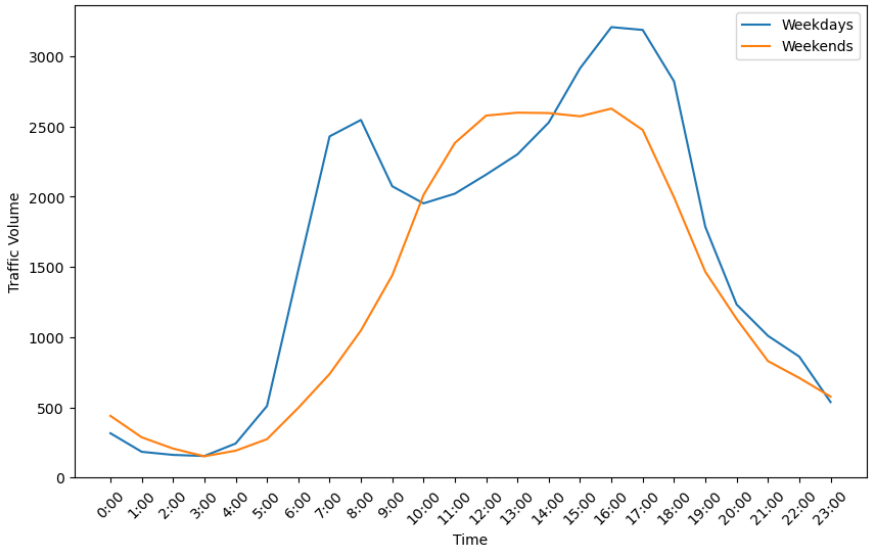
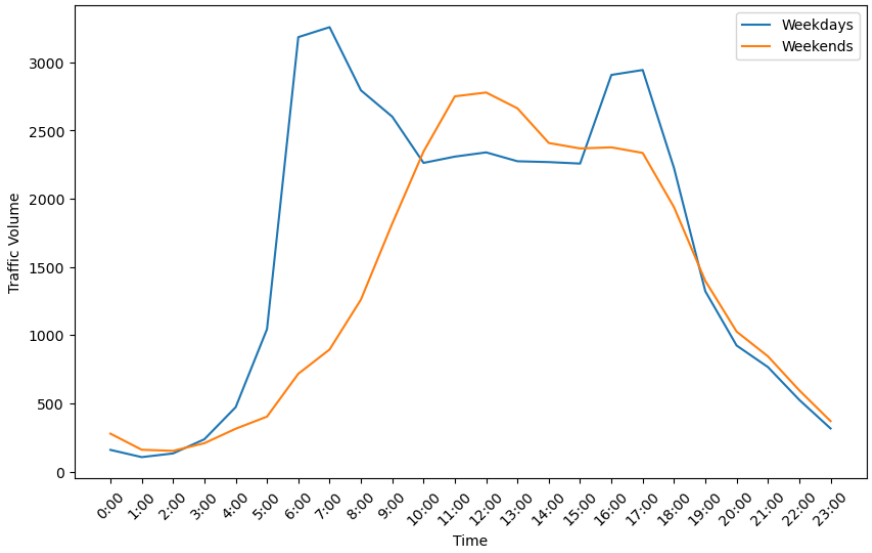


Figure 4.4: Queensferry DWIM sensors



(a) Northbound average daily traffic



(b) Southbound average daily traffic

Figure 4.6: Average daily traffic on QC

Hourly Average Traffic Flow by Month (vehicles)

Direction	Month (2019)	Time of Day																							
		00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
Northbound	March Weekday	278	154	145	148	234	505	1496	2490	2563	2050	1835	1875	2068	2227	2492	2867	3206	3159	2661	1646	1219	947	820	517
	March Weekend	410	222	178	151	174	248	475	694	998	1360	1841	2275	2534	2520	2528	2500	2592	2478	1896	1380	1112	783	715	539
	April Weekday	313	187	152	141	235	494	1456	2360	2461	2059	2047	2139	2227	2350	2535	2963	3230	3253	2915	1757	1252	1008	858	534
	April Weekend	430	295	216	139	183	273	495	732	1032	1436	2166	2488	2626	2676	2655	2621	2644	2436	1957	1496	1142	837	708	589
	May Weekday	357	210	189	172	260	531	1483	2439	2618	2117	1976	2052	2178	2328	2559	2914	3188	3152	2890	1956	1228	1074	909	562
	May Weekend	479	343	227	167	218	300	520	789	1114	1524	2032	2385	2575	2601	2604	2597	2649	2515	2136	1522	1136	870	706	604
Southbound	March Weekday	138	99	124	223	469	1064	3254	3390	2785	2593	2151	2169	2211	2153	2162	2535	2882	2891	2110	1228	828	699	484	287
	March Weekend	245	139	150	202	289	376	731	889	1255	1803	2305	2762	2745	2609	2326	2252	2253	2185	1793	1242	918	787	530	324
	April Weekday	162	97	130	227	446	1009	3112	3134	2758	2544	2352	2411	2417	2344	2332	2623	2926	2970	2279	1370	963	773	513	307
	April Weekend	272	162	145	194	305	393	656	819	1184	1793	2329	2758	2801	2675	2444	2455	2521	2500	2096	1542	1135	906	648	373
	May Weekday	177	123	146	262	502	1059	3188	3250	2841	2670	2287	2348	2392	2328	2314	2617	2917	2972	2286	1367	981	826	580	354
	May Weekend	309	180	160	228	348	441	764	976	1343	1858	2403	2733	2791	2706	2456	2401	2358	2324	1926	1409	1024	843	610	413

Notes:
Data extracted from Transport Scotland National Traffic Data System on 4 June 2019.
Counter references - NB NTC01309, SB NTC01310.
Data split by weekday/weekend to allow interpretation of 'peak' hours to be extracted.

Figure 4.5: Queensferry hourly traffic by Transfport Scotland

4.3.1 Traffic flow simulation

Traffic patterns on the Queensferry Crossing

The DWIM system on the QC provides continuous vehicle monitoring data. However, the sensors only started to provide stable data feeds in 2021. 5-month traffic data are used to provide the traffic patterns on the QC for simulations. Table 4.3 shows examples of DWIM system records derived from the raw dataset.

Vehicle attributes, e.g. gross weight, speed and class, are vital for generating a vehicle load sequence. Different features are extracted from the DWIM records. Table 4.4 shows the proportion of each vehicle class in the four entrance lanes, 1880, 1881, 1952 and 1953. Data from the hard shoulders are excluded. The DWIM system recorded a total of 919,222 vehicles passing the bridge in both directions in a 5-month period, which is significantly less than the actual traffic volume, 80,000 per day. This is due to a known issue in the Queensferry Crossing DWIM system: not all passing vehicles can be registered. Therefore, the total number of traffic recorded by the DWIM is not used in the microsimulation. However, since the records are random, the vehicle attributes derived from the DWIM system can be used to represent the actual situations.

Table 4.3: QC DWIM data examples

timestamp	sensor_id	axles	length	speed_kph	class	gross	headway
10/02/2021 23:00	1880	2	450	101	1	1010	0
10/02/2021 23:02	1880	2	490	96	1	1360	3900
10/02/2021 23:02	1880	2	690	105	31	2880	0
10/02/2021 23:04	1880	2	1300	99	31	10500	0
10/02/2021 23:05	1880	2	500	104	2	2360	0
10/02/2021 23:05	1880	2	440	100	1	1360	0
10/02/2021 23:08	1880	2	520	116	2	1920	0
10/02/2021 23:08	1880	2	480	111	1	1450	0
10/02/2021 23:09	1880	2	430	81	1	1000	0
10/02/2021 23:11	1880	2	430	100	1	810	0
10/02/2021 23:11	1880	2	450	102	1	800	0
10/02/2021 23:11	1880	5	1730	87	55	16300	0
10/02/2021 23:11	1880	4	1690	84	52	11400	3850
10/02/2021 23:16	1880	6	1740	85	56	38400	0
10/02/2021 23:17	1880	2	550	105	2	1770	0

Based on the observations in Table 4.4, the vehicle distributions in both directions are similar. Therefore, to simplify the simulation, we assume that the traffic volume and distributions in each direction are identical. Therefore, within the single direction, the proportions of vehicles in the left line and right line are 67% and 33%, and the volume is 40,000 per day.

Vehicle attributes

Once the traffic volume and vehicle distributions are determined, vehicle types need to be decided. From the records in Table 4.4, vehicles are classified into 11 types based on the number of axles. For a given vehicle type, its attributes (e.g. gross weight, speed) are further analysed. Based on the DWIM data, the probabilistic distributions of these attributes are modelled. Then, the parameters of these probabilistic models are used to generate traffic flows in each lane.

Table 4.4: Number of each type of vehicles in the four entrance lanes

	Lanes								
Axles	1880		1881		1952		1953		
2	242396	83.751%	146746	98.512%	280634	87.018%	156166	98.631%	
3	4876	1.685%	414	0.278%	4152	1.287%	384	0.243%	
4	9458	3.268%	554	0.372%	8926	2.768%	528	0.333%	
5	18754	6.480%	674	0.452%	16496	5.115%	664	0.419%	
6	13804	4.769%	570	0.383%	12134	3.762%	588	0.371%	
7	102	0.035%	4	0.003%	118	0.037%	4	0.003%	
8	14	0.005%	0	0.000%	30	0.009%	0	0.000%	
9	14	0.005%	0	0.000%	4	0.001%	0	0.000%	
10	6	0.002%	0	0.000%	4	0.001%	0	0.000%	
11	0	0.000%	0	0.000%	0	0.000%	0	0.000%	
12	0	0.000%	0	0.000%	4	0.001%	0	0.000%	Sum
Sum	289424	100.000%	148962	100.000%	322502	100.000%	158334	100.000%	919222
	31.486%		16.205%		35.084%		17.225%		100.000%

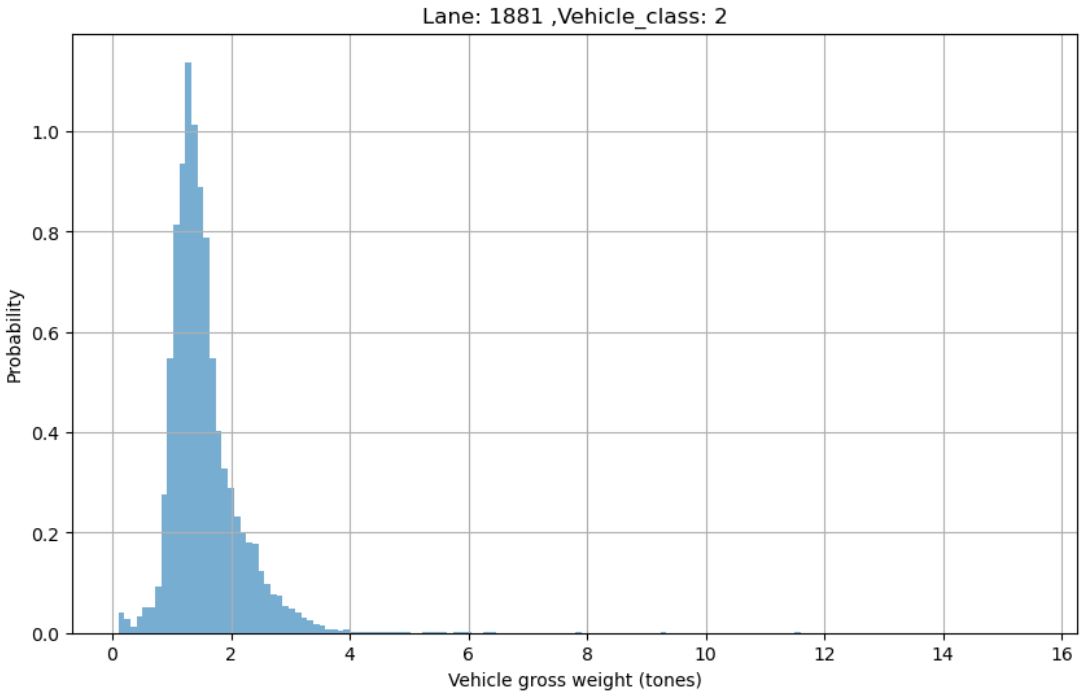


Figure 4.7: Class 2 vehicle gross weight distribution on lane 1881

Figure 4.7 shows an example of the vehicle gross weight distribution for 2-axle vehicles on lane 1881 (northbound fast lane). We can see that the vehicle gross weight on the bridge follows a probability distribution. Multimodal distribution is used to fit the histogram of the chosen vehicle attributes. A multimodal distribution can be defined as a probability distribution with more than one peak or distinct modes. In

the general case, a multimodal distribution could be any combination of distributions each having their own parameters. However, a typical example is a mixture of Gaussian distributions. The probability density function (PDF) of a multimodal distribution is given by:

$$f(x) = \sum_{i=1}^{k} w_i(g_i(x|\mu_i, \sigma_i)) \tag{4.8}$$

where $f(x)$ is the value of the PDF for value x. $g_i(x|\mu_i, \sigma_i)$ is the PDF of the individual Gaussian distribution with mean μ_i and standard deviation σ_i. w_i represents the proportion in the overall distribution that each Gaussian distribution represents.

The individual Gaussian distribution can be written as:

$$g_i(x|\mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2\right) \tag{4.9}$$

From Equation 4.8, we can see that i needs to be determined first. i represents the number of individual modes in the multimodal distribution model. The Bayesian Information Criterion (BIC) is a criterion for model selection. BIC can be expressed as:

$$BIC = kln(n) - 2ln(L) \tag{4.10}$$

where:

- n is the number of observations
- k is the number of parameters estimated by the model
- L is the maximised value of the likelihood function for the estimated model.

Based on these theories, a set of the potential number of modes, i, is defined, e.g. $[1, 6]$. Then, multimodal distributions with different i are fit to the vehicle attributes data (e.g. gross weight and speed). Next, the BIC for each fitted distribution is calculated and compared. The distribution with the minimal BIC value is selected. Figure 4.8 shows the BIC values to the multimodal distribution analysis for the

histogram in Figure 4.7. We can see that the BIC value decreases sharply when the number of modes i increases from 1 to 3. However, further increasing i has little effect on the BIC value. Therefore, the number of modes can be determined as 3 for this specific dataset. A trimodal distribution is used to fit this histogram, as shown in Figure 4.9. The parameters for the 3 modes in this trimodal distribution can be obtained for simulated vehicle generation. Figure 4.10 presents the PDF curve fitted for the speed attribute.

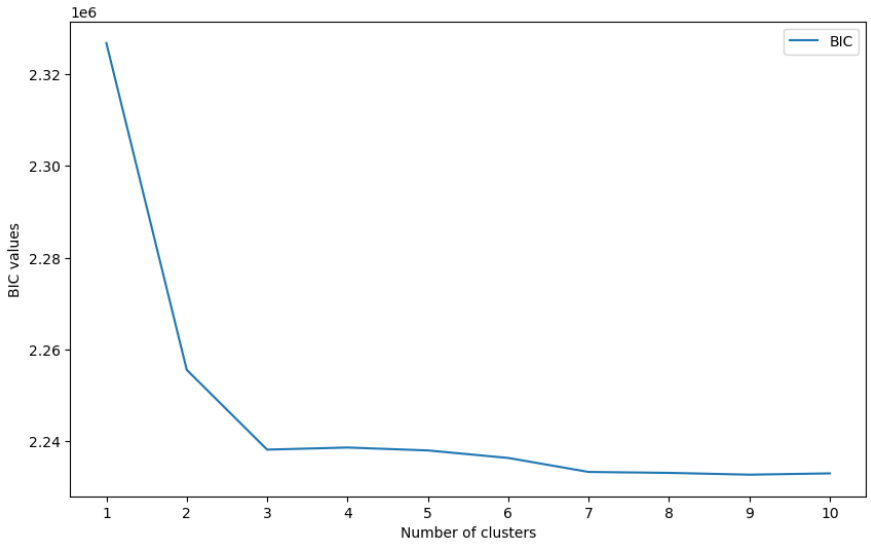


Figure 4.8: BIC plot

Subsequently, we implemented the multimodal fitting methodology across all vehicle categories, which includes eleven distinct classes, spanned over four lanes, for the evaluation of gross weight and vehicle speed distributions. This resulted in a total of sixty individual cases. Upon thorough examination of the Bayesian Information Criterion (BIC) plots corresponding to these cases, we ascertained that a trimodal distribution, with the number of modes i set to 3, adeptly characterises the data for

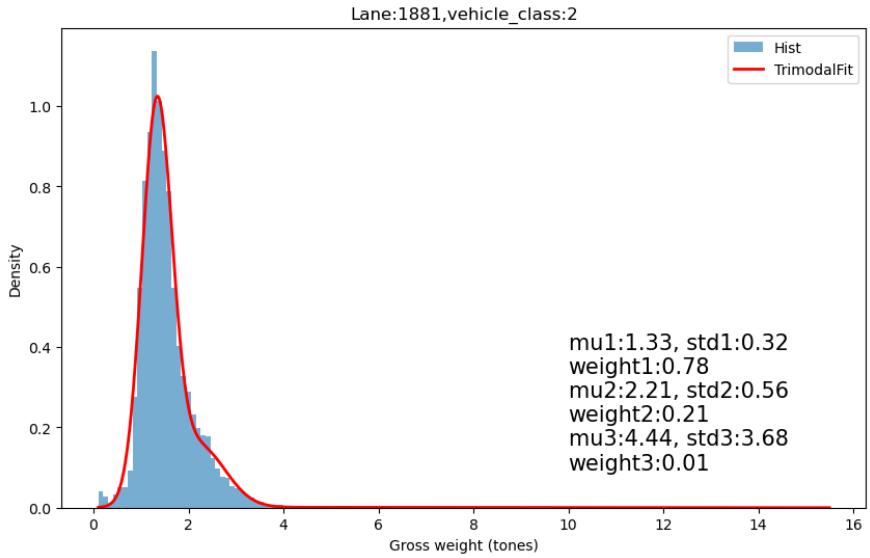


Figure 4.9: Trimodal fit for Class 2 vehicle gross weight distribution on lane 1881

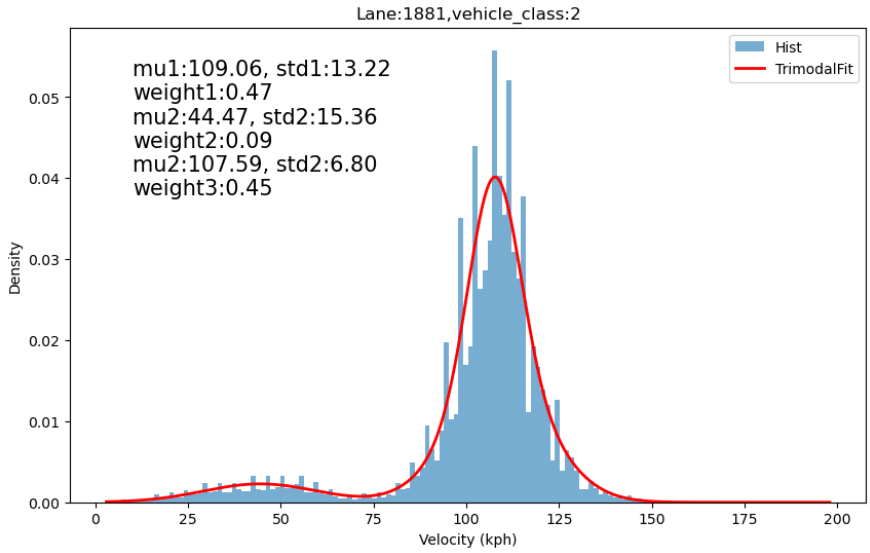


Figure 4.10: Trimodal fit for Class 2 vehicle speed distribution on lane 1881

each case under consideration. Therefore, in order to simplify the vehicle attributes simulating process, the trimodal distribution is employed to fit all 60 cases, where $i = 3$. The Probability Density Function (PDF) curves and the associated parameters of the models for all sixty cases are comprehensively presented in Appendix A.

Vehicle spatiotemporal distribution

Based on the average daily traffic volume, traffic volume distributions on each lane and traffic attributes distribution of each vehicle class, the vehicle spatiotemporal distribution on the QC can now be simulated. This simulation begins with the generation of a fleet of vehicles that are poised to enter the bridge. Three examples of this process can be seen in Table 4.5, which details the attributes of the first 12 vehicles in the fleet. The *Arrival time* denotes the time for the vehicle to arrive at the entrance of the bridge. In addition, the simulation calculates the speed, weight and number of axles. It is assumed that each vehicle, prior to entering the bridge, maintains a zero acceleration rate and sustains its initial speed. Subsequently, variables such as speed, acceleration, and position of each vehicle on the bridge are updated on a second-by-second basis. This is done in accordance with the car-following model and the lane-changing model.

Figure 4.11 depicts a fleet of vehicles traversing the northbound left lane (1880). Each individual lane in the illustration corresponds to the trajectory of a vehicle as it crosses the bridge. Within the specified 200-second timeframe, a total of 20 vehicles were recorded entering the bridge. In addition, we also observed different patterns. Primarily, the majority of vehicles were observed to maintain their initial speed throughout the course. Several vehicles accelerate due to the large gap to the front vehicle. Furthermore, an instance of lane change was documented during this observation period. With the microsimulation, we successfully simulated the spatiotemporal distribution of traffic loads on the Queensferry Crossing based on the observed traffic patterns.

Table 4.5: Simulated vehicle fleet

		1880	1881	1952	1953
1	Arrival time(s)	3.44	6.84	3.06	6.67
	Speed(kph)	113.04	102.49	102.38	108.32
	Weight(ton)	1.03	1.48	2.65	1.49
	Type	1	1	1	1
2	Arrival time(s)	6.70	14.04	6.01	13.32
	Speed(kph)	121.10	60.48	87.77	101.05
	Weight(ton)	1.40	1.31	1.16	2.44
	Type	1	1	1	1
3	Arrival time(s)	10.03	21.02	9.12	20.00
	Speed(kph)	103.75	110.12	123.52	113.33
	Weight(ton)	1.28	1.20	1.03	1.24
	Type	1	1	1	1

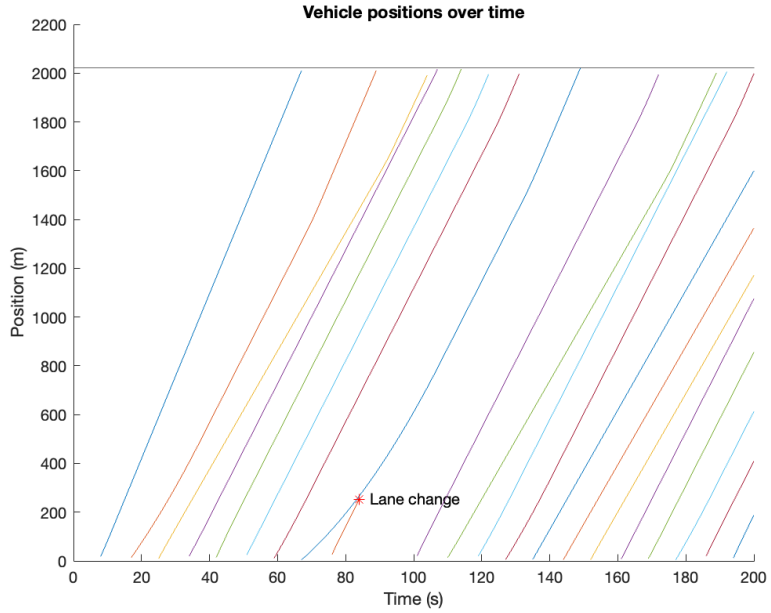


Figure 4.11: Trimodal fit for Class 2 vehicle speed distribution on lane 1881

4.3.2 Traffic-induced structural response

Deflections of the deck caused by traffic indicate changes in geometry, which introduce stress in the deck section. Furthermore, the movement of the deck due to cyclic traffic loading introduces stress in the expansion joints, which, over the long term, causes fatigue. Therefore, deflection measurements can serve as indicators of the bridge's general health status.

Based on the simulated spatiotemporal distribution of vehicle loads on the QC, the structural responses at pertinent locations can be deduced utilising the bridge's FE model.

First, the influence line needs to be obtained by performing a moving load analysis of the bridge. In order to carry out a moving load analysis, traffic lanes and vehicle loads were defined. The traffic load used here is the unit load defined by *Eurocode EN1991-2:2003 - road traffic actions and other actions specifically for road bridges* (BSI, 2004), clause 4.3.4 Load model 3 and NA.2.16. The UK National Annex describes two groups of vehicles, SV and SOV for bridge design. In this analysis, we used the SV196 model vehicle, loads and axles layout are depicted in Figure 4.12.

Then, the moving load analysis was performed in MIDAS by applying this unit load on each component across the traffic lanes. The displacement influence line at the location of interest can be plotted. Figure 4.13 shows the influence line for deflections in the vertical direction when the unit load was applied at the south middle span (node 1471 in the FE model). The maximum sagging deflection occurs at the south middle span, at 0.056mm and the maximum hogging deflection occurs at the north middle span, at 0.024mm.

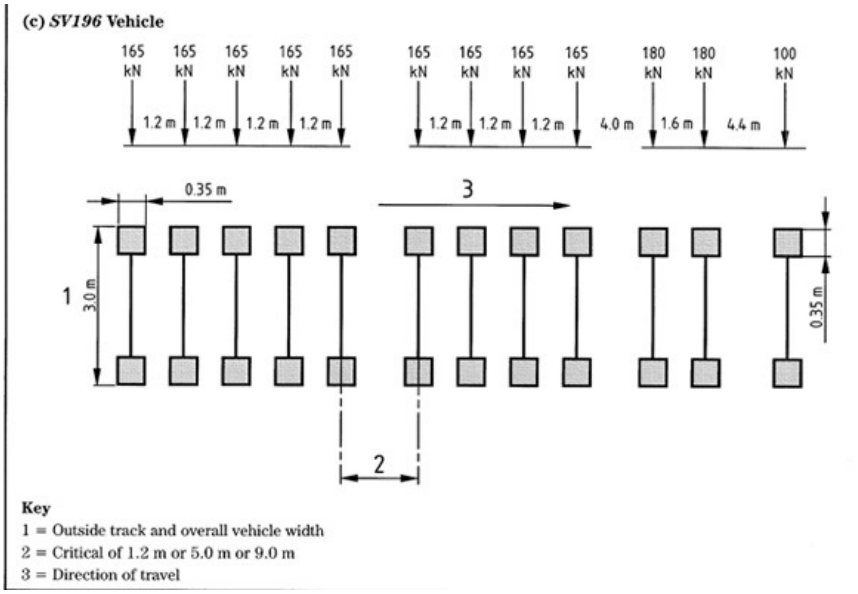


Figure 4.12: SV196 vehicle load distribution

Now, given an instant t, the number of vehicles on the main bridge, their locations and weights can be derived from vehicle tracks, e.g. Figure 4.11. Then, the weights of the vehicles are applied to the corresponding locations in the FE model of the bridge to obtain the deflection at the location of interest. However, the dynamic impact of moving vehicles on bridges must also be considered. A dynamic impact factor (μ) is normally used in various design codes to account for this impact (L. Deng, Yu, Zou, & Cai, 2015). Then, the actual vehicle loads on the bridge are calculated as:

$$P_{live} = (1 + \mu)P_{vehicle} \tag{4.11}$$

$$\mu = \phi - 1 \tag{4.12}$$

where:

P_{live} is vehicle live loads considering dynamic effects

μ is the dynamic impact factor

ϕ is the dynamic amplification factor

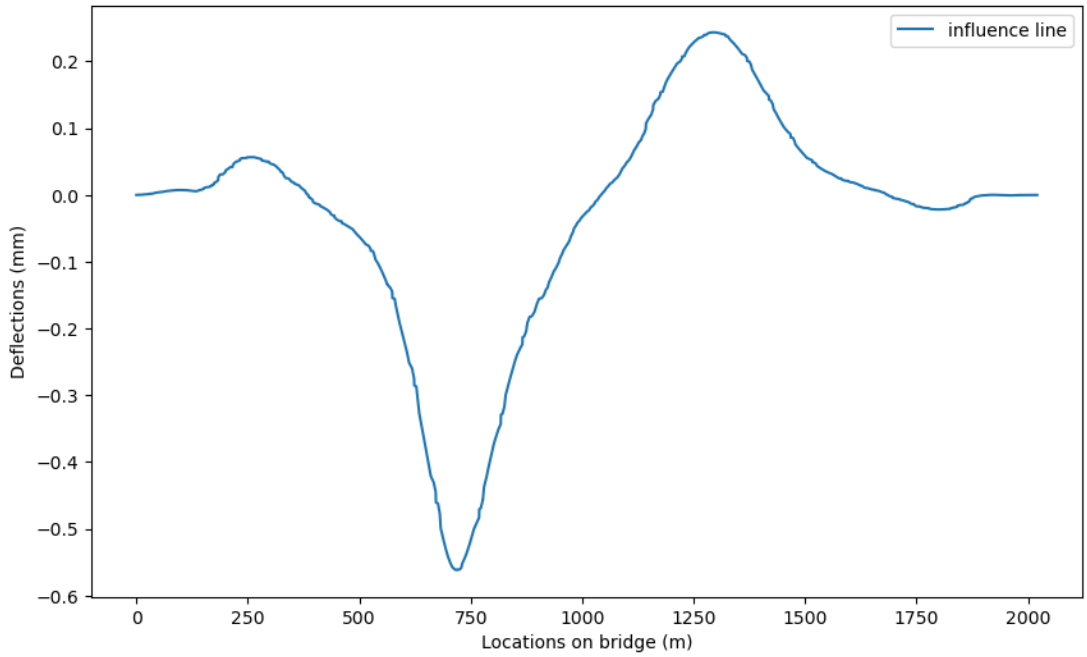


Figure 4.13: Influence line

In the *Eurocode EN1991-2:2003 - road traffic actions and other actions*, the dynamic amplification factor is determined as a function of the bridge span length for one, two and four-lane bridges. For two-lane bridges, ϕ is specified as:

$$\phi = \begin{cases} 1.3 - \frac{0.4}{100}L & L \leq 50m \\ 1.1 & L > 50m \end{cases} \tag{4.13}$$

where L is the bridge span length. For four-lane bridges, ϕ is specified as a constant of 1.1. In our case, the QC is a dual-carriageway bridge with 3 lanes on each carriageway. Therefore, ϕ is determined as 1.1 and the dynamic impact factor μ is 0.1.

Figure 4.14 presents an example of the simulated deflections at the middle span of the southern section of the bridge within a one-hour timeframe at a sampling rate 1Hz, the same as the QC SHM system raw deflection data. The deflections range from -80.19mm to 29.74mm in this simulation period.

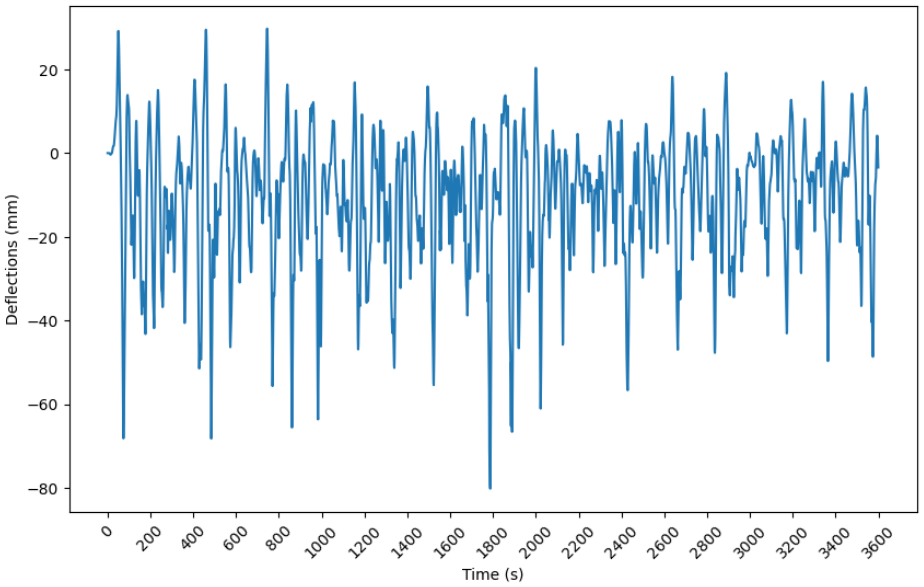


Figure 4.14: Simulated south middle span deflections

4.3.3 Validation

Data from sensors 826 and 957 on the QC were used to validate the effectiveness of the proposed simulation method. Both sensors are GPS rover stations installed on the deck rack and located at the south middle span, as shown in Figure 4.15. The displacements in the vertical direction were extracted from both sensors.

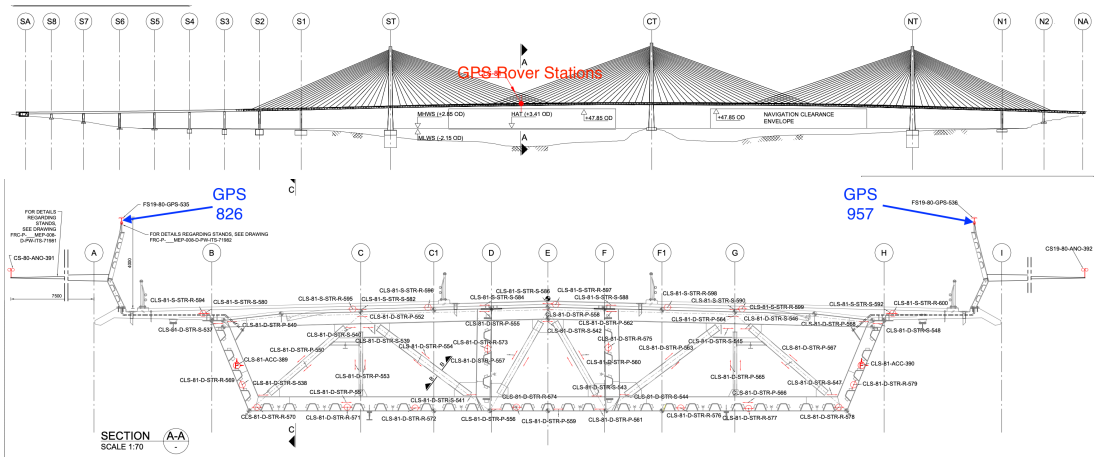


Figure 4.15: Sensor 826 & 957

In Figure 4.6, we have learnt that the traffic flow on the QC follows a specific pattern throughout the day and the traffic volume is calculated on an hourly basis according to Table 4.5. Observations indicate that the hourly traffic flow oscillates between 150 and 3000 vehicles. As such, to verify the proposed simulation model, it is imperative to conduct verifications under varying traffic flow conditions. Hence, the following hourly traffic volumes are established for the execution of the simulation program: 500, 1000, 1500, 2000, 2500, and 3000. These values are designed to reflect the actual variations in traffic flow on the QC. The actual monitored values are shown in Figure 4.16. The data are from 03/11/2020. The equivalent daily traffic volumes for these scenarios are 24000, 48000, 72000, 96000, 120000, and 144000, respectively. Then, simulated south midspan deflections are generated using these values, as shown in Figure 4.17.

Figure 4.17 shows that the simulated midspan deflections have similar patterns as the actual monitored values for different preset traffic flows. In this instance, error metrics are not utilised to compare the simulated and observed values. This decision stems from the fact that the vehicle simulations on the QC are predicated on observed vehicle attribute distributions. As such, the simulation represents the overall traffic flow rather than individual passing vehicles. Consequently, the proposed traffic load estimation strategies prove satisfactory for modelling the actual traffic loads on the Queensferry Crossing.

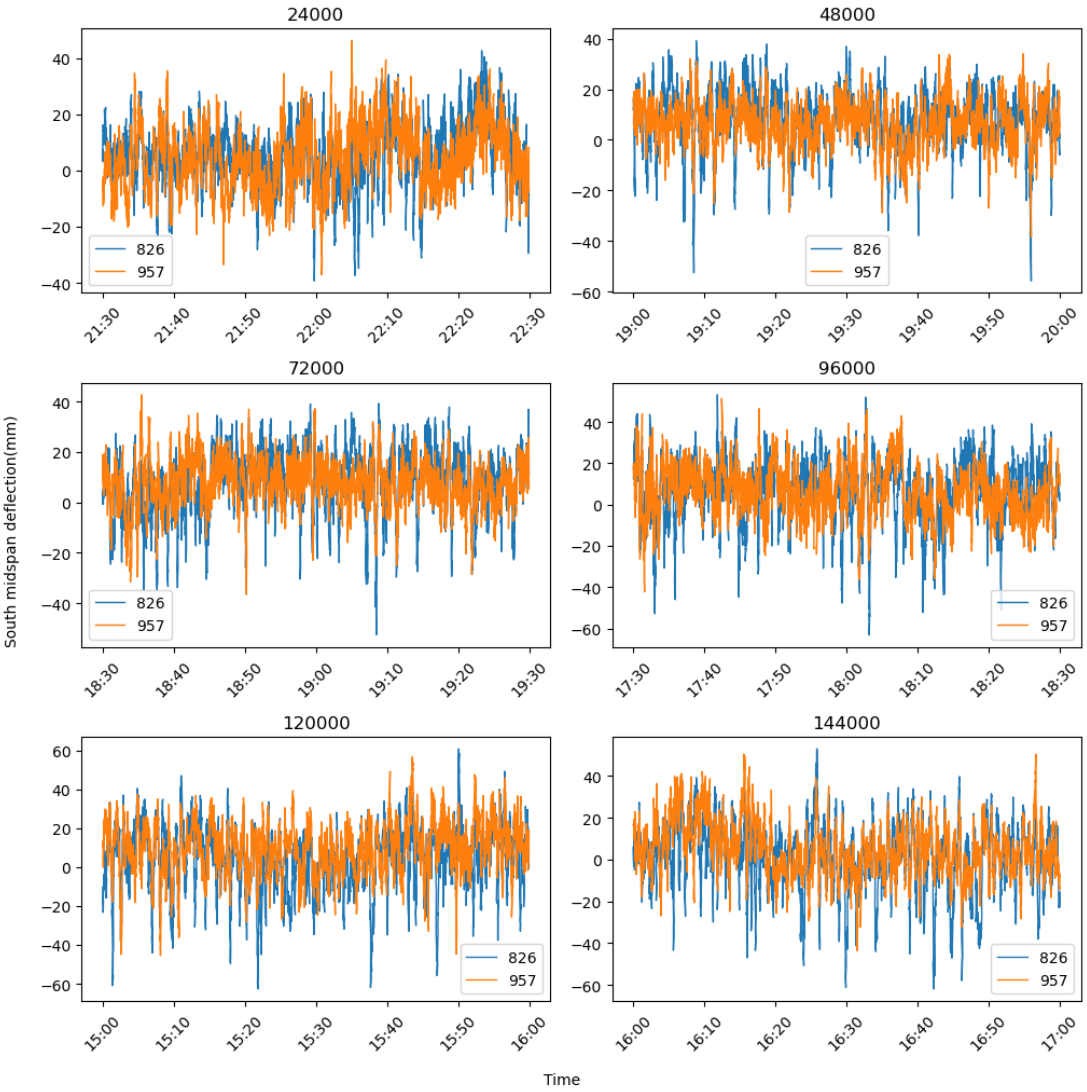


Figure 4.16: Monitored south midspan deflections

4.4 Summary

In this chapter, a traffic load simulation strategy for the Queensferry Crossing is proposed. The DWIM system, which is employed to record each passing vehicle on the QC, exhibits suboptimal functionality, rendering the monitored traffic data unreliable and unsuitable for direct use in analysis. Consequently, a simulation strategy was proposed to model vehicle flows on the bridge based on daily traffic volumes, vehicle attributes distributions and the displacement influence line from the bridge FE model.

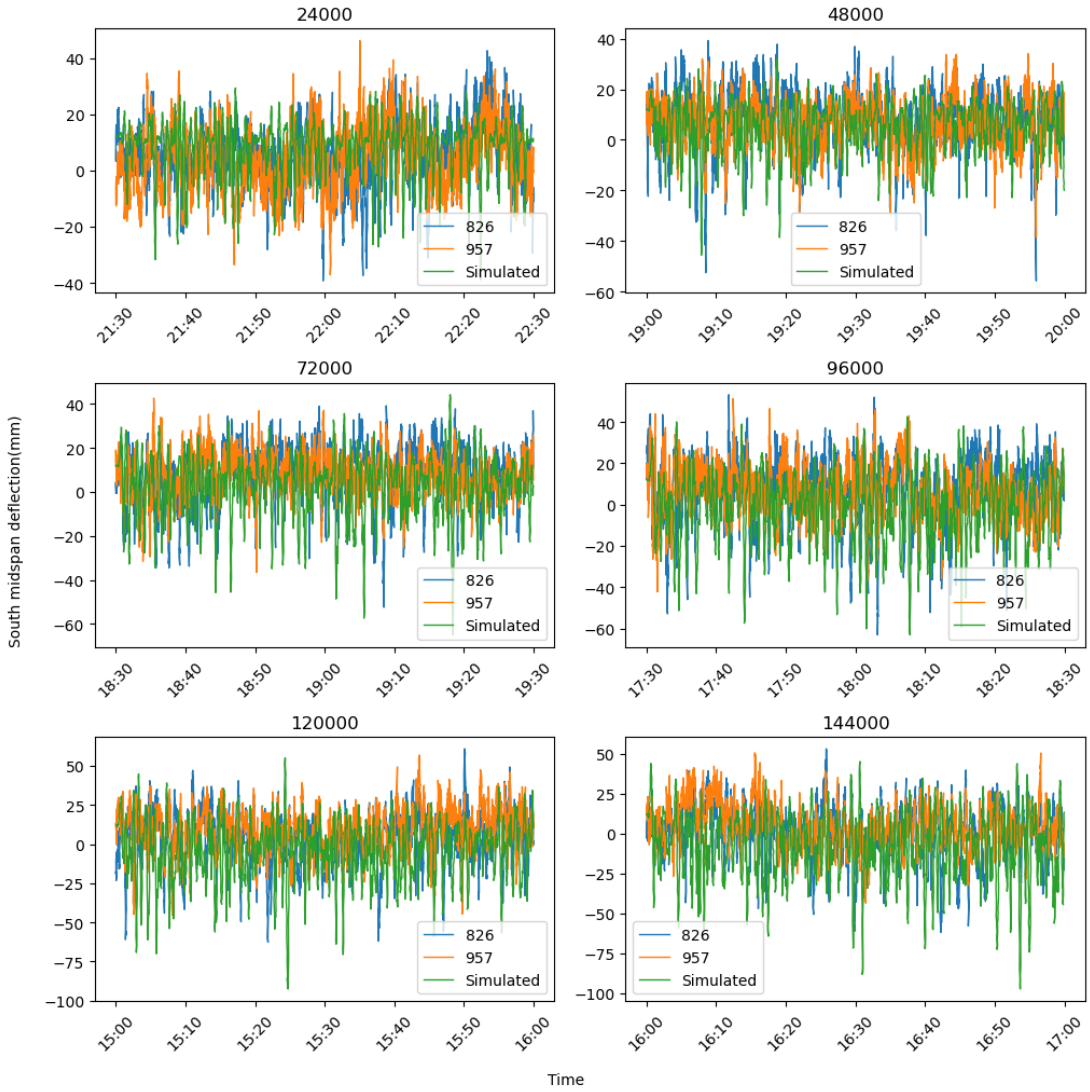


Figure 4.17: Simulation validations

On the other hand, the DWIM system can only yield vehicle information at a fixed location on the bridge, such as the entrance and the exit of each lane. To compute the bridge response at any given location, it is necessary to account for traffic behaviours on the bridge. In this regard, car-following and lane-changing models are also incorporated into the simulations. As a result, the locations and attributes of a fleet of simulated vehicles can be modelled on a second-by-second basis. Subsequently, the deflections at the location of interest (e.g., midspan) can be calculated using its influence line for a pre-set period of time.

Lastly, the simulated south midspan deflections were validated with GPS monitored values. Six scenarios, each representing different traffic volumes, were established to represent varying traffic conditions throughout the day on the bridge. Comparisons revealed that the simulated deflections exhibited similar fluctuations in terms of patterns and ranges across all scenarios. Therefore, it can be inferred that the proposed traffic load simulation strategy is fit for use.

We use deflections exclusively when discussing traffic-induced responses, as deflection represents the global response. Vehicles positioned at any location along the bridge deck contribute to mid-span deflections. In contrast, strain represents a local response and does not accurately reflect the overall effect of traffic on the bridge deck.

Next, the simulated traffic flows will be used in the prediction model training in the next chapter. Moreover, the proposed simulation strategy can serve as a universal approach for different bridges with the following inputs: bridge length, number of lanes, daily traffic volume and vehicle attributes distributions (speed, class and weight).

Chapter 5

Traffic-induced response predictions

5.1 Backgrounds

In Chapter 4, a strategy for simulating traffic loads was introduced. This simulation can emulate traffic loads on a bridge, taking into account traffic volume and the distribution of vehicle attributes (e.g., speed, weight, vehicle class). The results of this simulation were subsequently corroborated with data from monitored structural responses. It has been demonstrated that the proposed method is both efficient and precise in simulating midspan deflections.

In a real-world bridge monitoring scenario, the DWIM system is typically installed at either the entrance or exit of the bridge. As such, the traffic information provided by the DWIM is only applicable to its installation locations. To establish the relationship between the traffic loads and the structural response at a specific location p along the main span at time t, it is not only the vehicle recorded at time t that is relevant, but also vehicles that have passed previously. Consequently, the method used to predict structural response based on DWIM data should be capable of capturing time dependencies within temporal sequences.

We propose a prediction model using the recurrent neural network (RNN). The development of RNNs has been driven by the need to process data where temporal dynamics and sequence order are critical. Information flows in one direction from the input layer through one or more hidden layers to the output layer in a traditional

feed-forward model, e.g., MLP. Therefore, each training cycle is independent. In this regard, previous information or "states" are disregarded after each training, making them unsuitable for tasks that require an understanding of temporal relations. In contrast, RNNs have a cyclic structure where connections between nodes form a directed cycle, creating a "memory" that preserves previous training information. Due to their ability to process sequential data, RNNs are often used in tasks such as language modelling, machine translation, and speech recognition.

When handling long-term time series, the simple RNN model often encounters issues related to gradient explosion or gradient vanishing. A detailed discussion on this matter will be presented in the Methodology section.

In this chapter, we will use RNN to investigate traffic-induced structural responses and compare results with traditional methods. Furthermore, we also studied the effect of different parameters of RNN on the prediction accuracy.

5.2 Methodology

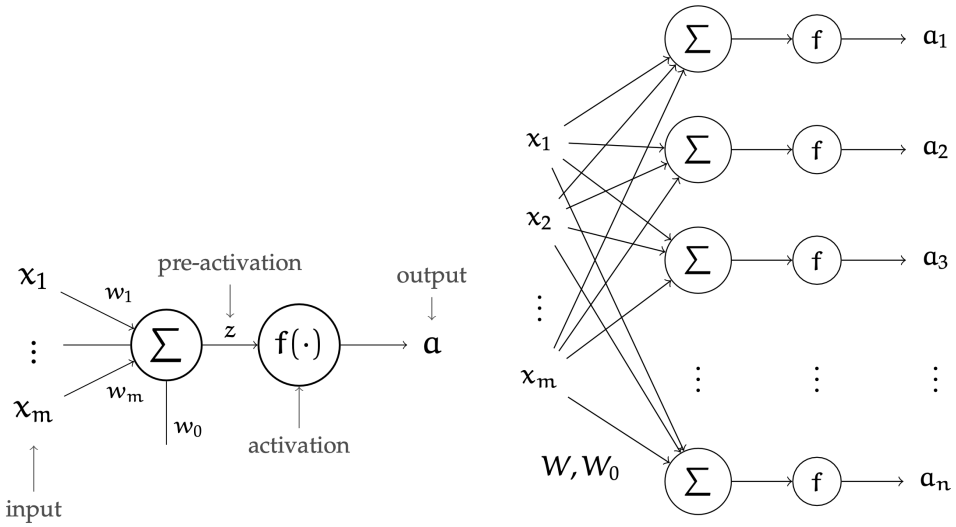
In this section, the primary mechanism of recurrent neural networks, the training and validation strategy and parameter analysis methods are introduced.

5.2.1 Feedforward neural network

The basic element of a neural network is a "neuron", also known as a "node" or "unit". The function of a neuron can be expressed as:

$$\alpha = f(z) = f\left(\sum_{j=1}^{m} x_j w_j + w_0\right) = f(w^T + w_0) \tag{5.1}$$

where f is an *activation function*, x is an *input* vector $x \in \mathbb{R}^m$, w is a *weight* vector $(w_1, ..., w_m) \in \mathbb{R}^m$ and $w_0 \in \mathbb{R}^m$ is an *offset*. The neuron can be described in Figure 5.1a. A *layer* is a set of neurons that are not connected to each other, described in Figure 5.1b. This layer is called *fully connected* which serves as hidden layers in MLP or the last several layers after convolutional layers in CNN or recurrent layers in RNN.



(a) A neuron **(b)** A layer

Figure 5.1: Neuron and layer

The *activation functions* in a neural network are normally non-linear functions except for the output layer in certain conditions. There are a few commonly used activation functions in neural networks, shown in Figure 5.2:

Step function

$$step(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{otherwise} \end{cases} \tag{5.2}$$

Rectified linear unit

$$ReLU(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases} \tag{5.3}$$

Sigmoid function, also known as *logistic* function

$$\sigma(z) = \frac{1}{1 + \sigma^{-z}} \tag{5.4}$$

Hyperbolic tangent

$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{5.5}$$

Softmax function

$$softmax(z) = \begin{bmatrix} \exp(z_i)/\sum_i(z_i) \\ ... \\ \exp(z_n)/\sum_i(z_n) \end{bmatrix} \tag{5.6}$$

Sigmoid and hyperbolic tangent functions are commonly used in hidden layers. The sigmoid function is normally used in the output layer for binary classifications and softmax is for multi-class classification. For regression predictions, the output layer employs the linear function.

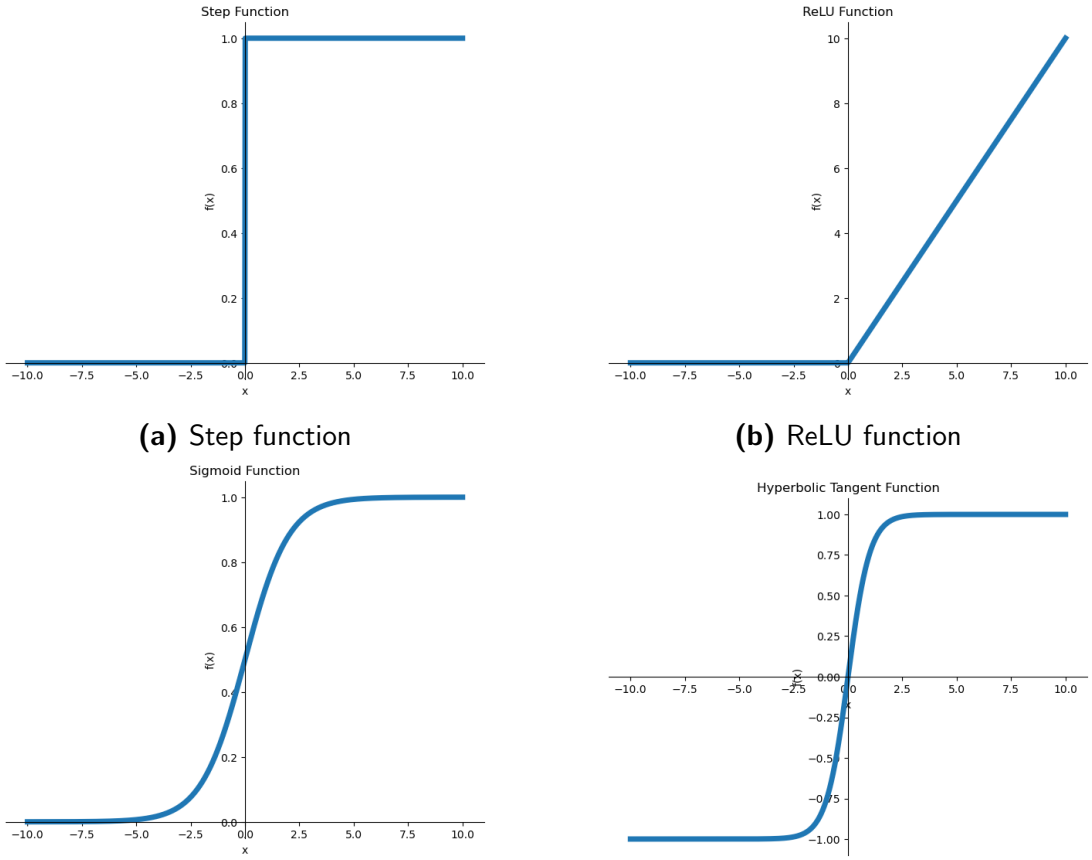
The training of a neural network minimises the error between the *output* a and the *actual* observations y. This error is normally described as a loss function L:

$$\sum_i L\left(f(x^i; w, w_0), y^i\right) \tag{5.7}$$

The minimisation of the loss can be achieved by a process called the gradient descent, an optimisation algorithm iteratively moving in the direction of the steepest descent as defined by the negative of the gradient:

$$W^l = W^l - \alpha \cdot \frac{\partial L}{\partial W^l} \tag{5.8}$$

$$W_0^l = W_0^l - \alpha \cdot \frac{\partial L}{\partial W_0^l} \tag{5.9}$$



(a) Step function

(b) ReLU function

(c) Sigmoid function

(d) Hyperbolic tangent funtion

Figure 5.2: Commonly used activation functions in neural networks

where

α is the learning rate, determining the step size during each iteration while moving toward a minimum of a function

$\frac{\partial loss}{\partial W^l}$ and $\frac{\partial loss}{\partial W_0^l}$ is the gradients of the loss function with respect to weights

The gradients are calculated using an algorithm called *Backpropogation*. It works by computing the derivative of the loss function with respect to each weight by using the chain rule, moving backwards in the network from the output layer to the input layer. The result of backpropagation is a gradient, or a vector of derivatives, which indicates the direction in which the loss function increases most rapidly. For example,

output A in layer l can be written as:

$$Z^l = W^{lT} A^{l-1} + W_0^l \tag{5.10}$$

$$A^l = f^l(Z^l) \tag{5.11}$$

The gradients of the loss function with respect to weights can be expressed as:

$$\frac{\partial L}{\partial W^l} = \frac{\partial L}{\partial Z^l} \cdot \frac{\partial Z^l}{\partial W^l} \tag{5.12}$$

$$\frac{\partial L}{\partial W_0^l} = \frac{\partial L}{\partial Z^l} \cdot \frac{\partial Z^l}{\partial W_0^l} \tag{5.13}$$

where

$$\frac{\partial L}{\partial Z^l} = \frac{\partial L}{\partial A^l} \cdot \frac{\partial A^l}{\partial Z^l} \tag{5.14}$$

Gradient descent is repeated until the algorithm converges to an optimal solution.

5.2.2 Recurrent neural network

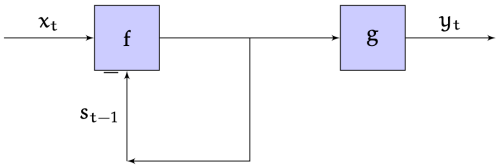


Figure 5.3: A RNN cell

Recurrent neural networks (RNN) are a particular type of state machine model. Unlike feedforward neural networks, RNNs use an internal state to process the sequence of inputs. The structure of the basic RNN model is shown in Figure 5.3. For the hidden layer of the RNN, it receives the input vector and the previous state vector, and

updates the hidden state vector using the following equation:

$$h_t = f_1(W^{sh}x + W^{sh}h_{t-1} + W_0^{sh}) \tag{5.15}$$

Then it generates an output vector:

$$y_t = f_2(W^O s + W_0^O) \tag{5.16}$$

Where W^{sh}, W^{sh} and W^O are weight matrices, W_0^{sh} and W_0^O are bias, f_1 and f_2 are activation functions to create non-linearity. During the training process, the parameters are updated by back-propagation through time. The backward propagation of the gradient along the sequence can be expressed as:

$$\delta^{h_{t-1}} = \frac{\partial h_t}{\partial h_{t-1}} \cdot [\frac{\partial Loss_{elt}(p_t, y_t)}{\partial h_t} + \delta^{h_t}] \tag{5.17}$$

In a scenario where the output at the end of the sequence is incorrect and relies on the input at the initial time step via weights, the loss at step n will be multiplied by:

$$\frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_3}{\partial h_2} \cdots \frac{\partial h_n}{\partial h_{n-1}} \tag{5.18}$$

Therefore, the loss in this case would either grow or shrink exponentially, making it hard to train. This is called the gradient vanishing or blowing up. (Hochreiter, 1998) This will prevent the RNN from learning from long-term dependencies. (Bengio et al., 1994) To solve this problem, the idea of *gating* was introduced in the hidden layer.

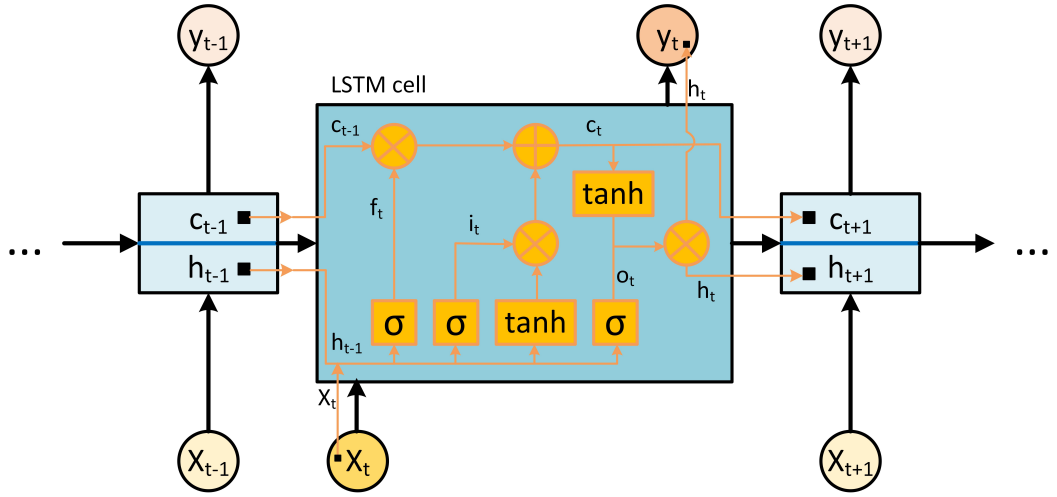


Figure 5.4: Basic cell architecture of the hidden layer in the LSTM networks

Long Short-Term Memory

Currently, the most popular modified LSTM structure was proposed by Graves and Schmidhuber (2005), called vanilla LSTM. The detailed structure of a vanilla LSTM cell is shown in Figure 5.4.

Three kinds of gates are added in the cell of the LSTM network to control the flow of information in the layers, forget gates, input gates and output gates.

First, a forget gate is used to decide whether to preserve the historical information from the cell state:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.19}$$

where σ is a sigmoid function with output $[0, 1]$.

The next step decides the new information stored in the cell state by an input gate, i_t. Then another forget gate is applied to create an intermediate vector, $\widetilde{C}_t$. Then the previous cell state, C_{t-1}, will be updated by the combination of the i_t and $\widetilde{C}_t$:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.20}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5.21}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{5.22}$$

where tanh is the hyperbolic function with output $[-1, 1]$.

In the final step, an output gate is used to decide which part of the updated cell state will be outputted:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5.23}$$

$$h_t = o_t * \tanh(C_t) \tag{5.24}$$

Therefore, in this way, LSTM has the ability to update the cell state based on previous output and current input, allowing information in the layer to be refreshed, instead of simply overriding the entire cell state in traditional RNN.

5.2.3 Training strategy

The general strategy for training, validation and testing using the LSTM network is shown in Figure 5.5. The dataset is split into training, validation, and test sets in any neural network training.

The conventional partitioning ratio for the dataset is typically established as 8:1:1, corresponding to the training, validation, and testing subsets, respectively. However, this specific ratio should be adjusted according to the dataset's total size and the hardware's specifications. Three main flows are involved in the strategy to train, validate and test the network model.

The training process is employed to enhance the predictive performance of the LSTM network. Initially, a training batch is randomly selected from the training dataset to constitute the network input. Subsequently, the network parameters are incrementally updated based on the loss computed between the predicted and actual values. Upon

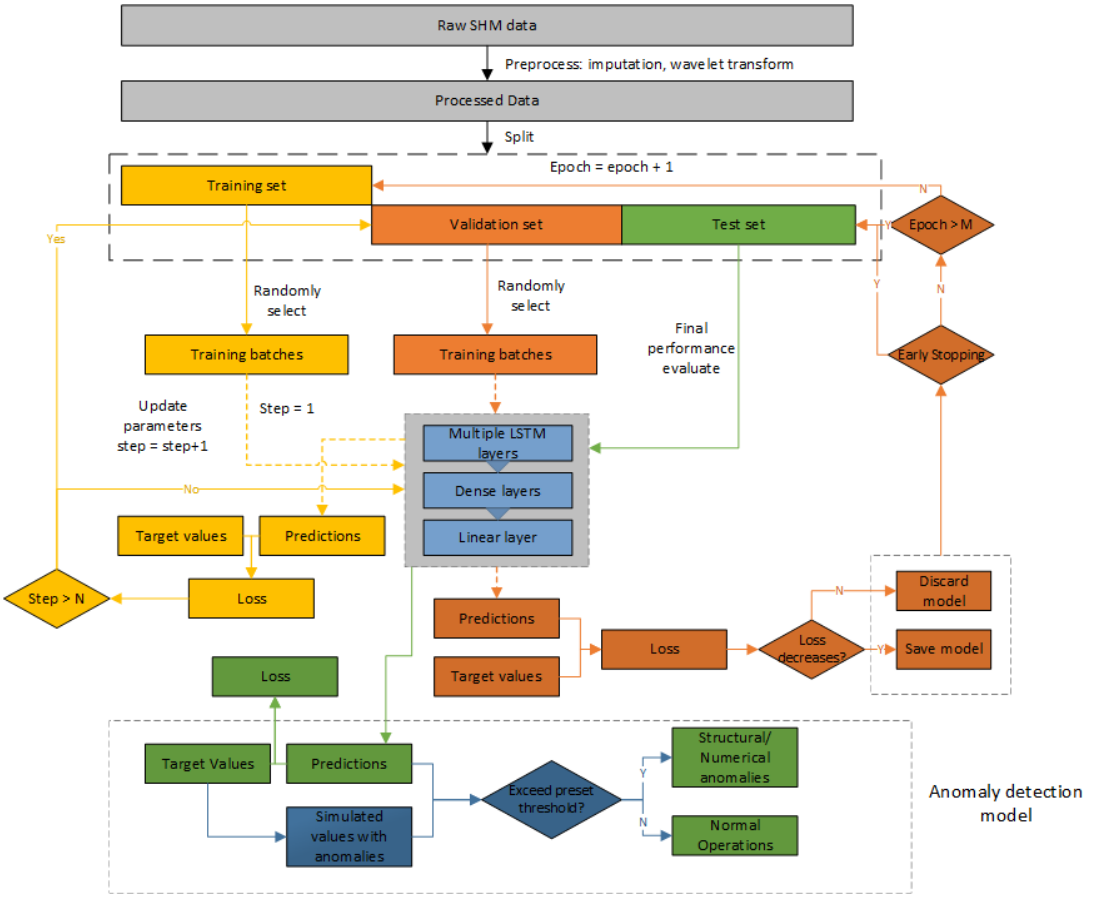


Figure 5.5: The general strategy for training, validation and testing by using LSTM networks

reaching a predetermined step value, denoted as N, the training for the current batch concludes, and the validation process commences. It is noteworthy that the training process will persist cyclically until an early stopping criterion is met or the predetermined epoch, denoted as M, is reached. An early stopping is a regularisation technique to prevent overfitting by terminating the training process once the model's performance on a validation set stops improving or even starts to deteriorate.

The validation process serves as a mechanism to evaluate the worthiness of preserving the trained model. Following each batch training, the trained network model is assessed using the validation data. Like the training process, a validation batch is initially randomly generated from the validation dataset. Subsequently, the loss

associated with this batch model is compared with the loss corresponding to the preceding batch model. If an improvement in the loss is observed, the current batch model is retained; otherwise, the model is discarded. This process aids in mitigating overfitting and ensuring the generalisability of the model. Overfitting is a phenomenon in machine learning where a model learns the training data too well, to the point that it performs poorly on unseen data due to its inability to generalise.

The testing process is employed to definitively assess the predictive performance of the model. Unlike the training and validation processes, the entirety of the test dataset is used as the input for the network. The loss computed between the predicted and actual values serves as the metric for evaluating the predictive performance of the trained RNN model. Given that the test dataset is never incorporated into any training or validation phases, it presents entirely novel data to the network model. Consequently, the performance of the network model, as evaluated using the test dataset, provides a reliable measure of the model's predictive capabilities in this strategy.

5.2.4 Parameter analysis

The proposed strategy has seven primary hyperparameters: data size, split ratio, window size, training batch size, validation batch size, number of neurons in the LSTM layer, and number of LSTM layers. The specific interpretations and roles of these hyperparameters within the context of the strategy are elaborated upon as follows:

- **Data size**. It is the size of the entire dataset prepared for the training.
- **Split ratio** As depicted in Figure 5.5, the entire dataset is split into three sections via a predetermined split ratio.

- **Windows size** As depicted in Figure 5.5, both the training and validation processes are conducted on a batch basis, wherein the window size corresponds to the length of a single sample within the batch. It is common practice to utilise natural time windows within the network, such as intervals of one hour or one day.
- **Training batch size** The number of samples used to train the network model.
- **Validation batch size** The number of samples used to validate the network model.
- **Number of neurons in the LSTM layer** This hyperparameter determines the scale of the network.
- **Number of LSTM layers** This hyper-parameter determines the scale of the network as well.

Generally, a large-scale dataset can be advantageous in mitigating overfitting, thereby achieving satisfactory predictive performance on the test dataset. However, an increase in the scale of the dataset leads to a corresponding rise in the cost of data acquisition and computational expenses. In reality, it is a trade-off between the data scale and cost. We used Graphics Processing Units (GPUs) to train all neural network models on the ECDF Linux Computer Cluster at the University of Edinburgh. Models are constructed using the Python package, Tensorflow.

The dimensions of the training or validation batch are determined by two parameters: the window size and batch size. Generally, the window size remains constant while the batch size is adjusted to modify the training or validation data size. To mitigate the overfitting issue, a larger batch size in practice is often preferable. However, due to the limitation of GPU memory, the batch size should fit the capacity of the computing machine. Consequently, the guiding principle for determining the batch size is to maximise it within the confines of the available GPU memory.

Based on the preceding discussions, three hyperparameters have been selected for further analysis: data size, number of neurons in the LSTM layer, and number of LSTM layers. For the data size, the scale is incrementally increased from data spanning 10 days, sampled at a rate of 1Hz, to data spanning 200 days, with increments of 10 days. Regarding the number of neurons in the LSTM layer, values range from 16 to 288, within the framework of a single-layer LSTM. Regarding the number of LSTM layers, configurations of single, double, and triple LSTM layers are explored, with varying numbers of neurons in each layer.

5.3 Case Study

5.3.1 Data preparations

In the context of this study, the deflection of the girder at the south mid-span is selected as the target output. Utilising the simulation process introduced in Chapter 4 at a sampling rate of 1Hz which is the same as the sampling rate on the SHM system on the Queensferry Crossing, the simulated data corresponding to the south mid-span are generated. A one-hour time window of data is presented in Figure 5.6.

The input data are the speed and gross weight of all simulated vehicles passing the south mid-span. The simulation program generates discrete vehicle data on the QC by updating its trajectory on a second basis. Given the requirement for LSTM networks to process time sequences, the recorded vehicle information is transformed into a matrix with 8 features. This matrix specifically includes the gross weight and speed of vehicles traversing the four lanes on the bridge. To form a complete time sequence input, instances without records are filled with zeroes. A segment of the input data, for illustrative purposes, is presented in Table 5.1. The output data is the simulated south mid-span deflections.

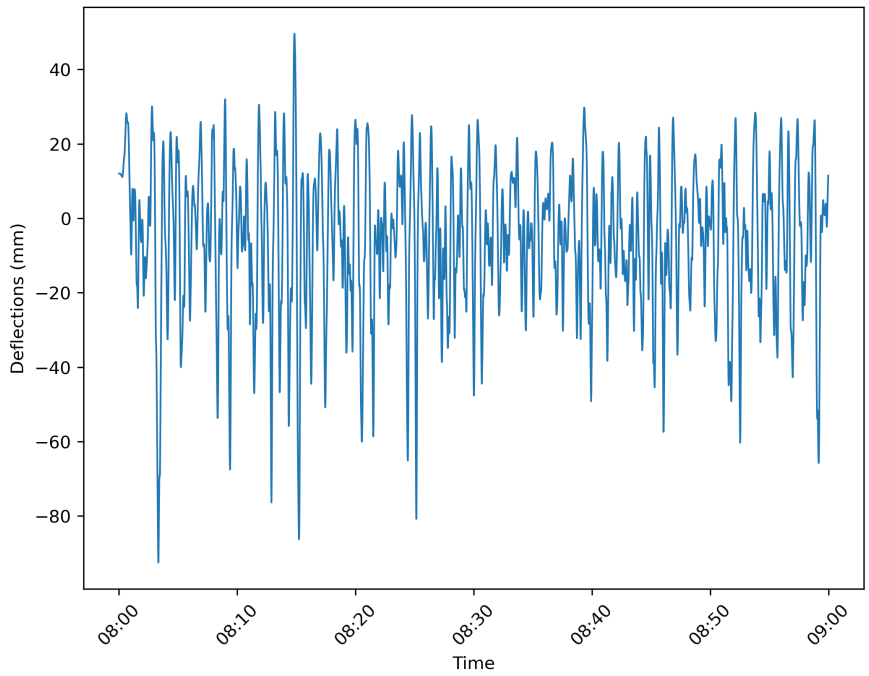


Figure 5.6: Simulated one-hour south mid-span deflections

Normalisation is crucial in preparing data for neural network training, as it scales the input and output data to a uniform range. In this study, the Min-Max scaling method is employed to normalise the features, effectively transforming them into a range of $[0, 1]$.

Next, the prepared data was partitioned into training, validation, and test sets, with a ratio of 8:1:1, respectively. The training set served the purpose of training the neural network models. The validation set was employed to assess the model following each training epoch, facilitating the tuning of the model's parameters and aiding in the prevention of overfitting. Lastly, the test set was utilised to evaluate the performance of the trained model on data that it had not previously encountered.

Table 5.1: Reconstructed input time sequence

Time	Lane 1		Lane 2		Lane 3		Lane 4	
	V_1	W_1	V_2	W_2	V_3	W_3	V_4	W_4
09:43:04	0.00	0.00	0.00	0.00	27.26	2.81	0.00	0.00
09:43:05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:06	0.00	0.00	0.00	0.00	29.46	2.81	0.00	0.00
09:43:07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:10	0.00	0.00	33.43	2.31	0.00	0.00	0.00	0.00
09:43:11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:13	0.00	0.00	0.00	0.00	28.18	15.75	0.00	0.00
09:43:14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09:43:15	30.01	23.66	0.00	0.00	0.00	0.00	0.00	0.00
09:43:16	0.00	0.00	0.00	0.00	0.00	0.00	29.87	1.87

(1) $V_i \& W_i$, vehicle velocity and gross weight in lane i.
(2) 1 is the northbound slow lane, 2 is the northbound fast lane, 3 is the southbound slow lane, 4 is the southbound fast lane.
(3) The unit for velocity is m/s; for weight is ton.

5.3.2 Model training

Within the scope of the training process, an initial training batch is randomly generated from the prepared training dataset. The size of this training batch is the product of the window size and the batch size, with the window size set at 3600 and the batch size determined through a process of trial and error. The LSTM network was trained 50 times (50 epochs). As each epoch's training batch is generated randomly, the model's dependency on a specific dataset is diminished, thereby enhancing its robustness and leading to improved generalisation. While a single-layer LSTM network already constitutes a deep architecture, given its nature as a feed-forward neural network unrolled over time, deep LSTM networks offer more efficient utilisation of parameters. This is achieved by distributing these parameters across multiple layers in the network, thereby enhancing the model's capacity to capture complex patterns and dependencies in the data (Sak, Senior, & Beaufays, 2014).

In addition to the input and output layers, between one and three LSTM layers were incorporated into the network design and trained using the provided dataset, with the aim of examining the impact of the number of LSTM layers on prediction performance. Following the LSTM layers, two fully-connected layers - specifically, a dense layer and a linear layer - were integrated to form the neural networks. These layers comprised 32 and 1 neuron, respectively. The number of neurons in the output layer must align with the dimension of the output data, in our case, only one dimension.

Adaptive moment estimation (ADAM) was used as the optimiser to adjust the parameters of the model. This choice was motivated by ADAM's relatively low memory requirements. The learning rate α was set to 0.001. The learning rate was reduced by 0.1 when the validation loss stopped improving. Mean absolute error (MAE) was chosen as the loss function and mean squared error (MSE) was adopted as a separate reference metric. They can be written as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{5.25}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5.26}$$

where y_i is the model predictions, $\hat{y}_i$ is the actual value. The training process is designed to iteratively adjust the model parameters with the objective of minimising the loss. The number of iteration steps was set to 200 in this study.

Upon completion of the iteration process, the algorithm transitions into the validation flow. During this phase, a validation batch is extracted from the validation dataset, and the predicted deflections and validation loss are computed. In this particular case study, the validation process was executed iteratively ten times. To mitigate the risk of overfitting, the early stopping technique was employed. This technique

stipulates that the training process will be terminated if the validation set loss ceases to decrease for ten consecutive iterations. Subsequently, a new round of the training process commences, persisting until the epoch approaches the predetermined value M or until the early stopping criterion is met.

After completing all the epochs or triggering early stopping, the testing phase was initiated. The entire test set was utilised as the input. The loss on the test dataset was computed based on a comparison between the predicted values and the ground truth. Additionally, other metrics, such as the correlation coefficient, were also employed to evaluate the predictive performance of the model:

$$R = \frac{\sum_{i=1}^{n}(y_{\text{true},i} - \bar{y}_{\text{true}})(y_{\text{pred},i} - \bar{y}_{\text{pred}})}{\sqrt{\sum_{i=1}^{n}(y_{\text{true},i} - \bar{y}_{\text{true}})^2 \sum_{i=1}^{n}(y_{\text{pred},i} - \bar{y}_{\text{pred}})^2}} \tag{5.27}$$

where y_{pred} is the LSTM predictions on the test set, y_{true} is the mid-span deflections generated with the Matlab program which were treated as monitored data.

5.3.3 Initial Results

Initially, simulation data spanning 10 days (equivalent to 864,000 samples) were employed to train, validate, and test a one-layer LSTM network with the proposed training strategy. The network was configured with a window size of 3600, a training batch size of 128, a validation batch size of 20, and 32 neurons in the LSTM layer. Consequently, the loss functions, specifically MAE in this instance, corresponding to the epochs within both the training and validation processes, are graphically represented in Figure 5.7. Initially, the training loss was observed to be greater than the corresponding validation loss. Then, both the training and validation losses decreased in conjunction with the training process, ultimately converging at epoch 17. Beyond epoch 17, the validation loss exhibited fluctuations around the value of 0.036, while the training loss plateaued. This behaviour signifies that the loss had reached a

minimum for the given data and model parameters, and it also indicates that the model's generalisation capability demonstrated comparable performance on both the validation and training datasets. Although the predetermined number of epochs was set at 50, the training was terminated at the 36th epoch due to the activation of early stopping.

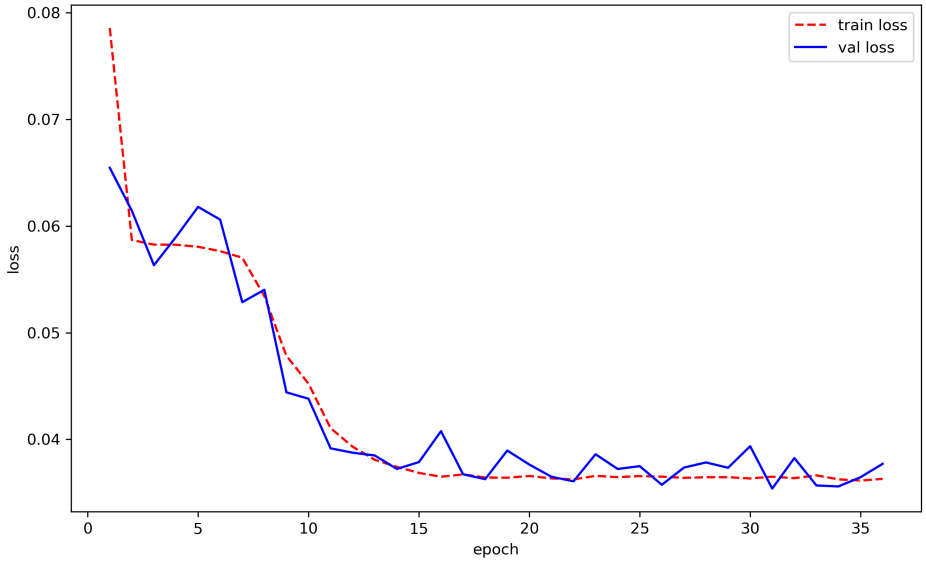
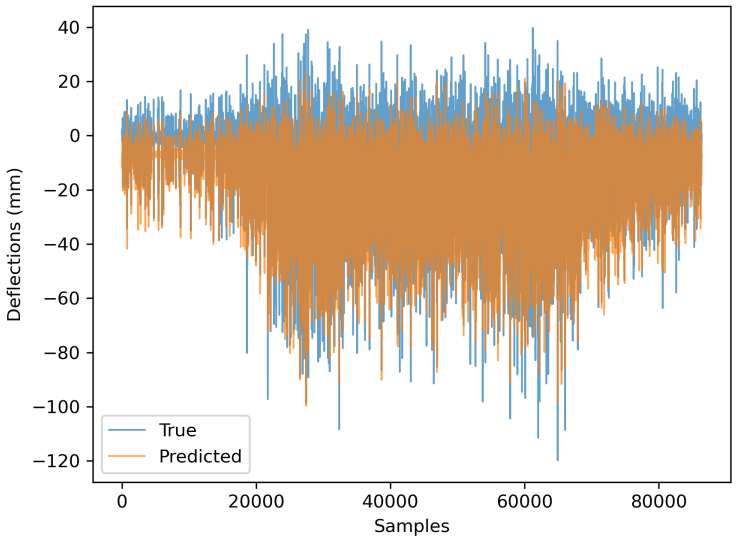
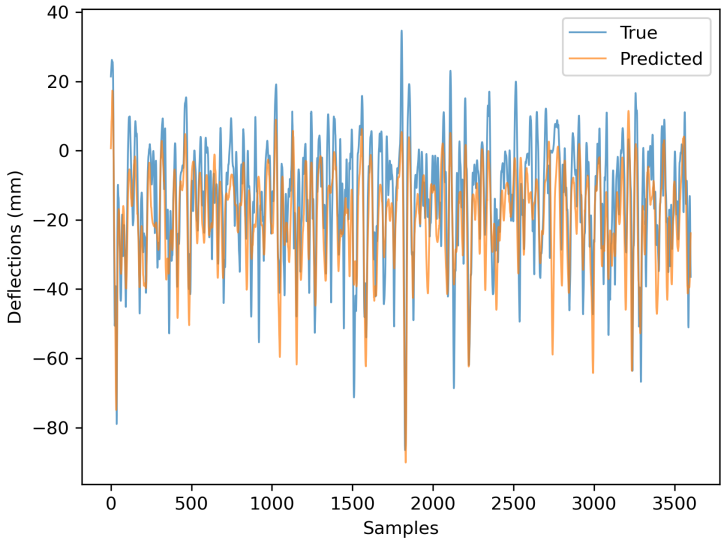


Figure 5.7: Training and validation loss for 10-day simulation data

Following the training and validation phases, the test dataset was employed to assess the predictive performance of the trained model. The predicted deflections and the ground truth from the test dataset are presented in Figure 5.8. Figure 5.8a illustrates the prediction results for the entire test set, comprising 86,400 samples. The MAE is measured at 0.055, and the correlation coefficient R is 0.805, signifying a strong positive relationship between the LSTM predictions and the mid-span deflections. From these results, it can be inferred that the model shows robust generalisation capabilities on unseen data. Figure 5.8b presents the prediction results within a randomly selected one-hour window.



(a) Predictions on test set



(b) Predictions on a 3600s time window in test set

Figure 5.8: Predictions results

The trained LSTM model is subjected to evaluation alongside two baseline models: a linear regression model and a 2-layer Multilayer Perceptron (MLP) model. The MLP model is specifically structured with two hidden fully connected layers, each comprising 32 neurons. The corresponding accuracy metrics are delineated in Table 5.2. Upon examination, it becomes evident that both baseline models show similar performance. By contrast, the proposed LSTM model distinguishes itself by achieving a substantially lower MAE and a higher correlation coefficient R on the test set.

Table 5.2: Model comparison

Model	**Structure**	MAE	MSE	R
Linear regression	$y = W \cdot x + b$	0.078	0.010	0.203
MLP	32 dense neurons (sigmoid) 32 dense neurons (sigmoid) 1 dense neuron (output)	0.078	0.010	0.204
LSTM	32 lstm neurons (sigmoid) 32 dense neurons (sigmoid) 1 dense neuron (output)	0.031	0.005	0.805

Due to the random nature of the training and validation batches in the strategy, the prediction performance is subject to uncertainties. To mitigate this uncertainty, each network model is independently trained, validated, and tested 10 times. The two indicators on the test dataset, namely MAE and correlation coefficient R, were computed independently for 10 iterations and are graphically represented in Figure 5.9. In general, MAE and R possess equivalent capabilities in evaluating the prediction performance of the trained networks. Across the 10 training iterations, 8 models achieved a low MAE around 0.031, while 2 cases showed a relatively high MAE around 0.031. The loss curve corresponding to iteration 6 is depicted in Figure 5.10. It can be observed that the validation loss fluctuates throughout the training process without a discernible decrease, and the training terminated at epoch 11. This phenomenon can be interpreted as the parameters' optimisation reaching and becoming "trapped" at a *local minimum*. A local minimum is a point in the parameter space where the loss

function is lower than in the surrounding area, but not necessarily the lowest possible value. By contrast, Figure 5.7 illustrates that the validation loss gradually reduced to a plateau during training. Similar trends are observed for the 8 models that achieved low MAE in the 10 training iterations. In these instances, the parameter optimisation is believed to have reached the *global minimum*. The global minimum is the point where the loss function reaches its absolute lowest value across the entire parameter space. Finding the global minimum ensures that the neural network has the best possible weights and biases, minimising the error on the training data.

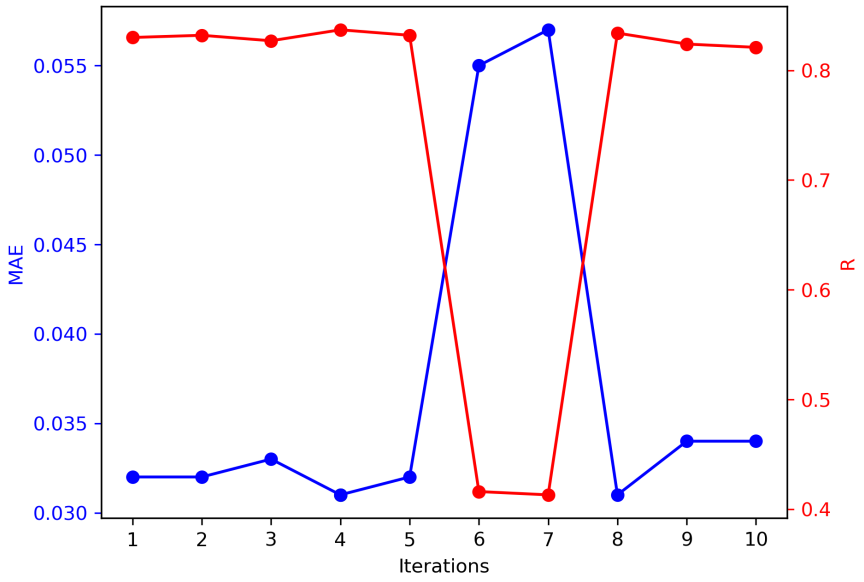


Figure 5.9: MAE and R values in 10 training iterations

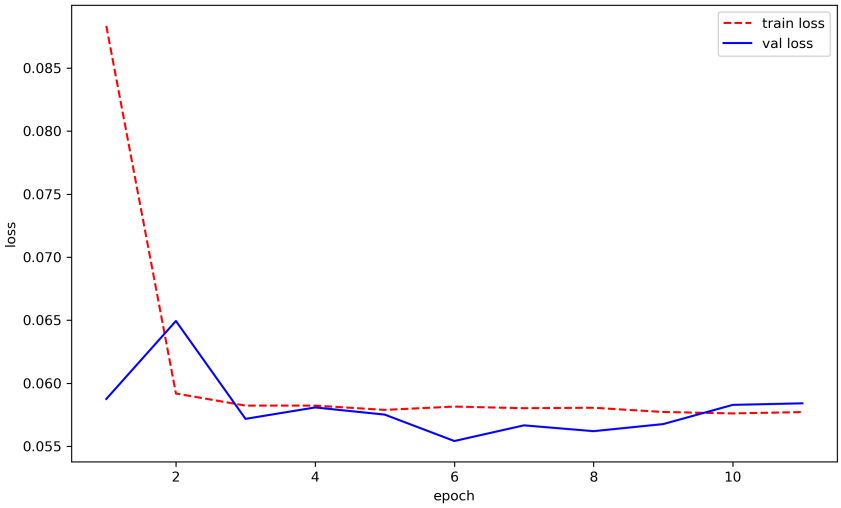


Figure 5.10: Loss curve for high MAE training model

At the current stage, the following conclusions can be drawn:

- The proposed LSTM model demonstrates superior performance in comparison to the baseline MLP and linear regression models.
- The training strategy employed introduces randomly generated training and validation batches. To verify the effect of this randomness on model accuracy, the same LSTM model was trained 10 times. In the majority of instances, the models exhibited similar generalisation capabilities, irrespective of data randomness.
- In certain training instances, the model's accuracy was compromised due to being trapped in a local minimum. This phenomenon can be identified by examining the validation loss curve, where the validation loss fluctuates without a noticeable decrease.

Consequently, in future training processes, a validation curve exhibiting a gradual decreasing trend will be interpreted as reaching the global minimum, and the corresponding prediction results will be considered optimal. Subsequent to these conclusions, a parameter analysis was conducted to investigate the effects of data size and LSTM structures on prediction accuracy.

5.3.4 Parameter analysis

Data size

Simulation data of varying scales were input into the model to assess the impact of data size on the LSTM model. The data size ranged from 10 days to 200 days, with an increment of 10 days at each experiment, sampled at 1Hz, resulting in a range from 864,000 samples to 10,368,000 samples. Apart from the data size, other hyperparameters were held constant during the training, validation, and testing processes. A single-layer LSTM network was employed, configured with a window size of 3600, training batch size of 128, validation batch size of 20, and 32 LSTM neurons. Within this network configuration, the total number of trainable parameters amounted to 6,337.

Each data size was subjected to training, validation, and testing 10 times, and the optimal MAE and correlation coefficient R from each 10-iteration training were utilised to evaluate the model's generalisation. The minimum MAE and maximum R for each data set were graphically represented in Figure 5.11. For datasets encompassing 10 to 90 days of data, both the validation loss and R exhibited significant fluctuations. This behaviour might be attributed to the effect of overfitting on smaller datasets. During the training process, 128 batches were randomly generated from the training set, corresponding to approximately 5.3 days of data. With smaller training datasets, the likelihood of training on the same data increases. Conversely, for datasets containing

over 100 days of data, the two metrics demonstrated relative stability. From these observations, it can be concluded that the overall size of the data set influences the model's generalisation capabilities. Consequently, for the purpose of predicting deflections in our study, the utilisation of datasets smaller than a 90-day length is not recommended.

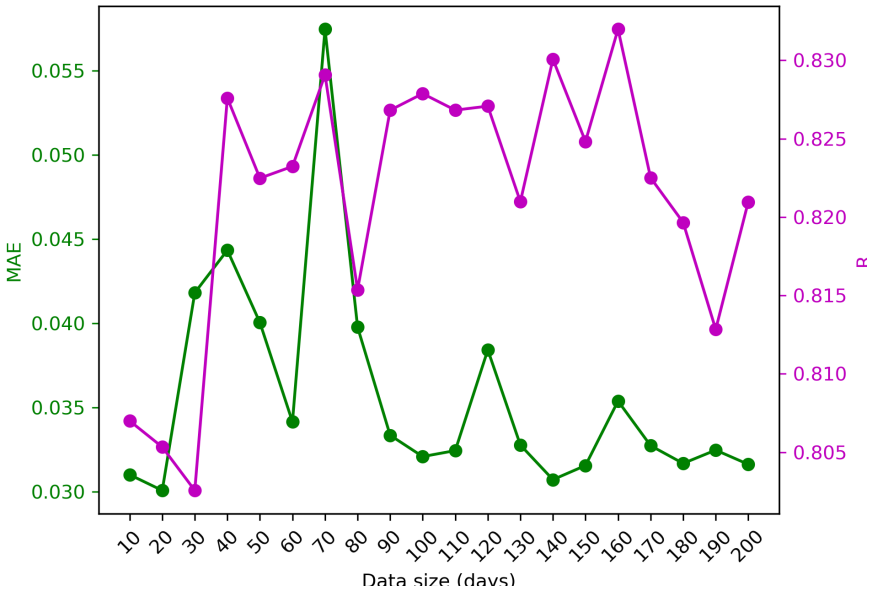


Figure 5.11: Prediction performance for different data size

Among the 200 experiments conducted, the highest correlation coefficient R, valued at 0.831, was attained on the 160-day dataset, and the corresponding prediction results are depicted in Figure 5.12. When contrasted with the results presented in Figure 5.8, it becomes evident that the model exhibited enhanced performance, capturing both the variation trends and extreme values more accurately. Therefore, model generalisation can benefit from a large training dataset even if the training batch size remains constant when the training batches are randomly generated.

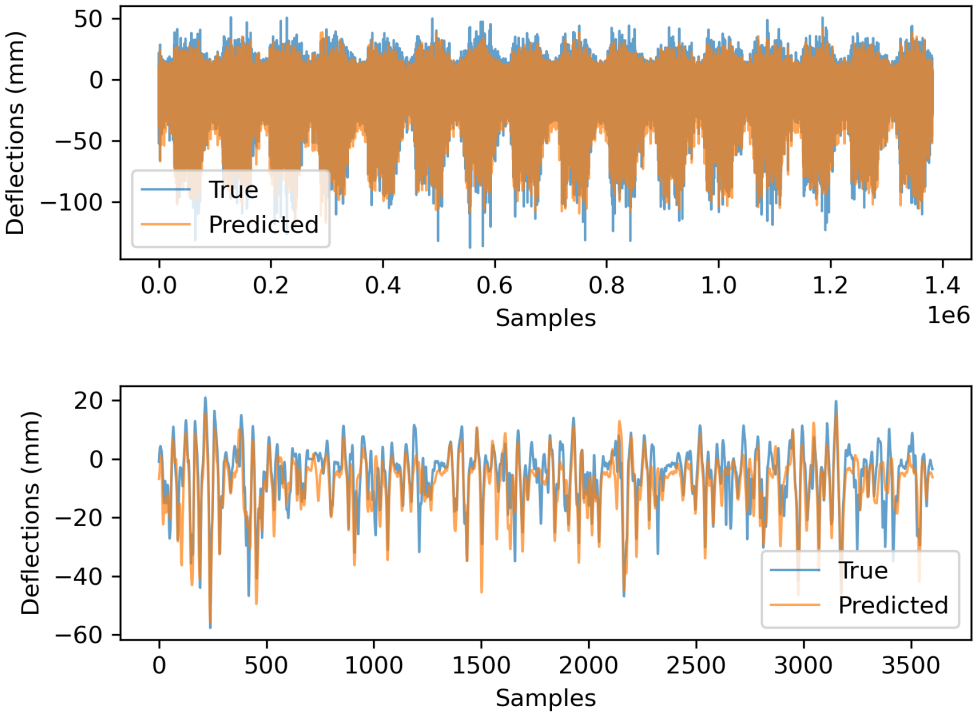


Figure 5.12: Optimal prediction results on 160-day dataset

Number of neurons

The number of neurons within the LSTM layer constitutes one of the pivotal hyperparameters that govern the complexity of the neural network. To investigate the impact of varying neuron numbers on prediction performance, an exploration was conducted using different quantities of neurons to train, validate, and test a network comprising a single LSTM layer. Specifically, the neuron count within the LSTM layer was systematically varied from 16 to 288, resulting in a corresponding increase in trainable parameters from 2,177 to 351,425. Concurrently, the remaining hyperparameters were maintained constant, with a data size encompassing 200-day samples, a window size of 3600, a training batch size fixed to 128, and a validation batch size fixed to 20.

The LSTM network, configured with distinct neuron numbers, underwent a process of training, validation, and testing, conducted 20 times for each configuration. The MAE and maximum correlation coefficient R on the test dataset were computed and subsequently plotted in Figure 5.13. This graphical representation serves to show the relationship between prediction performance and the number of neurons within the LSTM layer, with the MAE being calculated using the normalised data. From the figure, it is evident that the 1-layer LSTM model with 16 neurons shows a distinct difference in both MAE and R compared to other models, failing to converge. This failure indicates poor generalisation, rendering the model incapable of producing meaningful results. The lack of complexity in this model prevents it from identifying patterns within our datasets, a behaviour similar to the compared baseline models in Table 5.2. Further training of the 16-neuron LSTM model 10 times did not yield any converged results.

Figure 5.14 presents the model performance, excluding the 16-neuron model. Generally, as the number of neurons increases, MAE decreases, and R correspondingly rises, signifying an enhancement in prediction performance. The performance peaked when the neuron numbers reached 224, then R decreased (MAE increased) for 256-neuron and 288-neuron models. The highest R reached 0.831. This can be explained by the training difficulties with large-scale models as more parameters needed to be tuned, e.g. 351,425 parameters in the 288-neuron models.

Therefore, we can conclude that increasing the number of neurons in the model structure generally leads to better generalisation. However, if the model becomes overly complex, its performance may suffer. Thus, it is vital to find the peak model performance by carefully tuning the neuron numbers.

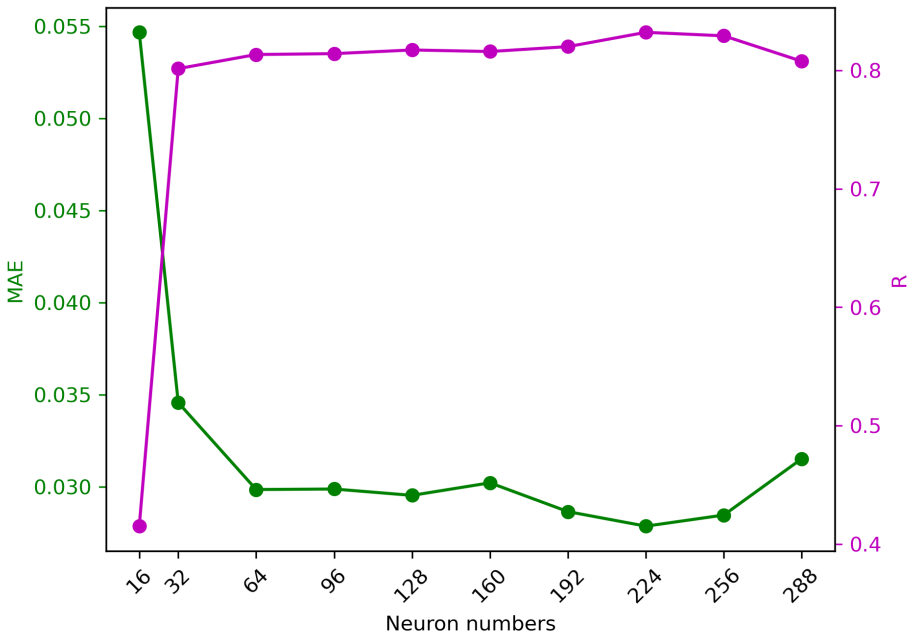


Figure 5.13: Prediction performance with neuron numbers

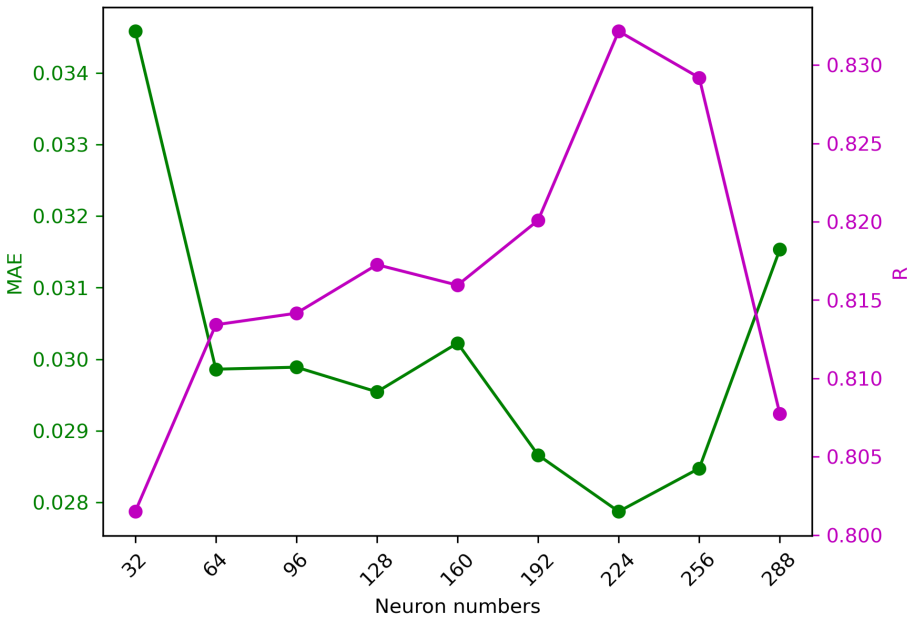


Figure 5.14: Prediction performance with neuron numbers

Number of LSTM layers

In addition to the number of neurons, the quantity of LSTM layers serves as another determinant in shaping the complexity of the network. To investigate the influence of layer numbers on prediction performance, networks comprising 1-LSTM, 2-LSTM, and 3-LSTM layers were constructed, each associated with varying neuron counts within the layers, specifically 32, 64, 96, and 128 neurons. Alongside the variations in neuron numbers and LSTM layers, the remaining hyperparameters were consistently configured, encompassing a data size of 200-day samples, a window size of 3600, and a training batch size of 128 and a validation batch size of 20.

Table 5.3: The optimal MAE and R on test dataset with different layer and neuron numbers

No.	LSTM layer numbers	Neurons in each layer	No. of parameters	Minimal MAE	Maximum R
1	1	32	6,337	0.034	0.802
2	2	32	14,657	0.027	0.842
3	3	32	22,977	0.026	0.847
4	1	64	20,801	0.029	0.813
5	2	64	53,825	0.029	0.820
6	3	64	86,849	0.029	0.846
7	1	96	43,467	0.029	0.814
8	2	96	117,569	0.028	0.831
9	3	96	191,681	0.026	0.837
10	1	128	74,305	0.028	0.817
11	2	128	205,889	0.028	0.835
12	3	128	337,473	0.027	0.837

Multi-layer LSTM networks, with varying numbers of neurons in each layer, were trained, validated, and tested 20 times. The optimal MAE and R on the test dataset were computed to evaluate the prediction performance concerning the number of LSTM layers. These results, calculated using normalised data, are detailed in Table 5.3. By comparing the model performance among different layer numbers with the same neuron numbers in each layer, we learn that the MAE decreases with the growth of LSTM layers, whilst R increases accordingly. Two 3-layer models with 32 and 64

neurons in each layer achieved the highest R, 0.847 and 0.846 respectively. However, the multi-layer models with more neurons in each layer (96 and 128) performed slightly worse. The number of trainable parameters for these four models is much larger than other multi-layer models (32 and 64). For example, the 3-LSTM model with 128 neurons has 337,473 parameters and 3-LSTM model with 32 neurons has 22,977 parameters. Large-scale trainable parameters request large GPU memories and significantly more training time. On the other hand, these models are more difficult to converge.

Therefore, increasing LSTM layers can contribute to the model generalisation capabilities for predicting deflections based on traffic data. However, deep neural networks are computationally more expensive and difficult to converge. In reality, it is suggested to balance prediction performance and computational cost. In this case study, the highest R, 0.847, was achieved by using the 3-LSTM model with 32 neurons in each layer, which is higher than the performance in any single-layer LSTM networks tested in the previous study.

Furthermore, an examination of the 1-LSTM network configured with 64 neurons (comprising 20,801 trainable parameters) reveals that the optimal Mean Absolute Error (MAE) and correlation coefficient R are 0.029 and 0.813, respectively. In contrast, the 2-LSTM model, designed with 32 neurons in each layer and containing only 14,657 parameters, achieved an MAE of 0.026 and an R value of 0.847. Intriguingly, although the 1-LSTM model possesses more parameters than the 2-LSTM, its generalisation capability is found to be inferior. This phenomenon is also observed in the comparison between the 1-LSTM model with 128 neurons and the 2-LSTM model with 64 neurons. Consequently, when the objective is to augment the scale of the network to enhance prediction performance, the preferred strategy appears to be an increase in LSTM layers rather than an expansion in the number of neurons, as the former demonstrates higher efficiency.

5.4 Summary

In this section, Long-Short Term Memory (LSTM) networks were employed to predict vehicle-induced deflections under stochastic traffic conditions, utilising simulated traffic flows generated by the strategy introduced in Chapter 4.

Initially, the simulated discrete vehicle flow data, analogous to data sampled from DWIM systems, were reconstructed into a time-series format, consistent with the structural response data format. Subsequently, the data were partitioned into training, validation, and test sets, and normalised to a uniform range. Following this data preparation, an initial LSTM network model with pre-established hyper-parameters was trained. The generalisation performance of this model was compared with two baseline models, namely, Multilayer perceptron (MLP) and linear regression, with the LSTM model demonstrating superior performance.

Subsequent analyses evaluated the effects of three hyper-parameters, overall data size, neuron numbers, and layer numbers on prediction performance. Generally, an enhancement in prediction performance was observed with the increment of these three hyper-parameters. However, beyond a certain threshold, the prediction performance plateaued. Particularly for neuron numbers and layer numbers, their increase significantly augmented the model complexity, rendering the neural networks more challenging to train, and consequently diminishing the test accuracy.

It was observed that multi-layer networks exhibited superior prediction performance compared to single-layer networks with equivalent trainable parameters. To amplify the network scale for improved prediction performance, the primary strategy is to augment the number of LSTM layers. Following meticulous fine-tuning, the maximum correlation coefficient R between predictions and ground truth reached 0.847, achieved on a 3-layer LSTM network with 32 neurons in each layer.

In conclusion, the proposed LSTM network model exhibits remarkable performance in predicting deflections based on vehicle attributes, including velocity and weight, at the entrance of the bridge. This model holds potential for application to other bridges that may not have a comprehensive SHM system in place. Provided that the traffic distributions and the deflection influence line are obtained, both the simulation strategy and the proposed LSTM model can be employed to simulate vehicle-induced deflections at any specified location. Such an approach can prove invaluable in evaluating the health status of a bridge during its operational stage, thereby contributing to more informed and effective maintenance and management practices.

Chapter 6

Temperature field analysis

6.1 Backgrounds

Temperature stands as a significant external load for bridge structures, alongside vehicle loads, particularly during the operational stage of long-span bridges (Catbas et al., 2017). The temperature experienced by a long-span bridge is influenced by a multitude of factors, encompassing the intensity of solar radiation, ambient temperature, wind, humidity, material properties, and the structural design of the bridge itself (Xia, Zhang, Tian, & Zhang, 2017). Recognising the critical role of temperature loads, thermometers are strategically installed on various components of a bridge to monitor both ambient temperatures and variations in structural temperature. Apart from the ambient temperature, the structural temperature contributes to the secondary stresses directly.

There have been several studies investigating thermal effects. For example, Y. Deng, Li, Liu, and Chen (2018) employed 1-year thermal data on a flat steel box girder to examine the temperature distributions. The study concluded that temperature differences in both transverse and vertical directions should be considered in the girder's cross-section. Similarly, research employing the block maxima approach to derive extreme temperature differences was conducted based on extensive long-term on-site monitoring data (Ding, Zhou, Li, & Wang, 2012). Furthermore, Zhou, Yi, Chen, and Zhang (2017) used the generalised Pareto model to produce the extreme

temperature gradient of girder sections. Subsequently, model parameters were updated using a bayesian estimation method based on historical data and live monitoring data. Notably, these studies concentrate on the temperature distributions within the box girder, investigations on other parts of the bridge are rarely reported. Bridge towers, for example, also exhibit sensitivity to thermal loads (X. Xu, Xu, Ren, & Huang, 2021). Consequently, for cable-stayed bridges, a comprehensive study into the temperature field on the girder, tower, and cables is needed.

In this Chapter, an investigation into the temperature field of the Queensferry Crossing (QC) is conducted, utilising data from temperature sensors installed on the structure. The analysis commences with a detailed description of the temperature sensors' layout on the QC, grounded in documents provided by Transport Scotland. Given the substantial quantity of sensors, the finite element model of the QC is employed to locate critical locations where large moments are observed. Subsequently, temperature sensors situated at these vital locations are selected for further inspections.

6.2 Queensferry Crossing temperature sensors

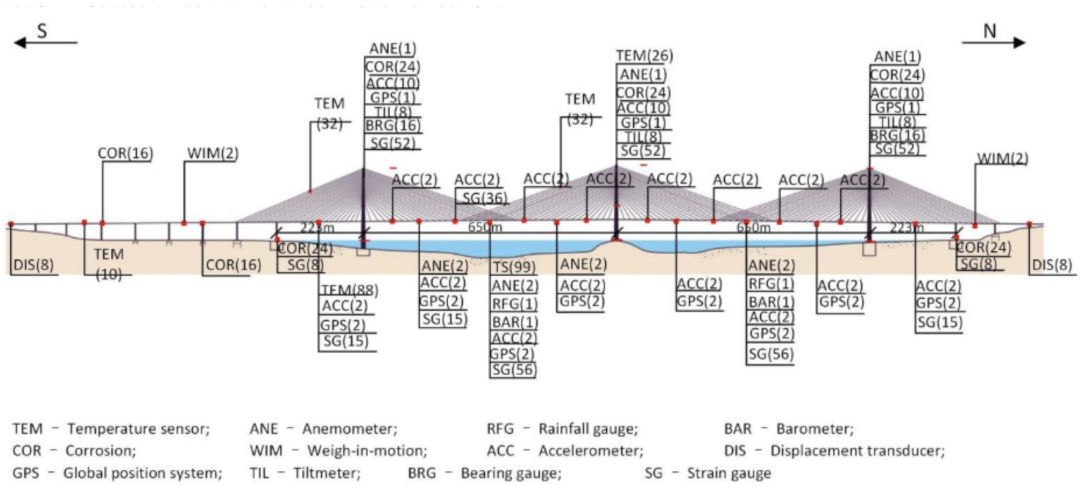


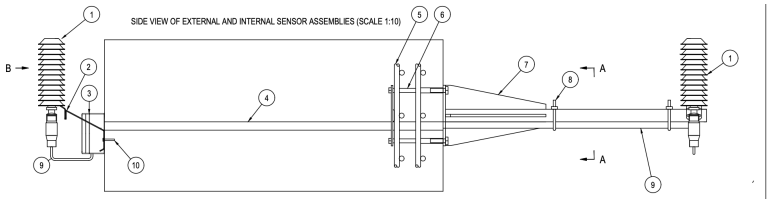
Figure 6.1: Sensor layout on the Queensferry Crossing

Figure 6.1 provides a comprehensive overview of the various types and locations of sensors deployed on the Queensferry Crossing (QC). In total, there are 357 temperature sensors installed across the structure. The detailed distribution of these sensors, categorised by different bridge components, is enumerated in Table 6.1.

Table 6.1: Temperature sensors

Serial	Temperature sensors location	Number of sensors
TMU-6	Air temperature (ambient)	13
TMU-7	Asphalt temperature	40
TMU-4	Concrete deck temperature	70
TMU-3	Concrete tower temperature	20
TMU-5	Stay cable temperature	56
TMU-2	Steel deck temperature	158
	Total	357

(a) Ambient temperature sensor

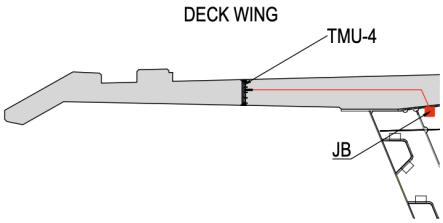


(b) Ambient temperature sensor layout

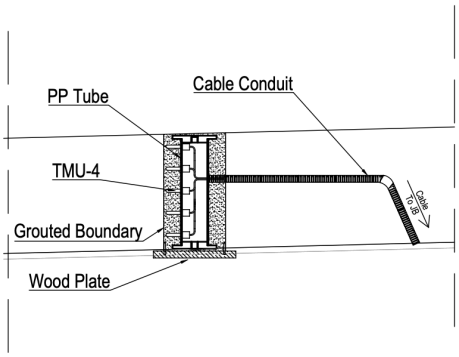
Figure 6.2: Ambient sensor TMU-6

Figure 6.2 shows the placement of ambient temperature sensors on the QC. The air temperature is gauged using the device depicted in Figure 6.2a. These sensors were positioned on the middle tower and on the decks at two distinct locations. Figure 6.3a provides a detailed view of the installation location for the concrete

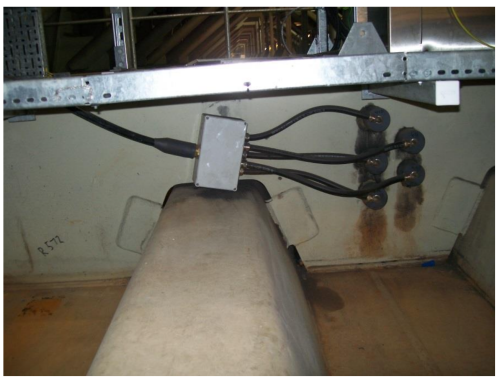
deck temperature sensors, as seen in the cross-section of the concrete deck. At the designated installation site, five thermal sensors were installed at uniform intervals across the depth of the concrete deck. Concurrently, Figure 6.3b describes the layout of the steel deck temperature sensor.



(a) Concrete deck temperature sensors location



(b) Concrete deck temperature sensors layout



(c) Steel deck temperature sensors

Figure 6.3: Deck temperature sensors layout TMU-4 & TMU-2

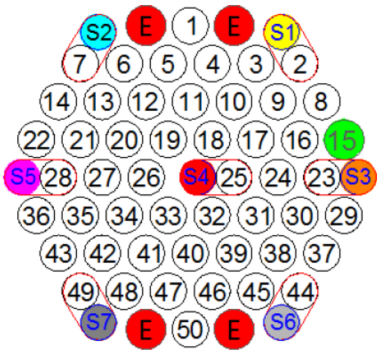


Figure 6.4: Cable temperature sensors (S1-S7) TMU-5

Figure 6.4 shows the layout of the cable temperature sensors, measuring the temperatures inside the stay cables at the top and bottom of the longest cable adjacent to the centre tower and the deck. These sensors are encased within an HDPE pipe and were installed by the stay cable manufacturer. In the context of concrete tower thermal sensors (TMU-3), these were positioned at the mid-tower, equidistant between the deck level and the apex of the tower. Figure 6.5 provides a detailed depiction of the thermal sensors' layout on the cross-section of the tower. A total of five sensors were installed at the midpoint of each side of the tower, as indicated by the red dots.

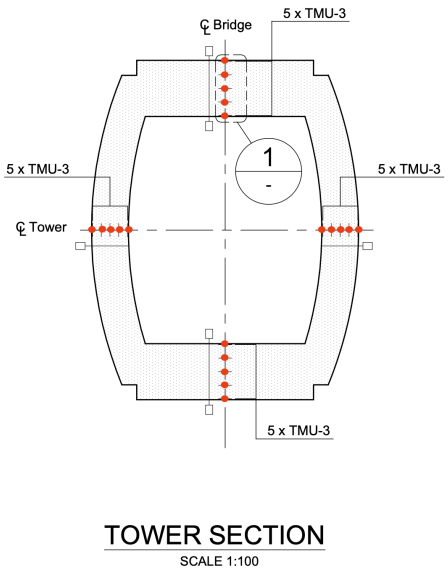


Figure 6.5: Tower temperature sensors TMU-3

In conclusion, the placement of temperature sensors on the QC was meticulously planned during the design stage. Both concrete deck and tower temperature sensors were installed prior to the concrete pour, thereby ensuring the capability to provide precise measurements of concrete temperatures. However, it is vital to recognise that, despite the 120-year design lifespan of the bridge itself, the electronic sensors are unlikely to endure for that duration. As the bridge continues to age, sensor malfunctions are anticipated to occur. Consequently, conducting analyses on these sensors during the earlier stages of bridge operation is deemed preferable.

6.3 Finite element analytics

Owing to the extensive scale of the bridge and the substantial number of thermal sensors installed, identifying the critical sections where thermal loads significantly influence the bridge's responses becomes paramount. During the design stage, sections experiencing maximum moments are typically selected as critical, guiding the design process. Consequently, it is advisable to determine the monitored sections for the proposed temperature field analysis as those corresponding to the maximum moments under the designed loads.

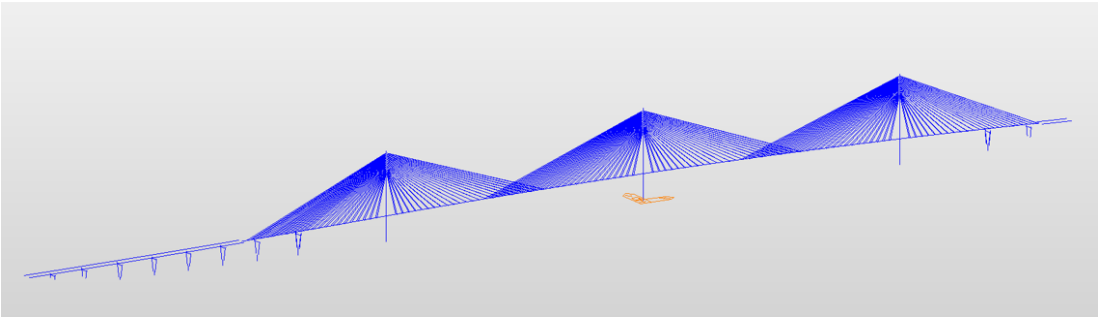


Figure 6.6: Finite element model of the QC

A MIDAS finite element model was developed by Arup at design stage. The model consists of 1793 nodes and 1405 elements. In order to locate critical sections, vehicle load models specified in the *Eurocode EN1991-2:2003 - road traffic actions and other actions specifically for road bridges* (BSI, 2004) were employed as the live load. Through the calculation and discussion, the load model 1 (LM1) in clause *4.3.2* was selected, depicted in Figure 6.7. A double-axle load, named the Tandem system, was applied in each traffic lane (including hard shoulders) with a uniformly distributed load (UDL).

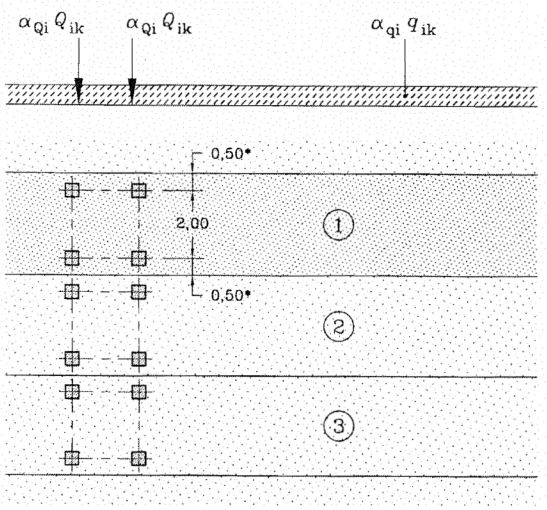
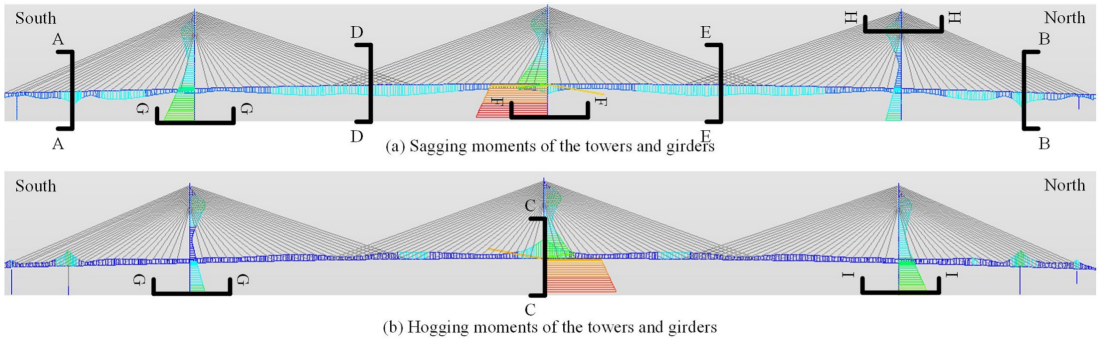


Figure 6.7: Finite element model of the QC (BSI, 2004)

(1) Lane 1: $Q_{1k} = 300kN$; $q_{1k} = 9kN/m^2$
(2) Lane 2: $Q_{2k} = 200kN$; $q_{2k} = 2.5kN/m^2$
(3) Lane 3: $Q_{3k} = 100kN$; $q_{3k} = 9kN/m^2$
Tandem axle spacing=1.2m

Under Loading Model 1 (LM1), the structural analysis of the QC was conducted utilising the MIDAS software. The resulting sagging and hogging moments are depicted in Figure 6.8. For the girder sections, the maximum sagging moments are observed at the south and north auxiliary pier sections (i.e., Sections A-A and B-B), while the centre tower section (i.e., Section C-C) experiences the maximum hogging moment. Additionally, the sagging moments at the south and north mid-span, denoted as Sections D-D and E-E, represent the secondary maximum values.

In the context of tower sections, the centre tower bottom section, referred to as Section F-F, bears both the maximum sagging and hogging moments. Generally, the moments in the south and north towers are relatively smaller compared to those in the centre tower. Specifically, for the south tower, the bottom section (i.e., Section G-G) endures the maximum sagging and hogging moments, while in the north tower, the maximum sagging moment occurs at the upper tower section (i.e., Section H-H),



(a) Sagging moments of the towers and girders

(b) Hogging moments of the towers and girders

Figure 6.8: Sagging and hogging moments of the QC towers and girders

Table 6.2: Sagging and hogging moments of QC (negative for hogging)

Critical section	Location	Moment ($10^7 kN \cdot mm$)
A-A	South auxiliary pier (S1)	5.30
B-B	North auxiliary pier (N1)	6.01
C-C	Centre tower - deck level	-8.18
D-D	South mid-span	4.91
E-E	North mid-span	4.62
F-F	Centre tower - bottom	28.70
G-G	South tower - bottom	6.34
H-H	North tower - middle	-12.84
I-I	North tower - bottom	6.47

and the maximum hogging moment is located at the bottom (i.e., Section I-I). The detailed moments of these critical sections are summarised in Table 6.2. Therefore, we chose sensor groups installed at the south mid-span to investigate the girder temperature field.

6.4 Temperature data

Based on the previously determined critical sections on the QC, several sensor groups were selected for the girder, cable and tower. In this section, the detailed temperature field analysis is carried out based on these three parts of the bridge.

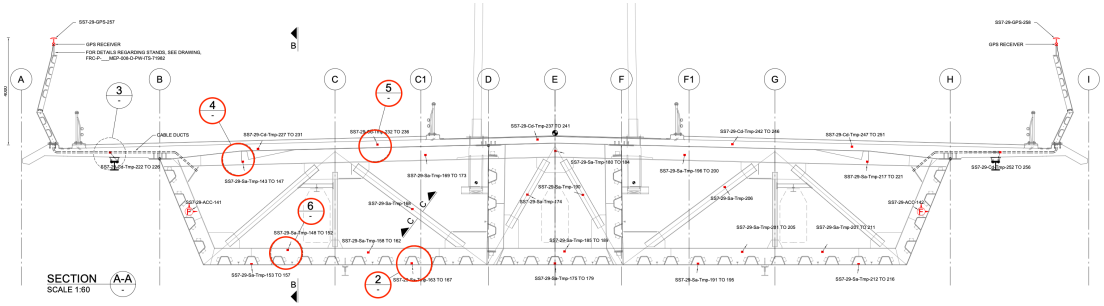
6.4.1 Girder sections

In the context of cable-stayed bridges, thermal effects are influenced by two types of structural temperatures: uniform temperature and temperature difference. Many thermal sensors are installed on various girder sections, including the soffit, deck chord, concrete chord, and bottom chord, to measure the temperature fields. The uniform girder temperature plays a crucial role in determining the maximum displacement of girder ends, which subsequently informs the selection of appropriate expansion joints and bearings. Additionally, the temperature difference within the girder section can be affected by factors such as solar radiation. According to existing literature, the temperature difference in the girder section is identified as a principal load contributing to the generation of thermal effects (X. Xu et al., 2019).

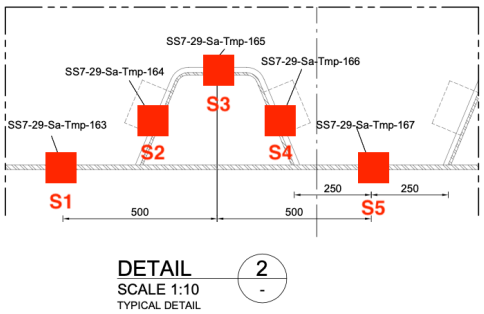
Figure 6.9 illustrates the configuration of thermal sensors on the south mid-span girder section of the Queensferry Crossing (QC), and the corresponding sensor IDs, essential for locating the thermal data within the Structural Health Monitoring (SHM) database, are enumerated in Table 6.3. The data collected by these 20 sensors over the course of one year (from July 2020 to June 2021) has been examined and will be discussed in the subsequent sections.

Soffit

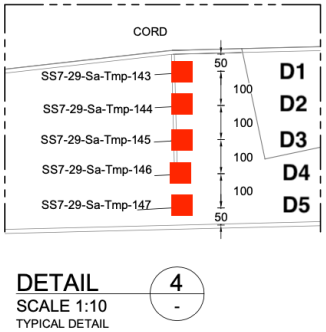
The soffit of a box girder is the underside or the bottom surface of the girder. Thermal sensors (S1-S5) were affixed to the internal surface of the soffit. The corresponding temperature data are plotted in Figure 6.10a. It is noteworthy that measurements for January and April 2021 are absent, owing to the evaluation and maintenance of the SHM system during that period. Subsequently, the temperature differences between each pair of sensors were computed, and the maxima of these differences are depicted in Figure 6.10b. From the analysis, it can be observed that the maximum



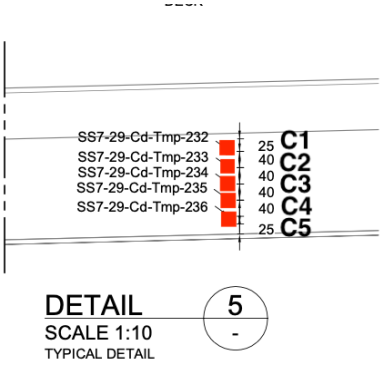
(a) South midspan cross-section



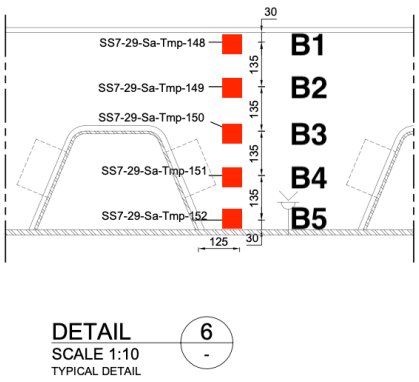
(b) Soffit



(c) Deck chord



(d) Concrete deck



(e) Bottom chord

Figure 6.9: Thermal sensors on the QC south mid-span girder section

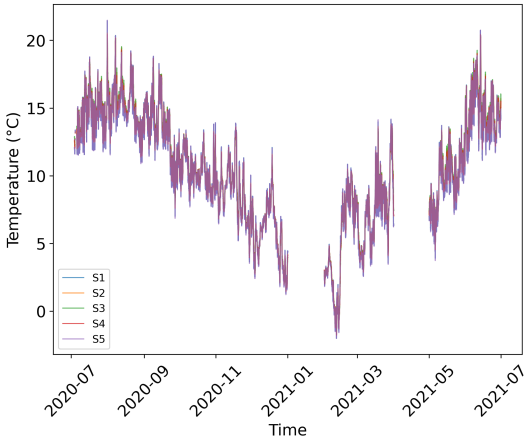
Table 6.3: Deck temperature sensors ID

Posistion	Measurement Type	Tag No.	Sensor ID
Soffit	Steel surface temperature	S1	306
		S2	307
		S3	308
		S4	309
		S5	310
Deck chord	Steel surface temperature	D1	286
		D2	287
		D3	288
		D4	289
		D5	290
Concrete chord	Concrete deck temperature	C1	344
		C2	345
		C3	346
		C4	347
		C5	348
Bottom chord	Steel surface temperature	B1	291
		B2	292
		B3	293
		B4	294
		B5	295

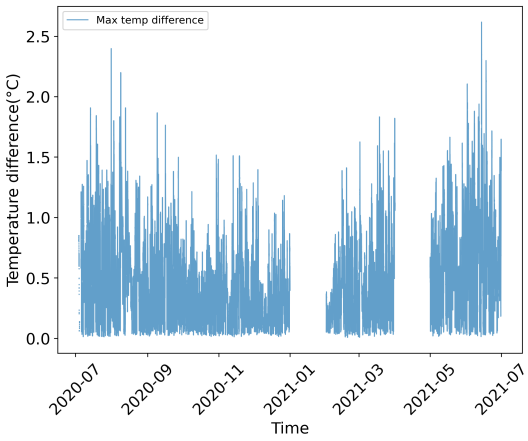
temperature difference of the soffit approximates 2.5°C. Furthermore, 81.49% of the temperature differences are observed to be below 2.0°C, and Figure 6.10c illustrates that the majority of differences are concentrated around 0 to 2.0°C. This relatively small temperature difference on the girder soffit can be attributed to the high thermal conductivity of the materials used.

Deck chord

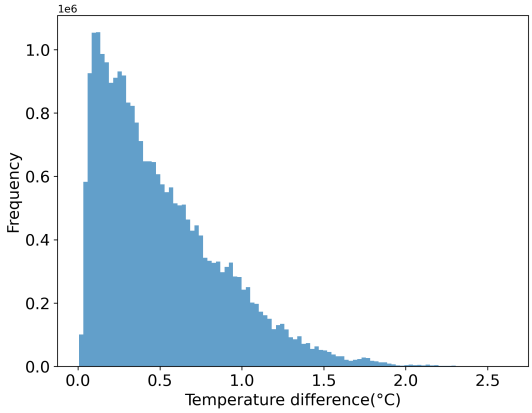
Five thermal sensors were positioned on the deck chord along the vertical direction, as depicted in Figure 6.9c. The temperature data, spanning a one-year period, are illustrated in Figure 6.11a. An analysis of Figure 6.11b and Figure 6.11c reveals that the maximum temperature difference on the deck chord is approximately 3.5°C, with the majority of the temperature differences, constituting 82%, falling within



(a) Measured soffit temperature from 5 sensors



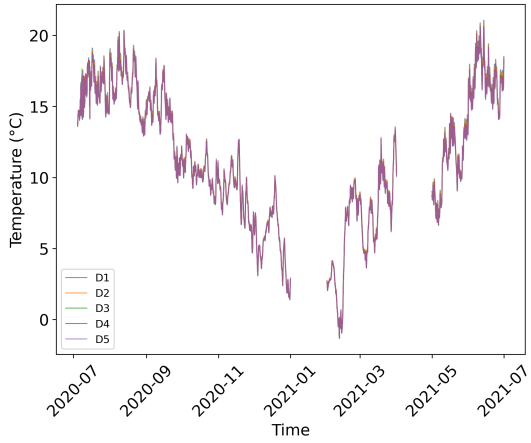
(b) The maximum temperature difference of 5 sensors



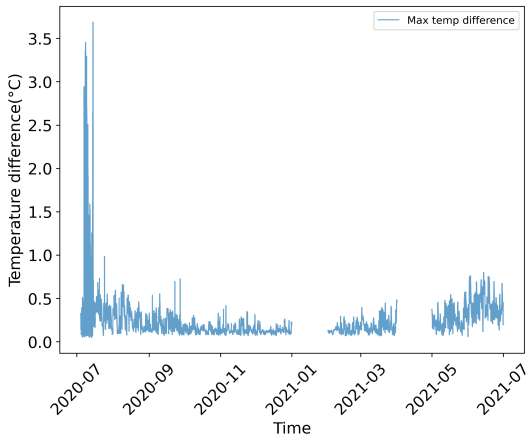
(c) Distribution of the maximum temperature difference

Figure 6.10: Temperature features of the QC soffit

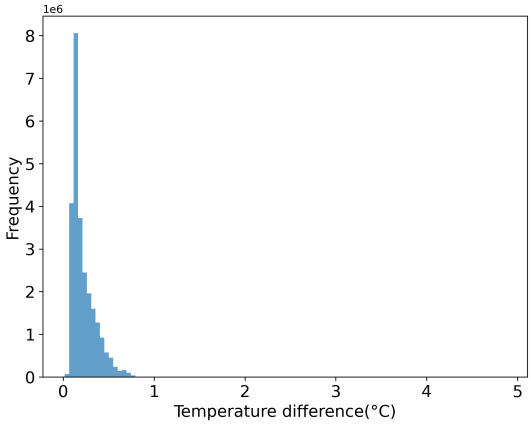
the range of 0 to 1°C. This distribution of temperature difference on the deck chord is observed to be much narrower in comparison to that of the soffit. Such a phenomenon can be attributed to the minimal thickness of the deck chord and the high thermal conductivity inherent to the steel material. Additionally, it is worth noting that the distance between sensors on the deck chord is 100mm, a dimension significantly smaller than the 250mm distance between sensors on the soffit. This spatial configuration further contributes to the observed temperature characteristics on the deck chord.



(a) Measured deck chord temperature from 5 sensors



(b) The maximum temperature difference of 5 sensors

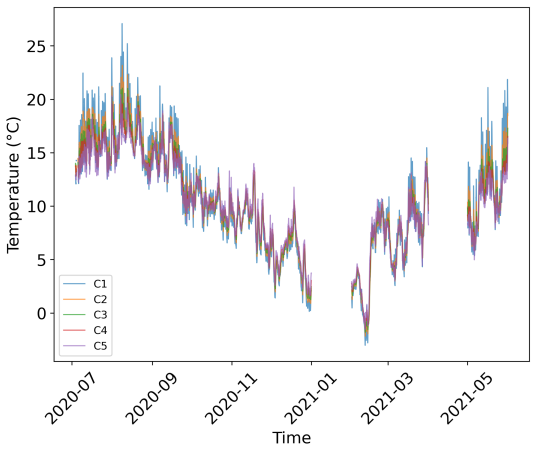


(c) Distribution of the maximum temperature difference

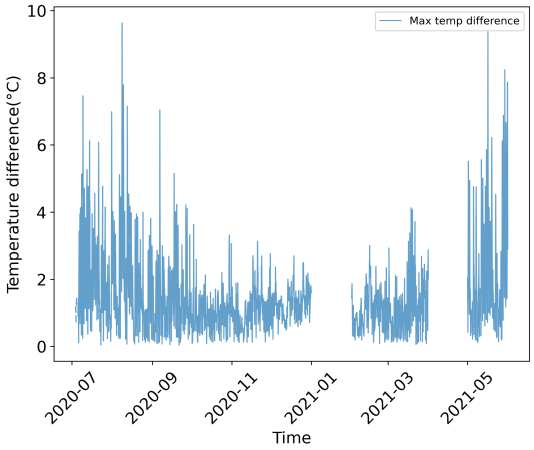
Figure 6.11: Temperature features of the deck chord

Concrete deck

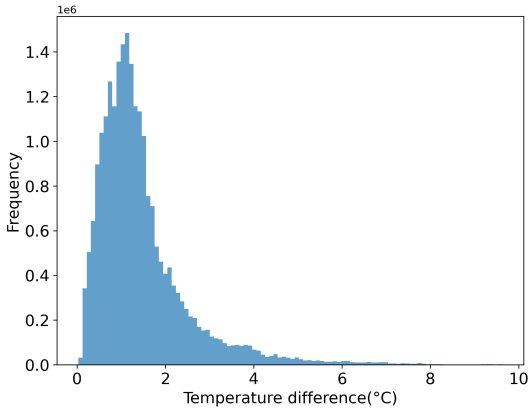
A parallel analysis was conducted for the concrete chord, the results of which are depicted in Figure 6.12. The maximum temperature difference observed reached nearly $10°C$, a value substantially higher than the records for other steel elements. The majority of the temperature difference is found to span from 0 to $8°C$, 73%.



(a) Measured concrete deck temperature from 5 sensors



(b) The maximum temperature difference of 5 sensors



(c) Distribution of the maximum temperature difference

Figure 6.12: Temperature features of the concrete deck

This relatively large temperature difference can be attributed to the low thermal conductivity of concrete. This observation holds true despite the fact that the distance between sensors on the concrete chord, ranging from 25 to 40mm, is considerably smaller than that of the steel deck chord, which is 100mm.

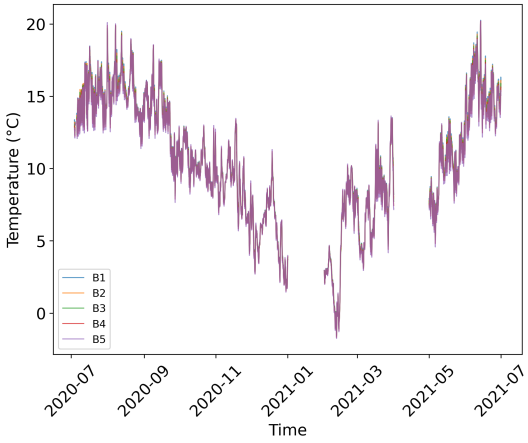
Bottom chord

Similarly, the temperature variations of the bottom chord, along with the corresponding analysis, are illustrated in Figure 6.13. The maximum temperature recorded is below $20°C$, and the difference predominantly ranges from 0 to $1.5°C$, constituting 81% of the observations. The relatively small temperature difference on the deck bottom chord can be primarily attributed to the high thermal conductivity of the steel and the minimal thickness of the bottom chord.

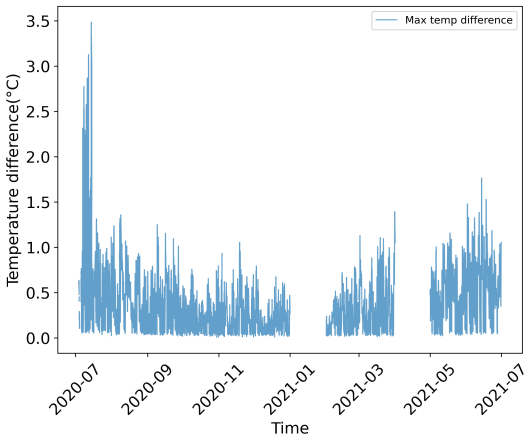
It is noticeable that in both Figure 6.11 and Figure 6.13, the temperature differences in early July 2020 are significantly higher than those observed throughout the rest of the year. This can be attributed to the relatively strong solar radiation during summer, which causes the surface of the steel to heat up more quickly. A similar phenomenon is observed in Figure 6.12, where the temperature difference between October and May is smaller.

When comparing Figure 6.9c and Figure 6.9e, we discovered that sensors are vertically installed across the section for both deck top and bottom chord at similar distances, 100mm and 135mm respectively. By comparison, the deck top chord temperature is slightly higher than the bottom chord. An example of 5-day data is given in Figure 6.14. This discrepancy can be explained by the solar radiation effect on the deck top chord.

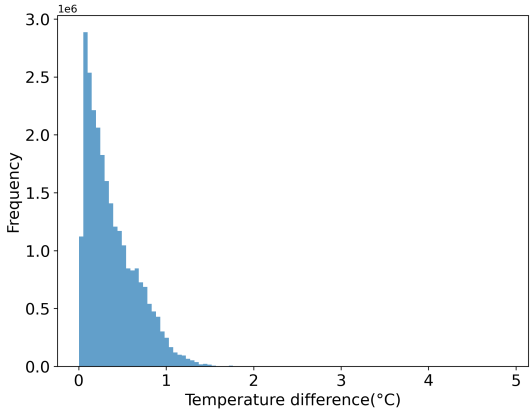
In Figure 6.14, the concrete deck temperature is also depicted. When contrasted with the deck top chord (steel) plate, the concrete deck variation exhibits a more pronounced daily seasonality. Moreover, the daily variation range of the concrete is substantially larger than that of the steel. As illustrated in Figure 6.9a, the deck top chord (steel) is situated beneath the concrete deck. Consequently, the concrete deck is more susceptible to the influences of solar radiation and ambient temperature.



(a) Measured bottom chord temperature from 5 sensors



(b) The maximum temperature difference of 5 sensors



(c) Distribution of the maximum temperature difference

Figure 6.13: Temperature features of the bottom chord

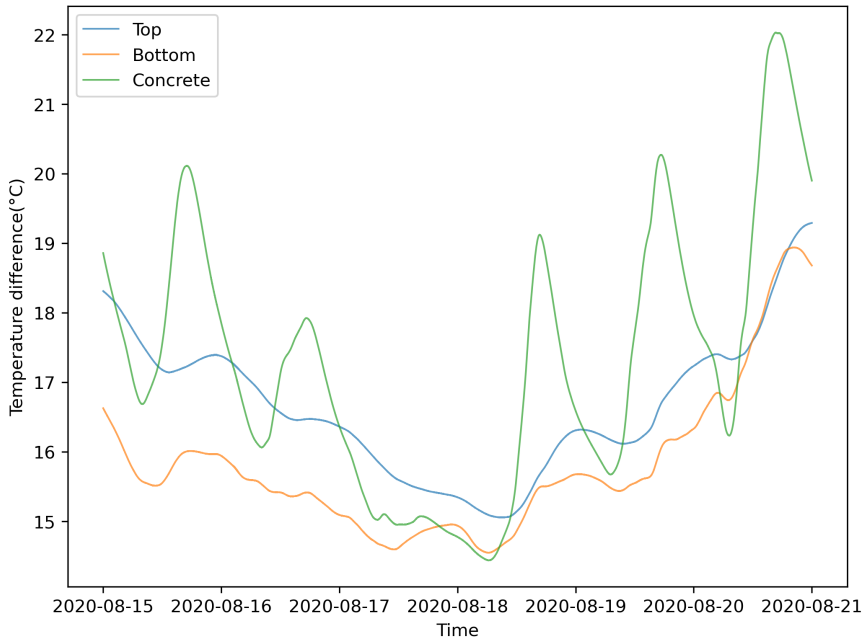
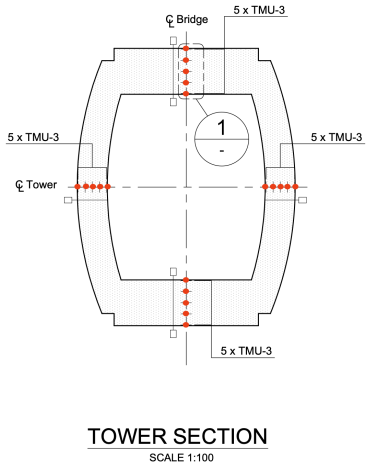
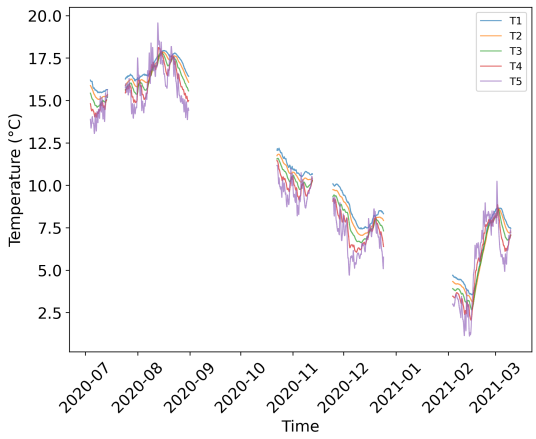


Figure 6.14: Top and bottom chord temperature

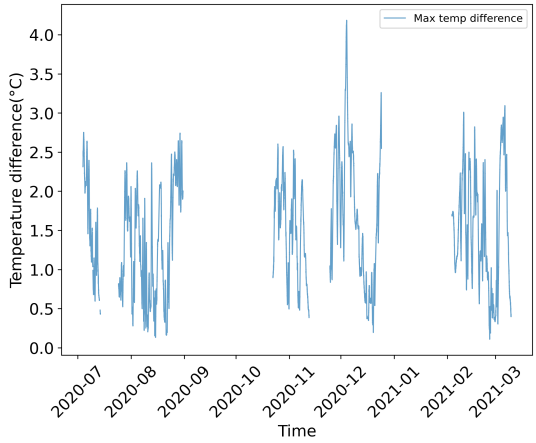
6.4.2 Towers



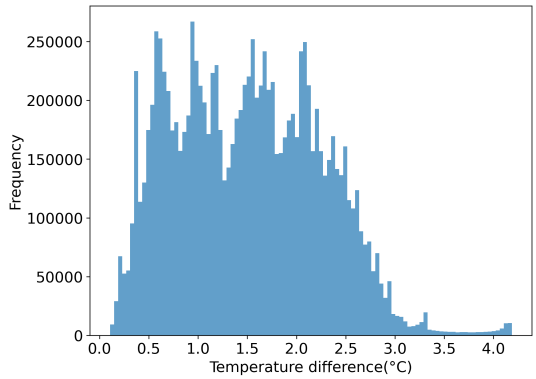
(a) Tower temperature sensors TMU-3



(b) Measured tower temperature from 5 sensors



(c) The maximum temperature difference of 5 sensors



(d) Distribution of the maximum temperature difference

Figure 6.15: Temperature features of the centre tower

One-year data are illustrated in Figure 6.15b, based on the five sensors depicted in Figure 6.15a. The issue of missing data is more pronounced for tower temperature sensors, with 64% of the data absent over a one-year span. The maximum temperature difference is represented in Figure 6.15c, and the distribution of this difference is displayed in Figure 6.15d, ranging from 0 to 5°C. Additionally, it is observed that the temperature measured by the thermal sensor near the outer tower shell (T5) exhibits

strong seasonality in comparison to the temperature recorded by the sensors on the inner side (T1), as depicted in Figure 6.16. This phenomenon can be attributed to the substantial thickness of the tower section, rendering the inner side less sensitive to variations in ambient temperature and solar radiation.

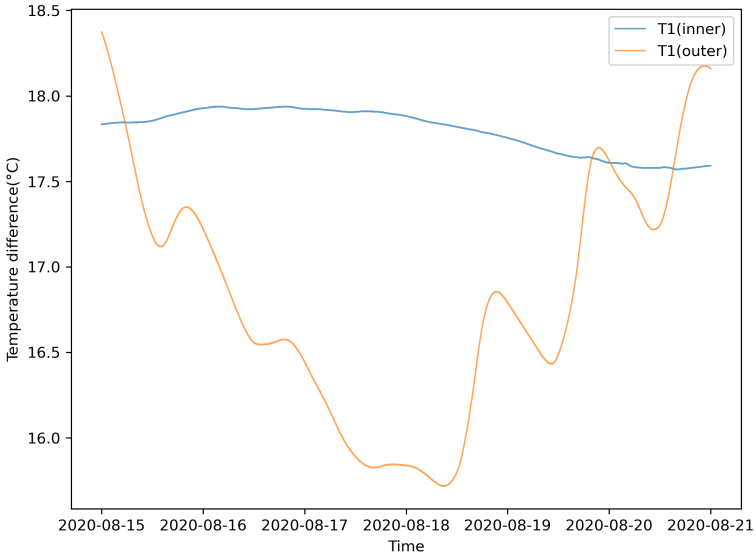
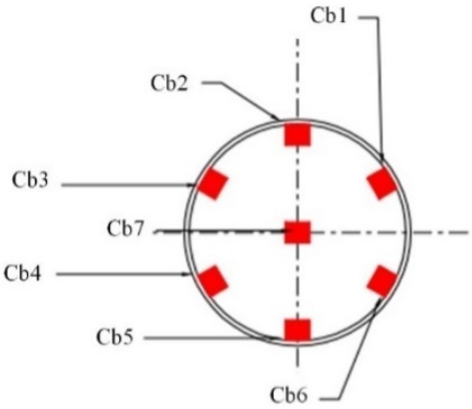


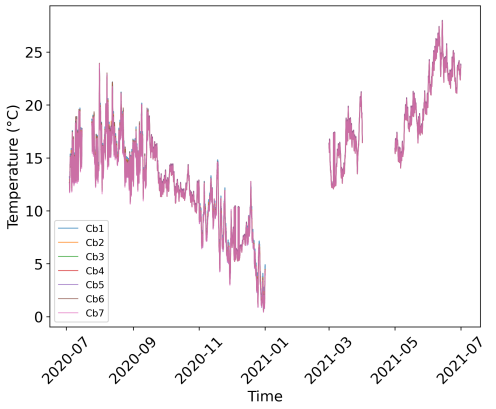
Figure 6.16: Temperature at inner and outer side of the mid tower

6.4.3 Cables

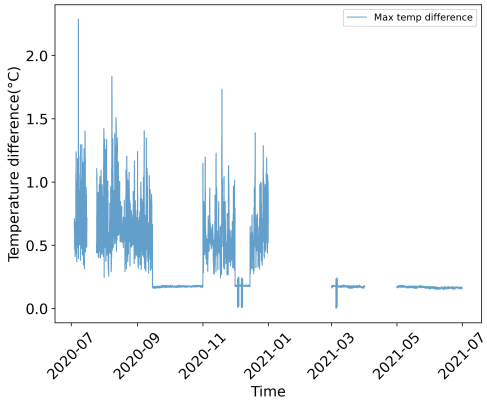
The layout of thermal sensors inside the cables is depicted in Figure 6.17a. Six sensors are installed on the internal surface of the cables, and one is positioned at the centre of the cable section. The distribution of the temperature difference is illustrated in Figure 6.17c, ranging from 0 to $1°C$, a relatively narrow range. An extremely small temperature difference is observed, even though one thermal sensor is situated at the centre of the cable section. This relatively minor temperature difference within the cable section can be attributed to the small scale of the cables and the high thermal conductivity of the steel material.



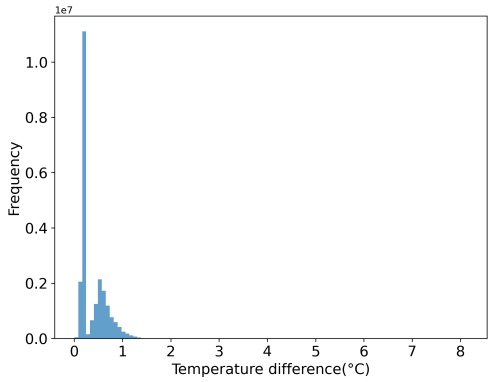
(a) Cable temperature sensors TMU-5



(b) Measured cable temperature from 7 sensors



(c) The maximum temperature difference of 7 sensors



(d) Distribution of the maximum temperature difference

Figure 6.17: Temperature features of the cable

In summary, we examined the temperature variations for the south-mid span box girder, including the soffit, deck bottom chord, deck top chord and concrete deck, the centre tower section and the stay cable. The temperature difference in the vertical direction was compared for each bridge component. The findings reveal that steel components exhibit relatively small temperature differences, attributable to the plate's thickness and steel's high thermal conductivity. While the temperature difference in a concrete section is larger. Therefore, in the thermal-induced deflections analysis, the temperature differences within a steel section are ignored and for concrete sections, data from all five sensors are used. A notable distinction in temperature variation

between the deck top and bottom chords was observed, with the top chord being directly influenced by solar radiation. Understanding temperature differences is vital as it can generate excessive thermal loads. Next, extreme temperature estimation on different bridge components is performed based on the temperature difference.

6.5 Extreme temperature estimation

In this section, the estimation of extreme temperatures for the Queensferry Crossing over 120 years is conducted utilising one-year SHM data. The vertical temperature difference of the bridge girder Δt_{vg}, the horizontal temperature difference of the top chord Δt_{hg} and the temperature difference on the tower section Δt_w are estimated. There have been several studies on the topic of estimating the extreme environmental factors on bridges. For instance, the Gumbel distribution has been widely employed to estimate extreme wind speeds (Kang, Ko, & Huh, 2015; Pinheiro & Ferrari, 2016). This method involves using recorded annual maximum wind speeds to determine the model and the dispersion in the distribution through a curve-fitting technique (Gumbel, 1958). However, it has been noted that a dataset comprising at least 20 years of annual maximum wind speed is necessary to attain a result with a satisfactory level of reliability (Palutikof, Brabson, Lister, & Adcock, 1999). For the estimation of extreme temperatures in the present study, the generalised Pareto distribution (GPD) was employed. In statistical theory, the GPD is recognised as a family of continuous probability distributions, and it is frequently utilised to model the tails of other distributions.

6.5.1 Generalised Pareto Distribution

The GPD functions by estimating a singular extreme event through the pre-setting of an appropriate threshold value and the selection of specific distribution parameters. In the context of extreme temperatures, if the estimated extreme temperature surpasses the value defined in the relevant design code, it serves as an indication that the design loads may be inadequate. Consequently, the evaluated bridge component (e.g., deck or tower) may be operating in an overloaded condition.

The standard cumulative distribution function (CDF) of the GPD is defined by:

$$G(x;\sigma,\xi) = \begin{cases} 1-(1+\xi\frac{x}{\sigma})^{-1/\xi} & \text{for } \xi \neq 0 \\ 1-exp(-\frac{x}{\sigma}) & \text{for } \xi = 0 \end{cases} \tag{6.1}$$

where σ is the scale parameter, ξ is the shape parameter.

In the context of the Generalised Pareto Distribution (GPD), the threshold is a critical parameter that separates the "extreme" values from the rest of the data. It is the point above which the tail behaviour of the distribution is modelled by the GPD. The choice of the threshold has significant implications for the analysis. If the threshold is too low, it may include values that don't truly represent the extreme behaviour of the distribution. This can lead to a biased estimation of the tail properties and may not accurately represent the risk of very extreme events. However, if it is too high, fewer data points are included in the analysis, which can lead to higher uncertainty and variability in the parameter estimates. In this study, the Mean Excess Function (MEF) is used to determine the threshold using the following process:

1. **Calculate the mean excess** For a given threshold u, the mean excess is the average of the values the threshold:

$$e(u) = \frac{1}{N_u}\sum_{i=1}^{N_u}(x_i - u) = \frac{\xi}{1-\xi}u + \frac{\sigma}{1-\xi} \tag{6.2}$$

   where N_u is the number of observations greater than u and x_i are the observations greater than u.

2. **Plot the MEF** Plot the mean excess $e(u)$ against the threshold u.

3. **Identify the threshold** In the GPD, the mean excess function is linear in the threshold for a suitable choice of threshold. The point where the plot starts to appear linear can be considered as a good threshold.

4. **Estimate parameters** The parameters ξ and σ can be determined by maximum likelihood estimation or Equation 6.2.

Next, the cumulative probability p against a preset exceedance probability Pr within the design life of N years can be calculated as;

$$p = (1 - P_r)^{1/N} \tag{6.3}$$

Finally, the estimated extreme value x_p subjective to the cumulative probability p is:

$$x_p = u_0 + \frac{\sigma}{\xi}\left[\left(\frac{n}{N_u}(1-p)\right)^{-\xi} - 1\right] \tag{6.4}$$

6.5.2 Deck vertical temperature difference

Based on the temperature variation analysis conducted in the previous section, a noticeable temperature difference was observed between the deck top and bottom chord. Figure 6.18 illustrates the daily maximum temperature differences from the south midspan southbound lane for the period from August 2020 to July 2021. Due to missing data, only 304 days of information are available. The temperature difference is calculated using temperatures from the top and bottom chords of the deck.

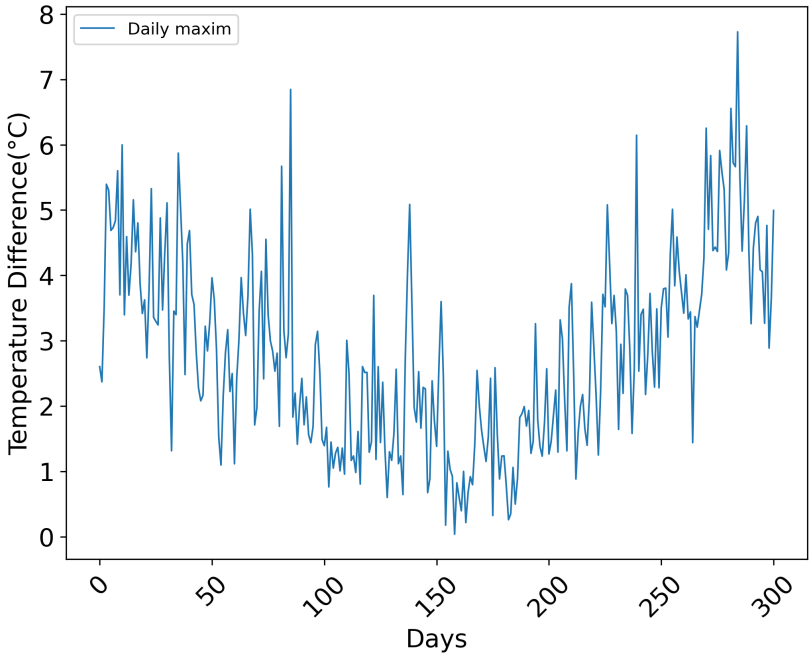


Figure 6.18: Maximum and minimum temperatures

The MEF for the vertical temperature difference in the deck was computed and is illustrated in Figure 6.19. The MEF plot exhibits an approximate linear trend above a threshold of $\mu_0 = 1.18°C$, providing a sufficient number of samples for the GPD to yield reliable results. The shape parameter ξ and the scale parameter σ are determined through maximum likelihood estimation, with values of -0.384 and 2.627, respectively. Figure 6.20 presents the Pareto distribution of the excess above the threshold $\mu_0 = 1.18$ alongside the histogram of the measured temperature difference,

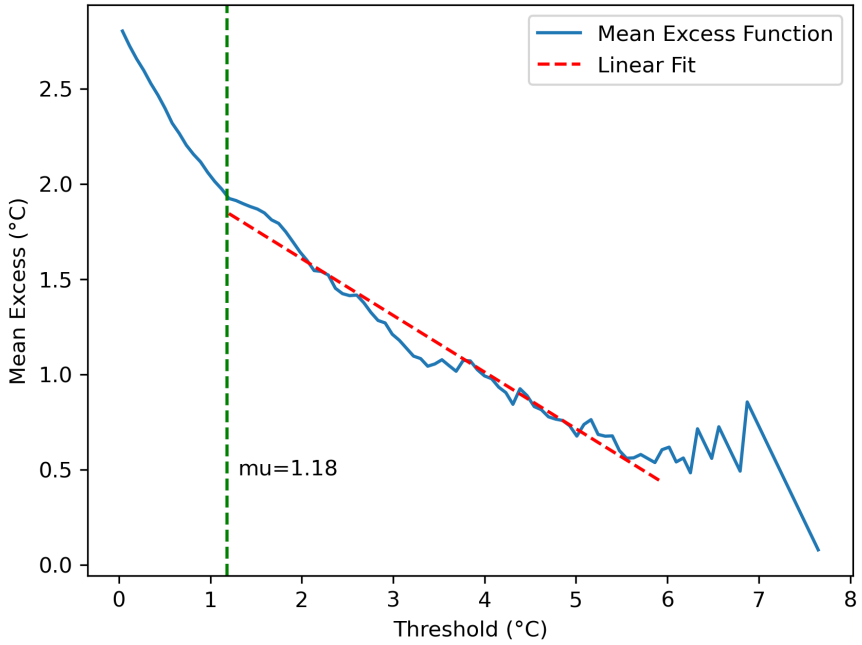


Figure 6.19: Mean excess function for deck vertical temperature difference

demonstrating a strong alignment between the Probability Density Function (PDF) curve and the distribution of the measurements. An exceedance probability $P_r = 5\%$ is set to calculate the cumulative probability p using Equation 6.3. The cumulative probability with a reference period of 120 years is 99.95 % and the estimated maximum temperature difference is 7.66 $^{\circ}C$. In comparison, the British Standard (BS 5400-2) (BSI, 2006) specifies the extreme vertical temperature difference for the girder to be 24$^{\circ}C$, a value significantly larger than the estimation. This discrepancy indicates that the design for the QC girder section is within safe parameters.

6.5.3 Deck horizontal temperature difference

Similarly, five sensors installed on the deck top chord are chosen to investigate the deck horizontal temperature distributions. The layout of these five sensors is depicted in Figure 6.21. Large segment outliers are discovered in sensor H3. Therefore, data from H3 is not used in the following analysis. The daily maximum deck horizontal temperature difference is plotted in Figure 6.22b, reaching 8 $^{\circ}C$. The MEF plot in

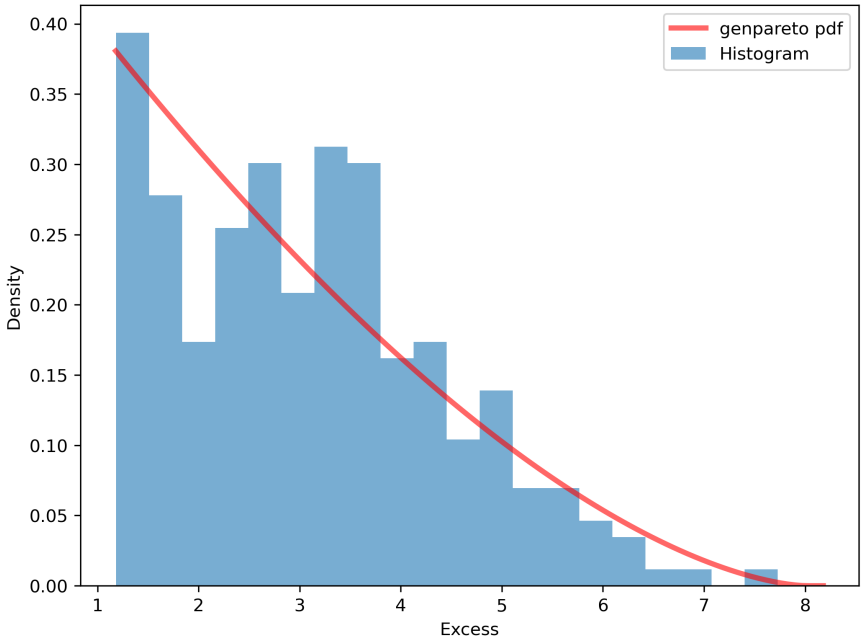


Figure 6.20: Pareto distribution for deck vertical temperature difference

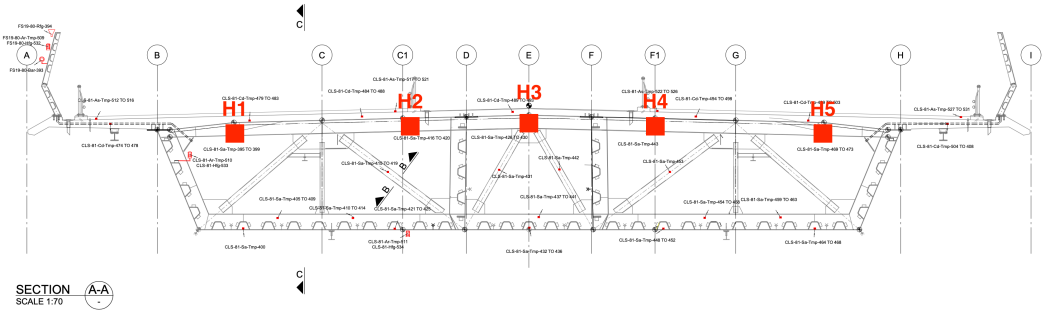
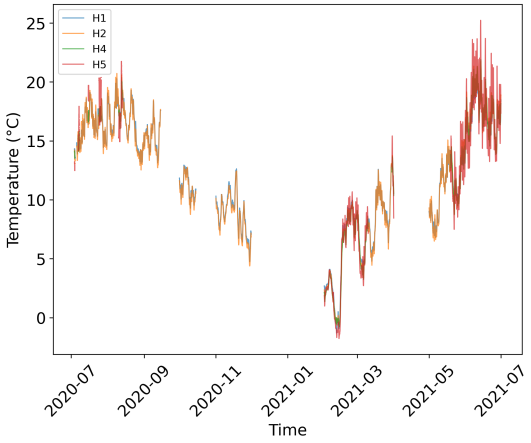
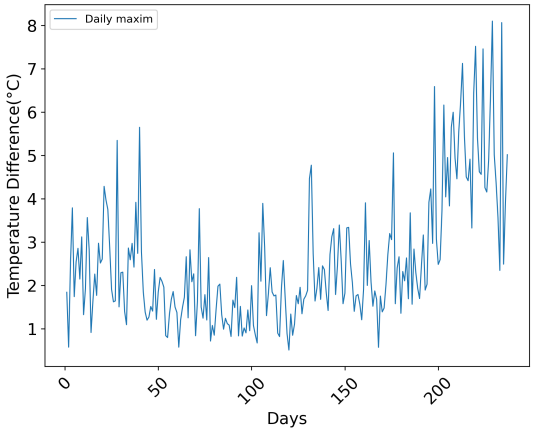


Figure 6.21: Horizontal temperature sensors

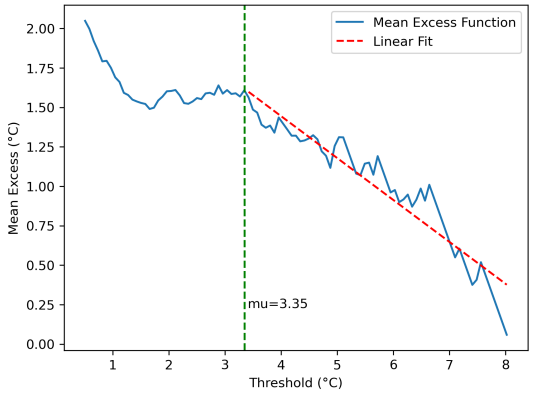
Figure 6.22c reveals that a linear curve exists between 3.35 to 8. The R^2 between the linear fit and original data is 0.904 and the p-value of the linear curve is 1.015×10^{-31} which is much smaller than 0.05, indicating the chosen linear line is sufficient to represent the MEF plot in the threshold interval. The GPD is then calculated based on these thresholds. The shape parameter ξ and the scale parameter σ are determined to be -0.265 and 1.981 respectively. The extreme deck horizontal temperature difference corresponding to a 5.0% exceedance probability over 120 years was estimated to be $9.86°C$.



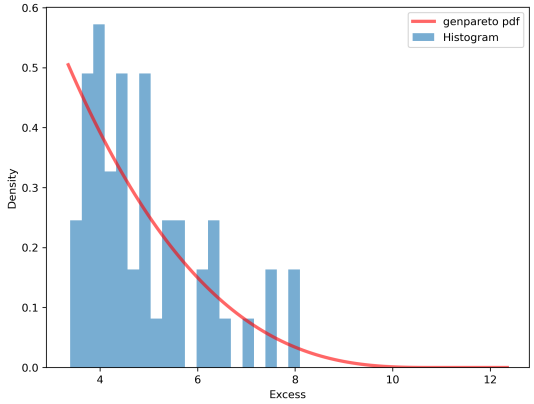
(a) Deck horizontal temperature



(b) Deck horizontal temperature difference



(c) MEF plot for deck horizontal temperature difference



(d) GPD plot for deck horizontal temperature difference

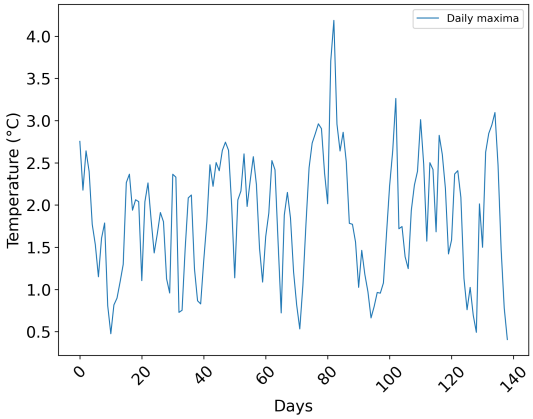
Figure 6.22: Deck horizontal temperature difference analysis

In the British Standard (BSI, 2006), no specification is provided for the maximum horizontal temperature difference. However, the Chinese Design Code (CMOT, 2015) specifies a design value to be $4.5°C$. In this case, the estimation based on SHM data exceeds the design value. Such a discrepancy in horizontal temperature differentials may induce transversal bending in the girder. This phenomenon shows as tensile stress on the side experiencing high temperatures, while the side subjected to lower temperatures incurs compressive stress (X. Xu, Xu, et al., 2021).

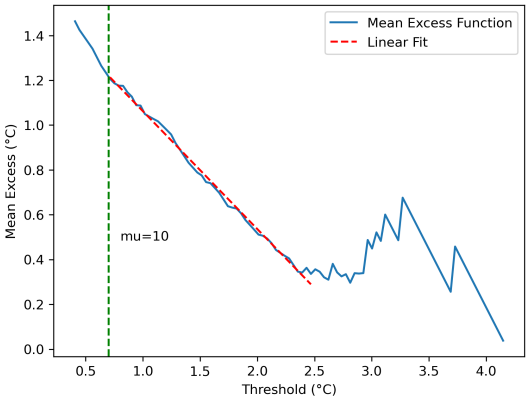
It is known that steel has a high thermal conductivity and is sensitive to ambient temperature change. Therefore, a temperature difference in the same steel section can cause its geometries and elevation to change which ultimately introduces additional stress in box girder (Miao & Shi, 2013). Furthermore, the temperature difference caused by ambient temperature and solar radiation can act as cyclic loading and eventually lead to fatigue which will affect the serviceability of the bridge deck (Y. Liu et al., 2017). Therefore, in design codes, temperature difference serves as an indication of thermal stress.

6.5.4 Tower temperature difference

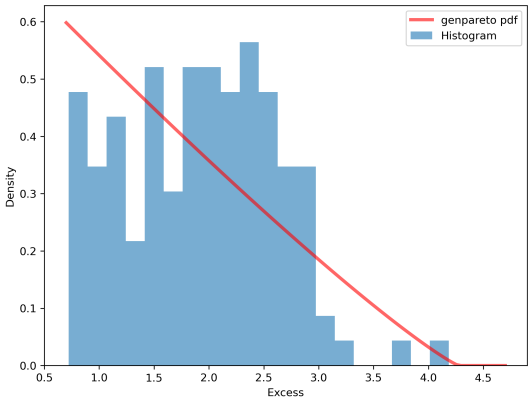
The inner and outer temperature difference on the centre tower is depicted in Figure 6.23a. Analysing the MEF plot in Figure 6.23b, it is observed that the curve exhibits strong linearity for thresholds between 0.7 and 2.5. However, when the threshold surpasses 2.5, the curve begins to oscillate and no discernible patterns are apparent. Moreover, the GPD plot in Figure 6.23c indicates that the excess does not conform to the Pareto distribution. This phenomenon can be attributed to the insufficient measurements on the tower, with only 140 days of data available for a one-year period. In other words, 63% of data were missing. Consequently, the measurements may not accurately represent the actual variation in temperature difference on the centre tower. In this case, the extreme tower temperature difference corresponding to a 5.0% exceedance probability over 120 years is estimated to be $4.66^{\circ}C$. Despite the absence of specific provisions in the British Standard (BSI, 2006), the Chinese Design Code (CMOT, 2015) stipulates a design value for the extreme tower temperature difference of $5.0^{\circ}C$. In this context, the estimated extreme value is marginally smaller than the design value prescribed in the Chinese code. Consequently, the design of the tower adheres to safety standards.



(a) Inner & outer temperature difference on centre tower



(b) MEF plot for tower temperature difference



(c) GPD plot for tower temperature difference

Figure 6.23: Tower inner & outer temperature difference analysis

6.6 Summary

In this Chapter, an analysis of the temperature field was conducted, focusing on the thermal sensors installed at various locations on the QC, encompassing the deck, tower, and cable components.

Initially, the layout of the thermal sensors was introduced, detailing their specific locations on the Queensferry Crossing QC and their installations on various bridge components. Subsequently, a one-year thermal data analysis was conducted on key locations, including the south midspan deck section, the centre tower mid-height

section, and the cable connecting the centre tower and deck lower end. Within the deck section, temperature data from four distinct locations were examined: the soffit, deck top chord (steel), deck bottom chord (steel), and the concrete deck. The analysis revealed that the temperature difference in the steel section was significantly smaller compared to the data from the concrete deck, attributable to the higher thermal conductivity of steel. A comparison of data from different locations further disclosed that the deck top chord temperatures consistently exceeded those of the deck bottom chord, with differences ranging from 0 to $7.5^{\circ}C$. This discrepancy was attributed to the direct solar radiation impacting the top chord. Regarding the temperature difference on the tower section, a substantial variation was observed between the inner and outer sides of the tower, ranging from 0 to $4^{\circ}C$, a phenomenon influenced by the thickness of the section and the low thermal conductivity of the concrete.

Utilising the insights gained from the analysis of temperature differences, the Generalised Pareto Distribution (GPD) was employed to estimate the extreme temperature differences for three specific scenarios over a reference period of 120 years. The first scenario focused on the vertical temperature difference on the deck, denoted as Δt_{vg}. The second scenario examined the deck horizontal temperature difference Δt_{hg}. The third scenario investigated the temperature difference between the inner and outer sides of the centre tower, denoted as Δt_{w}. The calculated value for Δt_{vg} was $7.66^{\circ}C$, significantly lower than the $24^{\circ}C$ specified in the British Standard. However, the estimation on Δt_{hg} was $9.86^{\circ}C$, larger than the design specifications. Furthermore, the value for Δt_{w} was found to be $4.66^{\circ}C$, slightly below the $5^{\circ}C$ stipulated in the Chinese Design Code. It is noteworthy that the GPD fitting for the tower temperature difference was somewhat weak, a limitation attributed to the restricted amount of data available from the tower.

From the comprehensive analysis conducted, several key insights and recommendations can be drawn:

- Data integrity is vital. For a robust estimation of extreme temperatures using the GPD, a minimum of one year of monitoring data at the locations of interest is essential.
- The current design specifications for both the deck vertical direction and tower appear to be adequate in accommodating thermal loads, as evidenced by the GPD analysis. However, the acquisition of additional temperature data on the cable is necessary to enhance the accuracy of GPD fitting.
- Attention should be paid to the thermal load in the horizontal direction of the bridge deck.
- A notable absence in the current British Standard is the lack of guidelines about thermal loads design for the deck horizontal direction and bridge towers, a provision found in the Chinese Design Code. The inclusion of such guidelines is recommended to assist engineers in their design processes.
- The investigation into data missing revealed that the issue is predominantly linked to the failure of data acquisition units rather than the sensors themselves. This failure leads to a loss of data across entire sections, rendering data imputation unfeasible.

These findings not only contribute to the understanding of thermal effects on bridge structures but also highlight areas for potential improvement in both design standards and data acquisition practices. In the next chapter, the thermal-induced mid-span deflections are discussed.

Chapter 7

Thermal-induced response analysis

7.1 Backgrounds

In Chapter 6, a comprehensive investigation was conducted into the variations of uniform temperature and temperature difference across different parts of the Queensferry Crossing, utilising one-year Structural Health Monitoring (SHM) data. This chapter extends the analysis to explore the effects of thermal loads on the structural response of the Queensferry Crossing.

The complexity of mechanical behaviour in a long-span cable-stayed bridge arises from the significantly different shapes, sizes, and materials of its components. Consequently, the structural response may exhibit varying sensitivity to thermal loads depending on the specific bridge components (L. Li et al., 2023). Therefore, it is vital to identify structural changes when thermal loads are applied to different bridge components. Thermal loads have an influence on both the dynamic and static behaviours of a bridge. The impact on dynamic behaviours has been explored in the context of vibration-based damage detection studies (C. R. Farrar et al., 1994; Peeters, Maeck, & De Roeck, 2001). However, analysing dynamic behaviours requires high-frequency data, and the raw data from the QC is sampled at a relatively low rate of 1Hz. This limitation precludes the possibility of conducting a dynamic analysis on the QC. Consequently, this section will focus exclusively on investigating and discussing the static behaviours of the bridge under the influence of thermal loads.

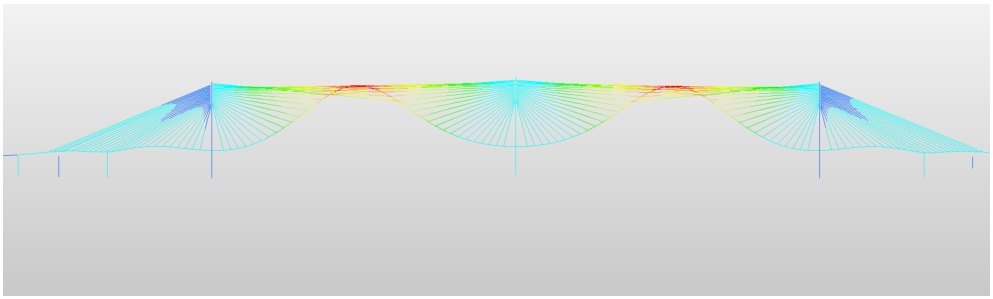
First, the finite element model of the QC is used to identify the influence of thermal loads on different bridge components. Next, the identified thermal loads and the deck deflections are compared in terms of variations and correlations. Finally, neural network models are used to predict the bridge deflections based on temperature data.

7.2 Digital twin analysis

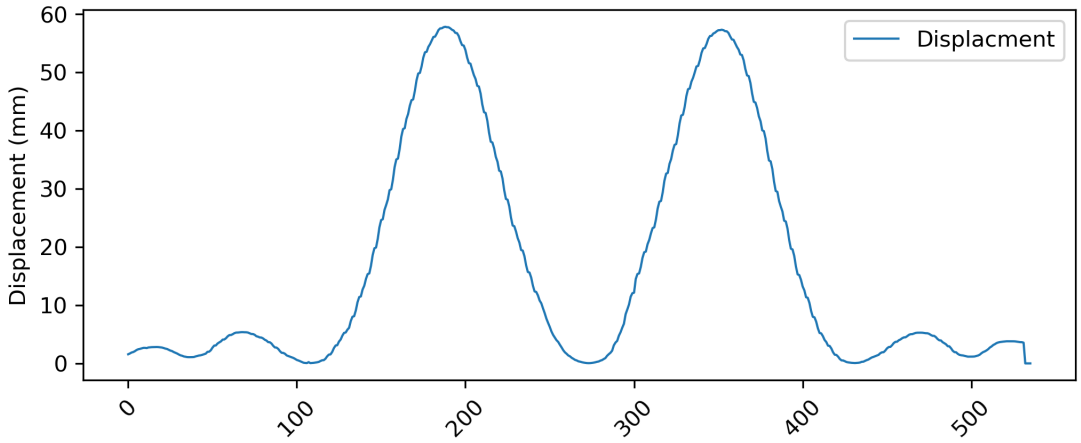
In the MIDAS Civil finite element software, temperature loads can be imposed on the target structure as uniform temperature changes (referred to as element temperature).

Figure 7.1 illustrates the deck deformation and vertical displacement under a uniform thermal load of 10 $^{\circ}C$ applied to the deck. Two peaks in displacement are observed in Figure 7.1b, located at the south and north midspans, with the largest displacement occurring at the south midspan, measuring 57.814 mm.

Similarly, the deck deformation diagram and displacement plot for a uniform temperature applied to all three towers are depicted in Figure 7.2a and Figure 7.2b, respectively. In this scenario, the highest displacement is recorded at the south midspan, reaching 16.402 mm.

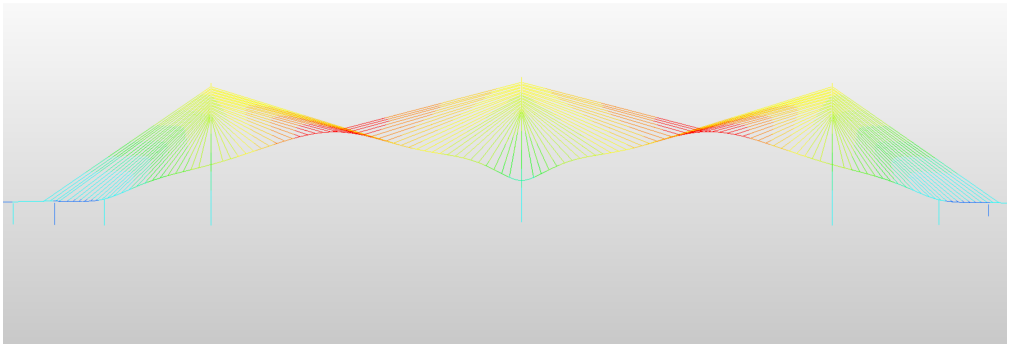


(a) Deck deformation

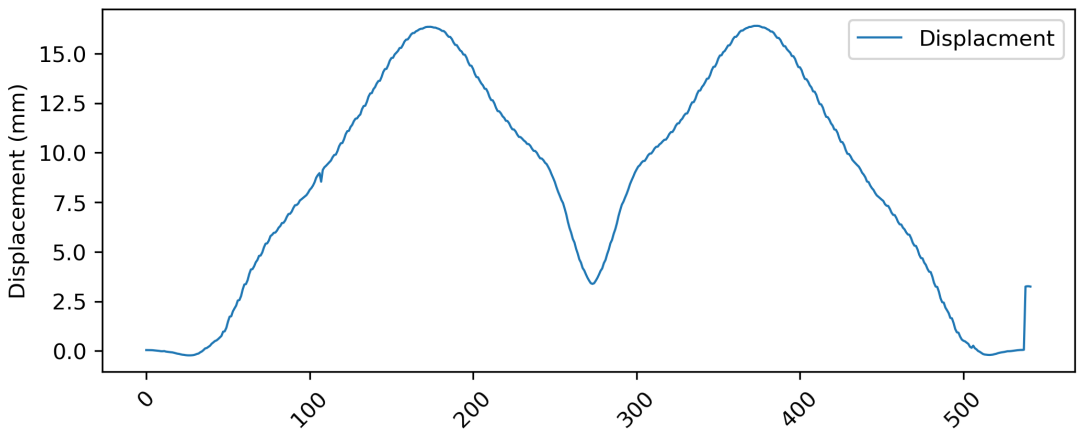


(b) Deck displacements in vertical direction

Figure 7.1: Deck deformation for $10°C$ thermal loads on deck



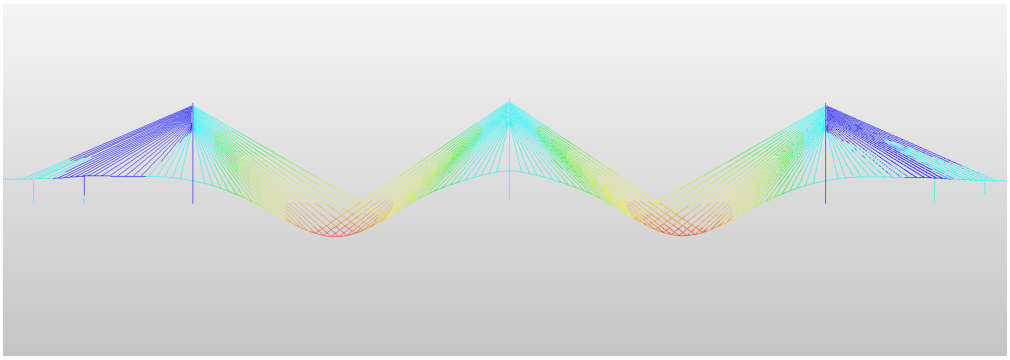
(a) Deck deformation



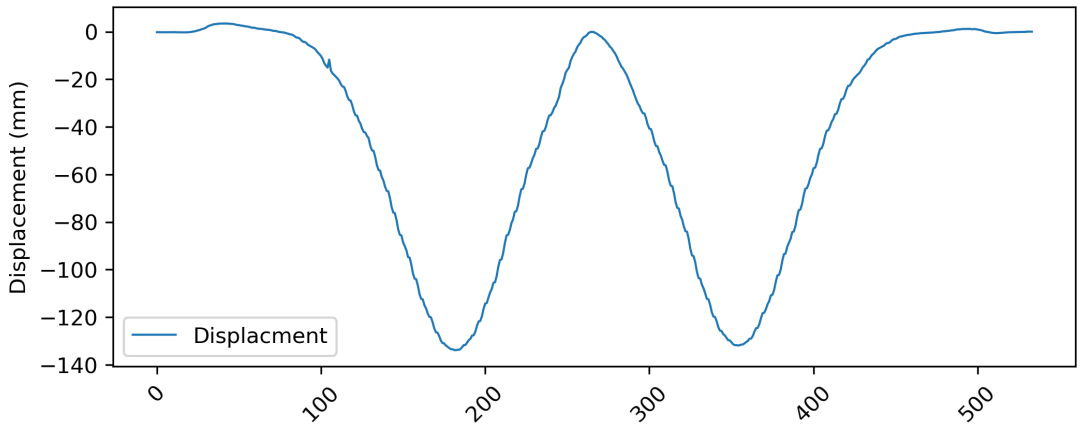
(b) Deck displacements in vertical direction

Figure 7.2: Deck deformation for $10°C$ thermal loads on tower

When the $10^{\circ}C$ thermal load is applied to all cables on the QC, a substantial maximum deck displacement of -133.866 mm occurs at the south midspan.



(a) Deck deformation



(b) Deck displacements in vertical direction

Figure 7.3: Deck deformation for $10^{\circ}C$ thermal loads on cable

These analyses of three uniform temperature load cases reveal that the maximum displacement was observed at the south midspan for all cases. Consequently, the data from GPS stations at the midspan will be used to study the deflection data on the QC. A comparative examination of the three cases uncovers that increasing temperatures of girders and towers resulted in hogging deflections, while increasing temperatures of cables resulted in sagging deflections. The specific deflections at the south midspan under various thermal loads applied to different bridge components are catalogued in Table 7.1. From the table, we can see that the south mid-span deflections are more sensitive to thermal effects from the cable and the deck.

Table 7.1: South midspan deflections under different thermal load cases

Thermal load case	South midspan deflections per degree($mm/^{\circ}C$)
Decks	5.781
Towers	1.640
Cables	-13.387
Combined	-5.960

In the next section, the conclusions drawn from the finite element analysis will be verified against the SHM data from the QC.

7.3 Data-driven analysis

7.3.1 QC SHM data

Deflections (GPS)

Having identified the critical section under thermal loads on the QC as the south midspan through FE analysis, attention is now directed to the actual monitoring data. Specifically, data from GPS stations (826 & 957), depicted in Figure 7.4 and situated on the deck rack at the south middle span, will be employed in the subsequent investigations.

Given the necessity for continuity in analysing time-series data, particularly from the standpoint of predictions and correlations, the SHM data selected for this study spans from September 2020 to December 2020. This selection was made in light of observed complete data missing in January and April of 2021. The vertical deflections of the south midspan, recorded at a 1Hz frequency, are illustrated in Figure 7.5a and Figure 7.5b. Notably, only 0.2% of the data are missing, in the form of occasional single

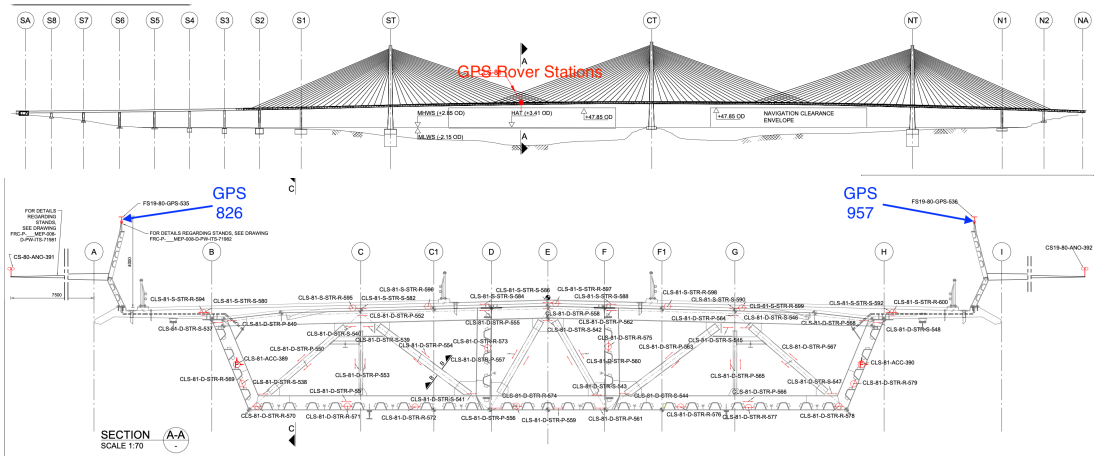
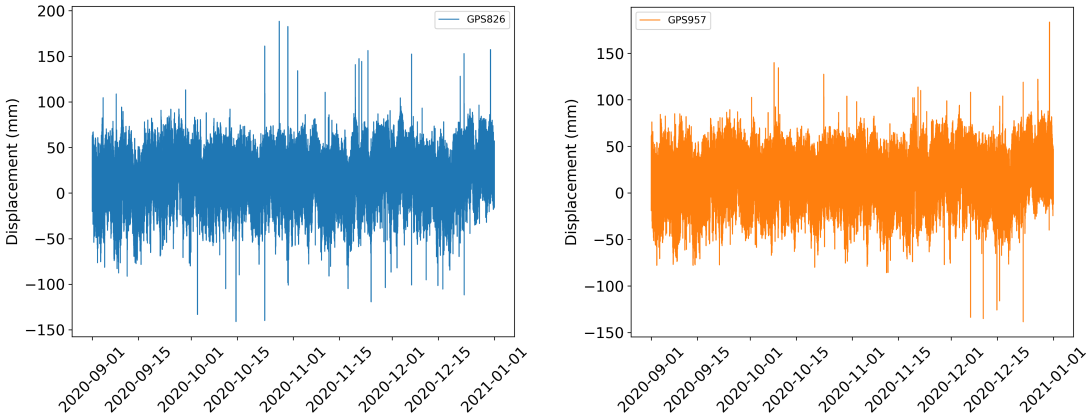


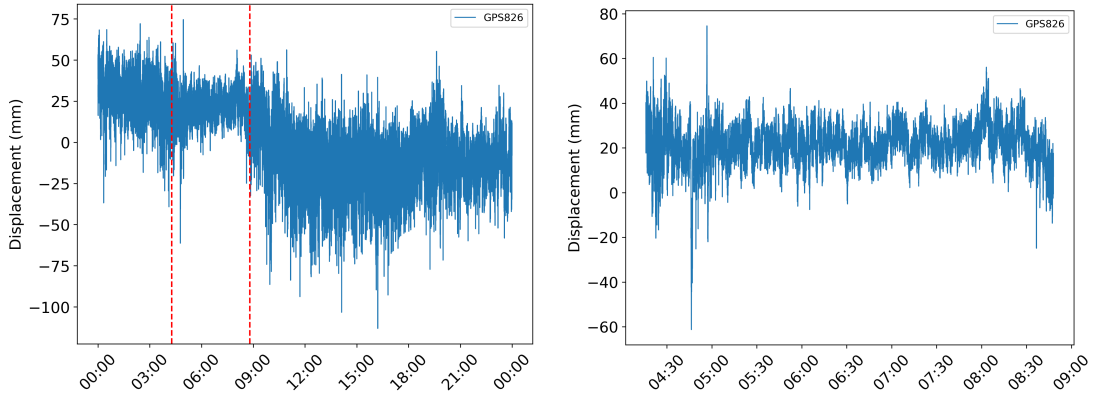
Figure 7.4: Sensor 826 & 957



(a) GPS826 09/20-12/21 **(b)** GPS957 09/20-12/21

Figure 7.5: South midspan deflections

data point losses. To maintain the integrity of the dataset, linear interpolation was employed to impute these missing values. Furthermore, the deflections from the two GPS stations show similar variations and value ranges. The measurements range from -100mm to 100mm, with occasional spikes above 100mm.



(a) GPS826 04/12 **(b)** GPS826 04:16-08:48

Figure 7.6: South midspan deflections at the bridge closure period

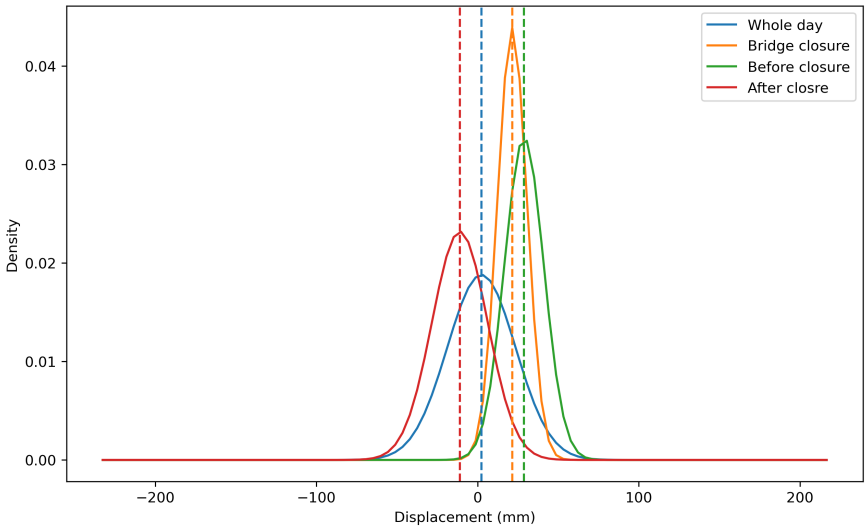
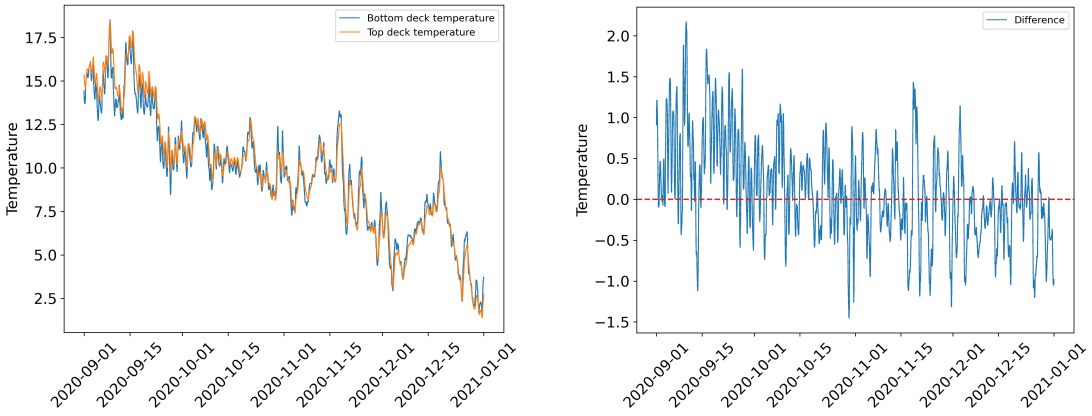


Figure 7.7: South midspan deflections distributions at bridge closure period

On 4 December 2020, the QC was temporarily closed for 4 hours and 32 minutes (from 04:16 to 08:48) due to falling ice. The GPS measurements for that day are displayed in Figure 7.6a. Notably, during the closure, the measurements, as seen in Figure 7.6b, exhibit a narrower variation range compared to the bridge's operational hours. This observation is further underscored by Figure 7.7, where the probability density function (PDF) curve for the closure period showcases reduced variance.

Steel deck temperature



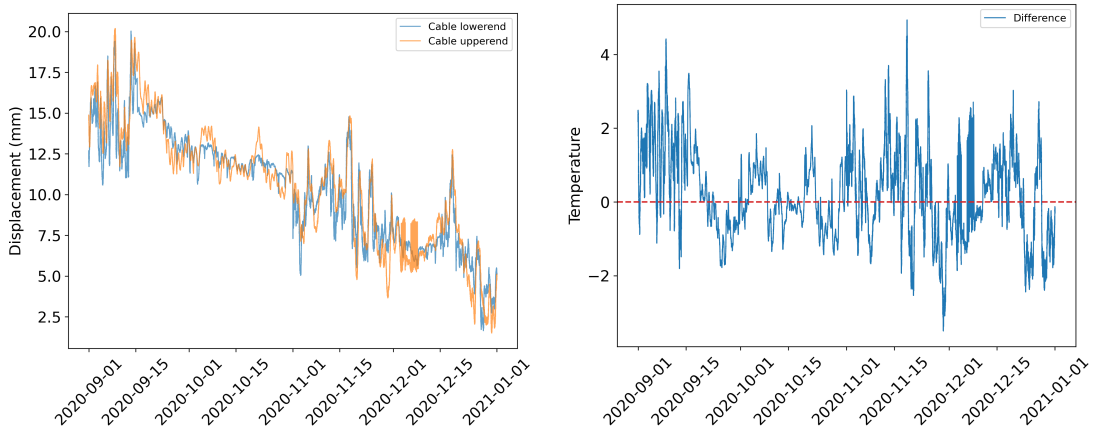
(a) Top & bottom chord temperature **(b)** Deck temperature difference

Figure 7.8: Deck temperaure

From September to December 2020, the temperatures of the deck's top and bottom chords are illustrated in Figure 7.8a. Over these months, there's a noticeable decline in the deck's vertical temperature difference. As depicted in Figure 7.8b, the top surface was warmer in September 2020, but this disparity diminished by late October, likely due to decreasing solar radiation on the top surface as the months progressed into the winter.

Cable temperature

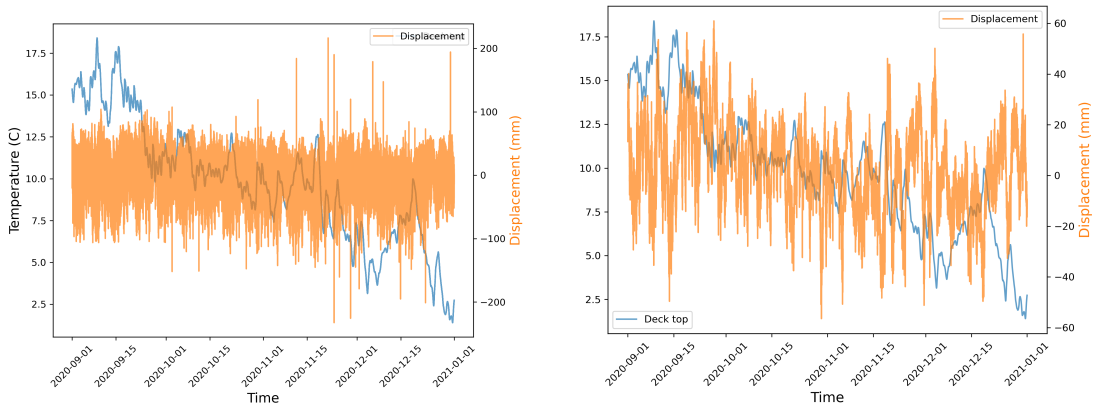
The temperatures at the upper and lower ends of the stay cable connecting the centre tower and the deck are depicted in Figure 7.9a. Their temperature difference, which doesn't display a clear seasonal trend, is shown in Figure 7.9b.



(a) Upper and lower ends of the longest cable

(b) Temperature difference

Figure 7.9: Cable temperature



(a) Midspan deflections & deck top temperature

(b) Midspan deflections & deck top temperature resampled

Figure 7.10: Deck temperaure

7.3.2 Temperature-deflection relationship

Figure 7.10a depicts the deck top chord temperature measurements and the south midspan deflections. Since no apparent relationship was observed, the raw data were resampled to 600 seconds, 1/600 Hz, shown in Figure 7.10b. It is observed that displacement measurements oscillate in a higher frequency compared with the deck top chord temperature varying in a much slower trend. This discrepancy is attributed to other dynamic loads on the bridge, such as vehicles and wind. To truly understand

the impact of thermal-induced deflections, these dynamic influences must be isolated. While the DWIM data on the QC was found to be unreliable for direct analysis, as discussed in Chapter 4, a simplified method was identified to filter out the high-frequency variations in displacement measurements caused by these dynamic loads, wavelet transform.

Wavelet transform

The wavelet transform (WT) is an advanced mathematical technique that evolved from the Fourier transform (FF). While the FF breaks down signals based on the assumption that they are composed of sinusoidal functions that persist throughout the entire signal, the WT offers a more advanced approach. Specifically, WT functions are localised in both time and frequency domains. This means that they can represent both short-lived high-frequency changes and more sustained low-frequency changes in a signal.

This localisation property makes the WT especially adept at analysing signals that have non-stationary or time-varying characteristics. For instance, when considering the high-frequency displacements in a bridge caused by dynamic loads, these displacements might not exhibit a consistent pattern or seasonality. The WT can capture these transient changes effectively. X. Xu et al. (2020) suggested using the multi-resolution wavelet approach to reconstruct and emphasise the dynamic load-induced structural responses (essentially highlighting the high-frequency changes), in this study, we aim to do the opposite. We'll use the WT to filter out these high-frequency changes, allowing us to focus on and reconstruct the deflections caused by thermal loads, which operate at a lower frequency.

WT of the target signals can be expressed as:

$$W_f^{\psi}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\psi^* \left(\frac{t-b}{a} \right) dt \tag{7.1}$$

where a and b represent two parameters scale and translation, $f(t)$ is the target signals, ψ^* is the complex conjugate of the predefined wavelet function ψ. The wavelet function can be written as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{7.2}$$

In real applications, discrete wavelet transform (DWT) is more popular for reducing the computation cost:

$$x(n) = \sum_k \sum_j \alpha_{j,k}\psi_{j,k}(n) \tag{7.3}$$

$$\psi_{j,k}(n) = \frac{1}{\sqrt{j}}\psi\left(\frac{n-k}{j}\right) \tag{7.4}$$

$$\alpha_{j,k} = \frac{1}{\sqrt{j}}\sum_n x(n)\psi^*\left(\frac{n-k}{j}\right) \tag{7.5}$$

where m and n represent the scale and translation parameters. $\psi_{j,k}(t)$ is the scaled and shifted $\psi(n)$; $\alpha_{j,k}$ is the preset wavelet function. The Multi-Resolution Analysis (MRA) using the Wavelet Transform is used to process the GPS displacement data. MRA is typically described through its decomposition and reconstruction processes. The decomposition involves filtering the signal with a low-pass filter $h(n)$ and a high-pass filtering $g(n)$ to generate two coefficients, *approximation* $A(n)$, representing the low frequency and *detail* $D(n)$, representing high frequencies:

$$A_j(n) = \sum_k f(k)\cdot h(2n-k) \tag{7.6}$$

$$D_j(n) = \sum_k f(k)\cdot g(2n-k) \tag{7.7}$$

The approximation coefficient can be further decomposed in subsequent levels. Therefore, in MRA, the displacement measurements $f(n)$ can be represented by A_j, D_j for j-level decomposition:

$$f(n) = \sum_k (A_j[k] \cdot \hat{h}[n-2k] + D_j(k) \cdot \hat{g}[n-2k]) \tag{7.8}$$

The decomposition level j can be determined using:

$$j = \log_2 \left(\frac{n}{f_s - 1} \right) \tag{7.9}$$

where n is the length of the signals, f_s is the length of the wavelet function.

Temperature-Deflection

The wavelet transform was executed using the PyWavelet Python library. We decomposed the signal into 8 levels, and the detailed coefficient plots can be seen in Figure 7.11. From this, we noticed that the higher decomposition levels (from $D4$ to $D8$) captured the high-frequency variations in the displacement data. To focus on the lower frequencies, which are more indicative of thermal effects, we removed these high-frequency components by setting D_i to 0, $i \in [4, 8]$.

The outcome of this wavelet processing is shown in Figure 7.12. To investigate more profound into the results, we zoomed into four random time windows, as shown in Figure 7.13. From this closer examination, it's evident that the displacement data, after being processed by the wavelet transform, correlates well with the deck's top temperature. However, there are noticeable time lags between the two sets of data. This suggests that the impact of temperature changes on the bridge's displacement doesn't occur instantly but takes some time to manifest. A similar phenomenon was also observed between cable temperature and displacement data.

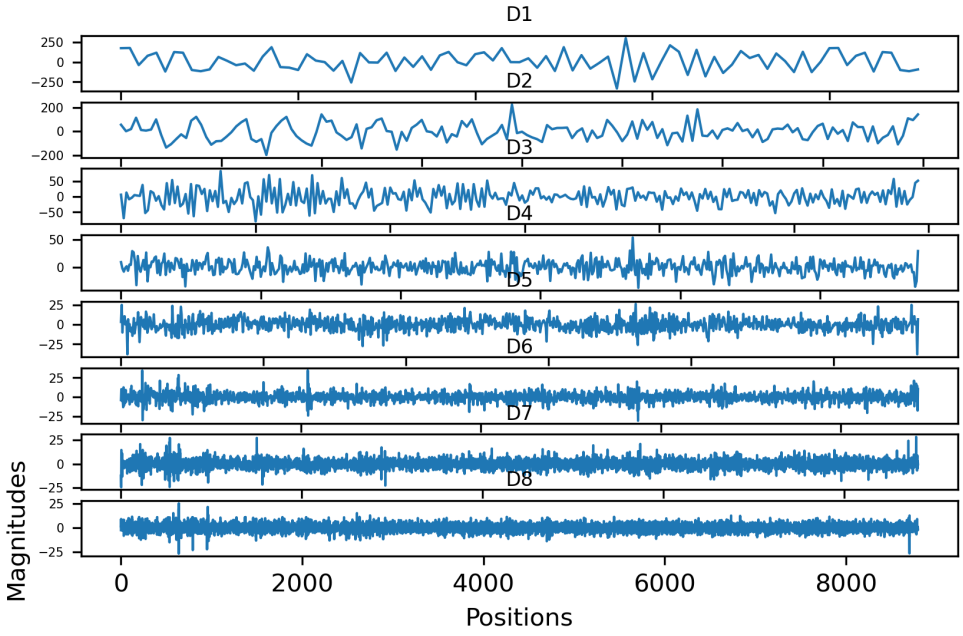


Figure 7.11: Detail coefficients for 8 levels decomposition

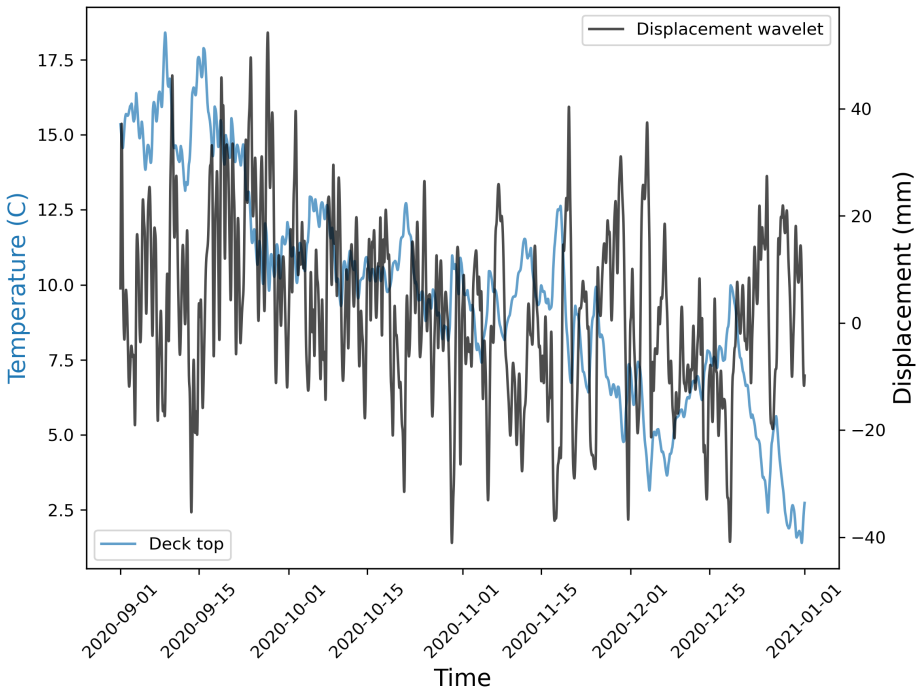


Figure 7.12: Resampled displacement data using wavelet transform

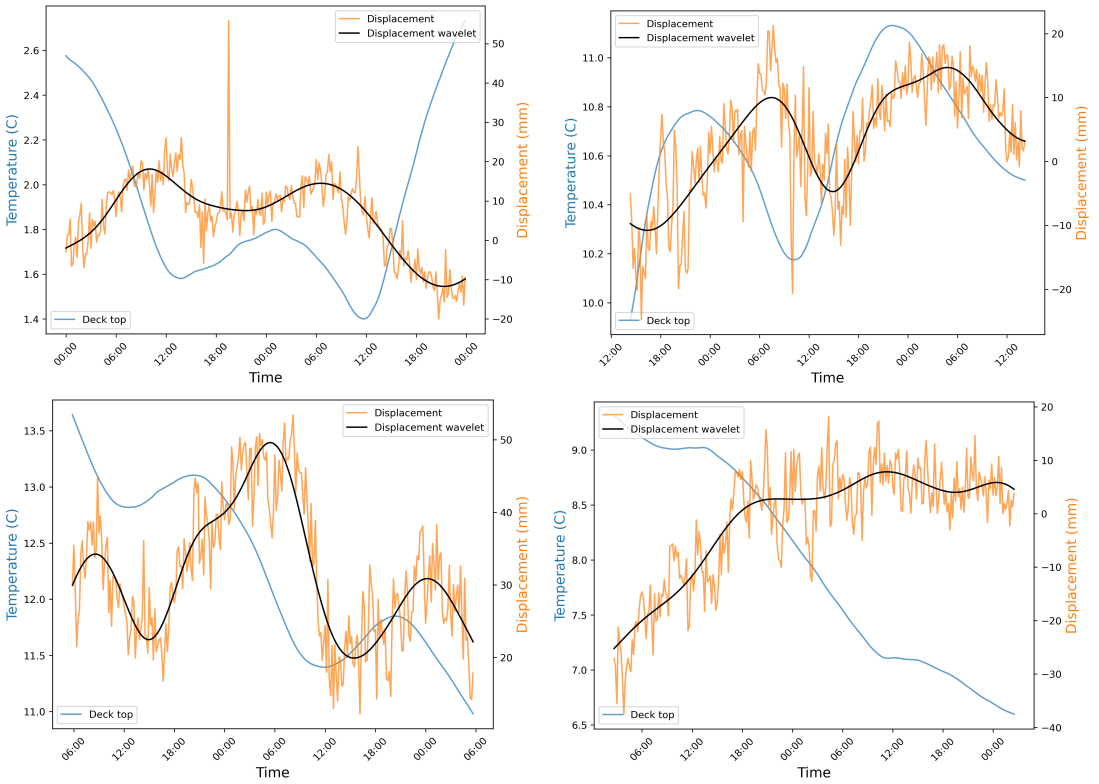


Figure 7.13: Deck temperature and resampled displacements comparison

7.3.3 Thermal-induced deflection predictions with LSTM

Dataset

The previous data-driven analysis based on 4-month QC SHM data demonstrates a correlation exists between the south midspan deflections with the time lags existence. Furthermore, the finite element analysis confirms the thermal loads applied to the deck and cables will have a large contribution to the deflections. Consequently, to accurately predict thermal-induced deflections, the application of LSTM is deemed appropriate due to its inherent capability to recognise time-dependent patterns.

The dataset for the LSTM model was prepared and some examples are displayed in Table 7.2 and Table 7.3. Temperatures recorded from various parts on the deck and from the cable are used as the input to the neural networks model. From the analysis in Chapter 6, we have learnt that multiple sensors were installed on the sites of interest.

Given the high thermal conductivity inherent to steel and the slender nature of steel plates, the observed temperature differential was found to be negligible. Consequently, for temperature readings from the soffit, as well as the deck's top and bottom chords, a singular channel suffices. In contrast, for the concrete deck temperature, a relatively large temperature difference across the section was observed due to the low thermal conductivity of the concrete materials. This necessitated the inclusion of data from all five sensors positioned at the same locale. Analogously, for the cable temperatures, readings from both the lower and upper end sensors were incorporated, accounting for the temperature difference induced by their spatial separation. This concluded in an input comprising ten distinct channels of thermal data.

For the output data, both the raw GPS measurements at the south midspan and the wavelet-transformed resampled GPS data are delineated in Table 7.3. Nonetheless, during the training phase, only the reconstructed data is utilised, serving as a singular output channel. It is assumed that the raw displacement measurements include the combined effects of both static and dynamic loads, such as traffic and wind. These dynamic effects present as high-frequency signals in the raw data. Consequently, when employing LSTM to investigate the deflections induced by thermal variations, these high-frequency signals are treated as noise.

In the training dataset, the sampling rate was adjusted from 1Hz to 1/600 Hz, given that temperature variations occur at a slower pace. As noted in the preceding section, there exists a temporal lag between the displacement measurements and the thermal data. Utilising the original sampling rate would necessitate a prolonged sequence for the LSTM to capture this lag, potentially leading to the vanishing gradient issue.

Table 7.2: Thermal-induced deflection predictions data samples (part 1)

Time	Sdeck_bottom ($^{\circ}C$)	Sdeck_top ($^{\circ}C$)	cabletem lowerend ($^{\circ}C$)	cabletem upperend ($^{\circ}C$)	Sdeck soffit ($^{\circ}C$)
15:10	14.505	14.971	13.778	15.845	14.304
15:20	14.505	14.978	14.003	15.875	14.327
15:30	14.510	14.988	14.250	15.894	14.367
15:40	14.520	15.001	14.347	15.901	14.408
15:50	14.535	15.016	14.565	15.958	14.464
16:00	14.555	15.032	14.429	15.990	14.488
16:10	14.571	15.049	13.931	15.982	14.453
16:20	14.574	15.063	13.601	15.935	14.391
16:30	14.565	15.075	13.538	15.854	14.328
16:40	14.547	15.084	13.338	15.772	14.254

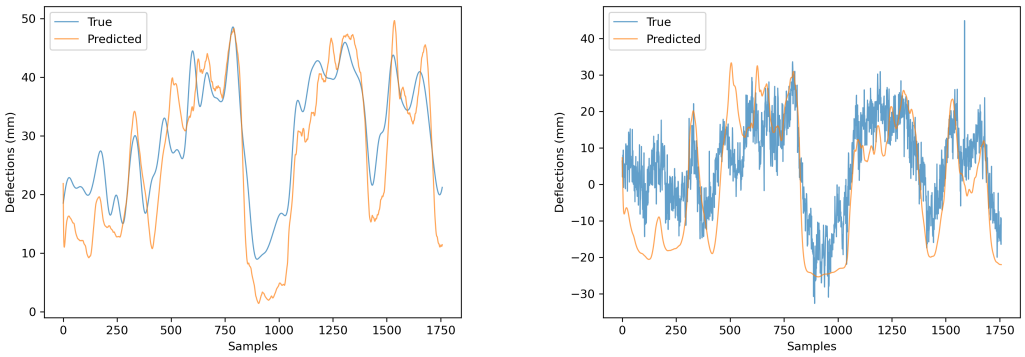
Table 7.3: Thermal-induced deflection predictions data samples (Part 2)

Time	cdeck_1 ($^{\circ}C$)	cdeck_2 ($^{\circ}C$)	cdeck_3 ($^{\circ}C$)	cdeck_4 ($^{\circ}C$)	cdeck_5 ($^{\circ}C$)	GPS (mm)	gps wavelet (mm)
15:10	16.575	15.564	14.728	14.460	14.266	-0.065	-1.734
15:20	16.595	15.583	14.754	14.469	14.263	0.343	-1.603
15:30	16.606	15.602	14.779	14.480	14.297	4.730	-1.542
15:40	16.611	15.620	14.803	14.492	14.346	-2.684	-1.526
15:50	16.614	15.637	14.825	14.506	14.397	3.921	-1.526
16:00	16.617	15.657	14.846	14.522	14.429	-3.585	-1.521
16:10	16.618	15.674	14.867	14.540	14.390	-1.843	-1.491
16:20	16.610	15.691	14.887	14.558	14.307	1.209	-1.419
16:30	16.583	15.707	14.907	14.575	14.247	-10.630	-1.285
16:40	16.532	15.722	14.926	14.590	14.180	-9.717	-1.071

Results & discussions

The LSTM model used to predict the thermal-induced deflections is similar with the model structure introduced in Chapter 5. However, in order to capture the time lag feature, the window size is set to be the equivalent of 6 hours. For each case, the model was trained 10 times to ensure the results were unbiased. A structure using a 3-layer LSTM model with 32 neurons in each layer was employed.

Figure 7.14a and Figure 7.14b depict the LSTM predictions results on the wavelet transform resampled displacement data and raw displacement data respectively. When the output was set to the resampled data, the model achieved the highest correlation coefficient R at 0.927, signifying a high prediction accuracy. As evident from Figure 7.14a, the predicted displacements closely align with the actual measurements in terms of both trends and values in most instances. This suggests that the proposed LSTM approach effectively identified the arbitrary time lag observed between the temperature and deflection measurements. In Figure 7.14b, the LSTM model struggled to capture the oscillations present in the raw displacement measurements. This discrepancy can be attributed to the absence of reliable DWIM data for the QC, rendering the dynamic loads unrecognisable to the LSTM model. Consequently, the wavelet transform resampling of displacements becomes essential for accurate predictions of thermal-induced deflections.

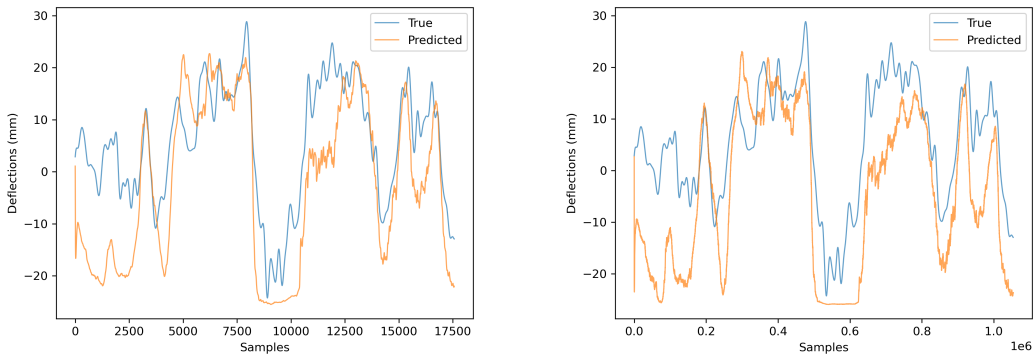


(a) LSTM predictions on resampled displacement data

(b) LSTM predictions on resampled displacement data

Figure 7.14: LSTM predictions

Figure 7.15a presents the prediction outcomes when the entire dataset was resampled at a frequency of 1/60 Hz. The peak correlation coefficient R recorded was 0.838. While the model successfully predicted the overarching trend throughout the test set, it exhibited inaccuracies in capturing local trends and specific values. This indicates



(a) LSTM predictions on 1/60 Hz data **(b)** LSTM predictions on original samples

Figure 7.15: LSTM predictions for different sampling rates

the model encounters difficulties in detailed patterns. In this dataset, the window size was set to 1440, which is tenfold larger than that of the 1/600 Hz resampled data. As the data sequence lengthens, the model's accuracy appears to diminish. Similarly, when the model was trained with the dataset on the original sampling rate, the performance was also undesirable, as shown in Figure 7.15b. R is 0.828 in this case.

In conclusion, the LSTM-based approach effectively predicts the thermal-induced deflections at the south midspan of the QC. It is crucial to remove dynamic load factors from the deflection data as they become noise in the slow variation forecasting. Concurrently, the presence of a time lag between thermal data and deflection data necessitates a more extended window size (sequence) for the LSTM model. Consequently, training should steer clear of using data with a high sampling rate, as elongated sequences pose challenges for the LSTM model.

7.4 Summary

In this chapter, temperature-induced deflection on the Queensferry Crossing was investigated.

Initially, the finite element model of the bridge was utilised to study the effects of thermal loads on different bridge structures, including deck, tower and cables. When a uniform temperature increase of 10 $^{\circ}C$ was applied to these components, the analysis revealed that the thermal loads exerted on the cables had the most pronounced effect on deck deflection. Moreover, the deck's thermal loads also played a significant role. Notably, the most substantial deflection occurred at the south midspan across the three scenarios.

Subsequently, we employed a 4-month dataset from the QC to explore the interrelation between temperature and midspan deflections. On a broader timescale, no discernible relationship between midspan deflection and temperature was evident. However, when the displacement data were resampled by the wavelet transform to remove dynamic load effects, a correlation with random time lag was observed in zoomed-in time windows. This suggests a localised interplay between temperature and deflections.

Finally, a 3-layer LSTM model was introduced to predict thermal-induced midspan deflections based on temperature data from different parts of the deck and both ends of the stayed cable. The model successfully forecast the deflection on the low sampling rate dataset. The diminished accuracy observed in raw displacement data predictions indicates the importance of removing dynamic load factors from the signals.

Nevertheless, this wavelet-based LSTM prediction model is not without its limitations. Owing to the absence of DWIM data from the QC, as highlighted in Chapter 4, the efficacy of the wavelet transform remains unverified. Ideally, a comparison should be made between displacement measurements resampled by wavelet transform and those adjusted considering actual dynamic load factors derived from wind and traffic data.

Chapter 8

Conclusions

8.1 Overall conclusions

Over the past several decades, the deployment of Structural Health Monitoring (SHM) systems on long-span bridges has significantly advanced, offering engineers an innovative methodology for condition assessments and damage detection. These SHM systems generate large datasets that include invaluable insights into the operations of bridges. However, the complex nature of these data, coupled with a general lack of expertise in data science, poses considerable challenges for bridge owners in terms of data interpretation and analysis. This thesis is devoted to the rigorous examination of SHM data from two long-span bridges, namely the Queensferry Crossing and the Forth Road Bridge, focusing on investigating bridge responses attributable to thermal and traffic-related factors. The scope of the research undertaken and the principal findings are summarised as follows:

Comparison between initial Aims & Achievements

In Chapter 1, the aims of this project were described as gaining an understanding of raw long-term SHM data on long-span bridges and investigating the environmental and operational impacts on the bridge responses. Chapter 3 provided a detailed examination of the raw SHM data collected from both bridge structures, identifying data quality challenges and proposing methodologies for their mitigation. Chapter 4

and Chapter 5 studied the vehicle-induced deflections using a novel traffic simulation method. A neuron network approach was trained to accurately predict traffic-induced midspan deflections. Chapter 6 conducted a detailed temperature field analysis to investigate thermal loads on different parts of the QC. Chapter 7 proposed the novel wavelet-based neuron network model to forecast thermal-induced deflections. Based on the comprehensive analyses and methodologies presented, it can be concluded that the research aims have been successfully achieved.

Specific conclusions

1. Maintaining data integrity is of paramount importance for any data analysis tasks. In our examination of the raw SHM data from both the Queensferry Crossing and the Forth Road Bridge, we encountered multiple data quality issues. These included both point-level and section-level data omissions, as well as the presence of outliers. To address these challenges, a suite of strategies was proposed and implemented. For point-level missing data, linear imputation was employed, while section-level data gaps were filled using a novel Random Forest algorithm based on relevant sensor data. Outliers were identified through a z-score filter and subsequently interpolated using Exponentially Weighted Moving Average (EWMA) techniques. As a final preprocessing step prior to further data analysis, feature engineering methods such as data normalisation were applied.
2. To address the issue of incomplete Weigh-in-Motion (WIM) data on the QC, a new comprehensive traffic simulation strategy was developed. The primary objective of this approach was to generate accurate vehicle trajectories on the bridge. The simulation commenced with the creation of a vehicle fleet, which was modelled based on hourly vehicle counts and associated attributes such as gross weight and speed. Utilising the bridge's finite element model, the influence line corresponding to vertical displacement at the south midspan was

calculated. Subsequently, simulated vertical deflections at the south midspan were computed using the influence line and simulated vehicle trajectories. This simulation methodology offers a viable alternative for generating traffic-related information, particularly when WIM data are either unreliable or entirely absent.

3. A Long-Short Term Memory (LSTM) recurrent neural network model was deployed to forecast bridge deflections based on vehicle attributes at the bridge entrance. The simulated vehicle data were restructured into a time-series format to serve as the model's input. A comprehensive parameter analysis was conducted to examine the influence of three key hyperparameters, overall data size, number of neurons, and number of layers on the model's predictive accuracy. The findings demonstrated that the LSTM model was highly effective in predicting deflections at the south midspan of the bridge. Moreover, multi-layer LSTM architectures outperformed their single-layer counterparts in terms of prediction accuracy, given an equivalent number of trainable parameters. In practical applications, the model can be trained on simulated vehicle attributes. Then, the trained model can be provided with the actual WIM data to acquire the actual deflections at locations of interest. This novel predictive approach offers engineers the capability to assess bridge deck deflections even at locations where GPS monitoring stations are not installed to perform further assessment of the bridge.

4. An analysis of the temperature field was conducted on several locations, including the top and bottom chord of the deck, the concrete deck, the tower and the longest stay cable. The analysis revealed that the temperature difference in the steel section was significantly smaller, generally ranging from 0 to $1.5°C$,

compared to the data from the concrete deck, attributable to the higher thermal conductivity of steel, ranging from 0 to 8°C. Therefore, the temperature difference should be considered in the analysis for concrete sections but can be neglected for steel sections.

5. Extreme temperature difference analysis was performed using the Generalised Pareto Distribution (GPD) for the vertical and transversal temperature difference on the deck and temperature difference between the inner and outer sides of the centre tower for a reference period of 120 years. The GPD results were then compared with both the British Standard and the Chinese Design Code. The extreme temperature estimations for deck vertical, horizontal and tower inner & outer surface temperature difference are 7.66°C, 9.86°C and 4.66 °C respectively. It revealed that the current design specifications for both the deck vertical direction and tower appear to be adequate in accommodating thermal loads. However, the extreme estimation for the deck horizontal thermal load exceeds the Chinese Design Code, 4.5 °C. Furthermore, a notable absence in the current British Standard is the lack of guidelines about the thermal loads design for thermal loads on the deck horizontal direction and towers. Therefore, a revision of the British Standard is needed.
6. FE model of the bridge was employed to investigate temperature-induced deflections of the deck when a uniform thermal load was applied to different bridge components, including the deck, towers and cables. The results showed that thermal loads on the cables and the deck had the most pronounced effect on deck deflection, by $-13.387mm/°C$ and $5.781mm/°C$ respectively.
7. Utilising wavelet transform techniques, this research examined the relationship between temperature variations and midspan deflections, as captured by the SHM data from the QC. The wavelet transform effectively isolated the effects of dynamic loads, revealing a correlation between temperature and deflections

that exhibited random time lags. To further explore this relationship, a 3-layer LSTM model was introduced to predict thermal-induced midspan deflections based on temperature data from different parts of the deck and both ends of the longest stay cable. The LSTM model demonstrated robust predictive capabilities, particularly when applied to datasets with low sampling rates, thereby confirming its utility for forecasting thermal-induced midspan deflections.

8.2 Limitations

1. The data quality from the SHM system presented significant challenges for analysis. Specifically, the dataset was contaminated by extensive missing segments and numerous outliers. It was not clear from the substantial volume of data that we received that there were gaps in the data - i.e. the data has to be preprocessed before gaps or faults are visible. There was no possibility timewise of obtaining extra data before the end of this project. Hence, we had to fill the gaps in the data and propose estimation strategies.
2. Furthermore, we observed that data missing could occur for the entire cross section due to operational issues of the data acquisition unit instead of the sensors.
3. The original sampling rate is not sufficient to perform dynamic analysis. The sampling rate of the measurements needs to be at least twice the natural frequencies of the bridge to provide accurate estimations (Chang, Chang, & Zhang, 2001). For example, the raw data from both bridges have a sampling rate of 1Hz. Therefore, it is only sufficient to estimate a natural frequency of 0.5Hz. Therefore, it is recommended to use a higher sampling rate in case of dynamic analysis, e.g. 10Hz.

8.3 Recommendations for future works

1. To further validate the accuracy of the proposed traffic simulation strategy, reliable WIM data should be used. Specifically, real traffic attributes should be employed as input to the pre-trained LSTM model. The resulting predictions can then be compared with actual monitored deflections to assess the model's efficacy and accuracy in a real-world context. This step would provide a more robust verification of the simulation strategy and its applicability in practical engineering scenarios.
2. To utilise an extended dataset, ideally spanning more than one year, for extreme temperature simulation. To compare the results when different data sizes are used in the GPD model and determine the minimal data size when a stable estimation result can be obtained.
3. To use the Transformer model and compare it with LSTM. At the end of this PhD project, Transformer has become popular in processing time sequence data. The current use of Transformer in SHM topics is rare.
4. To further enable damage detections based on observing deviations between the predicted displacement and monitored data. There, the damage detection threshold needs to be determined. A Gaussian process can be used for determining the threshold.

Appendix A

Trimodal Distributions for Each Vehicle Class

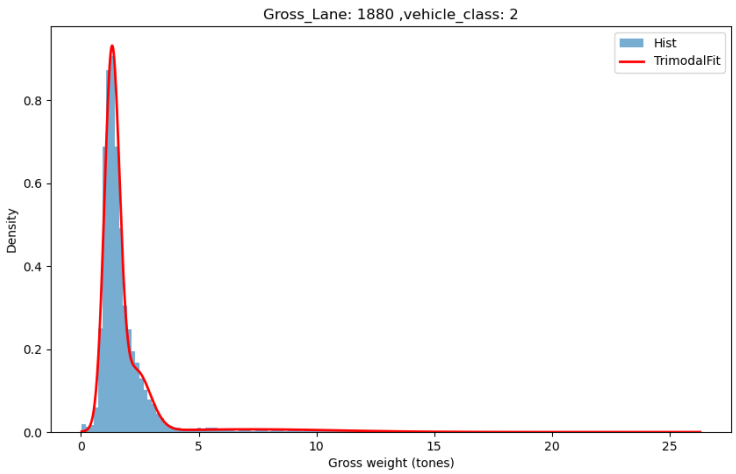


Figure A.1: Gross weight trimodal distribution for class 2 on lane 1880

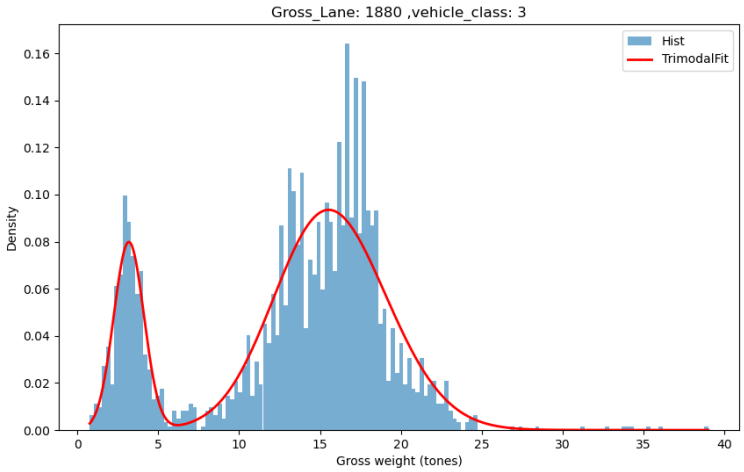


Figure A.2: Gross weight trimodal distribution for class 3 on lane 1880

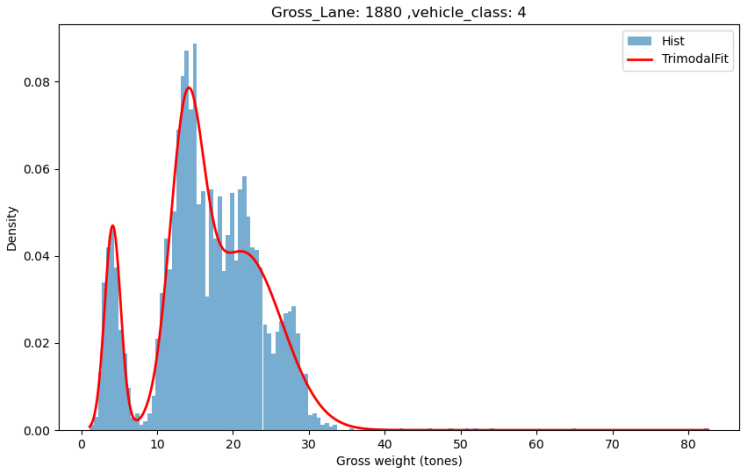


Figure A.3: Gross weight trimodal distribution for class 4 on lane 1880

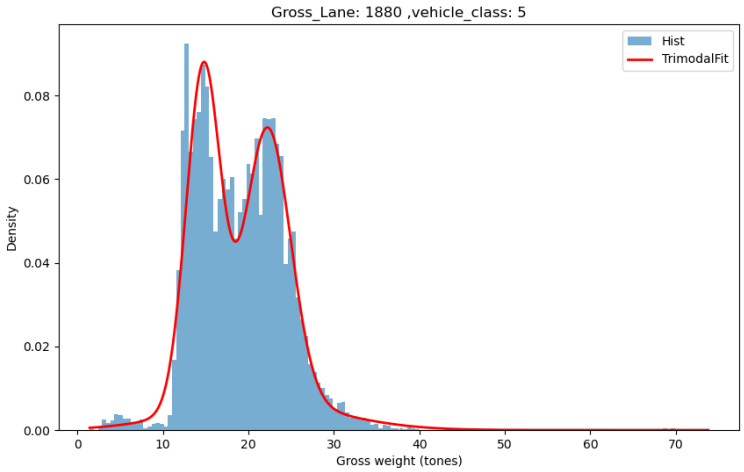


Figure A.4: Gross weight trimodal distribution for class 5 on lane 1880

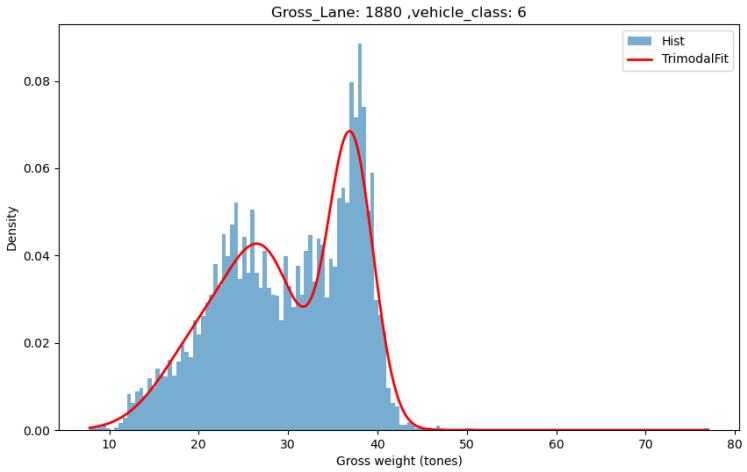


Figure A.5: Gross weight trimodal distribution for class 6 on lane 1880

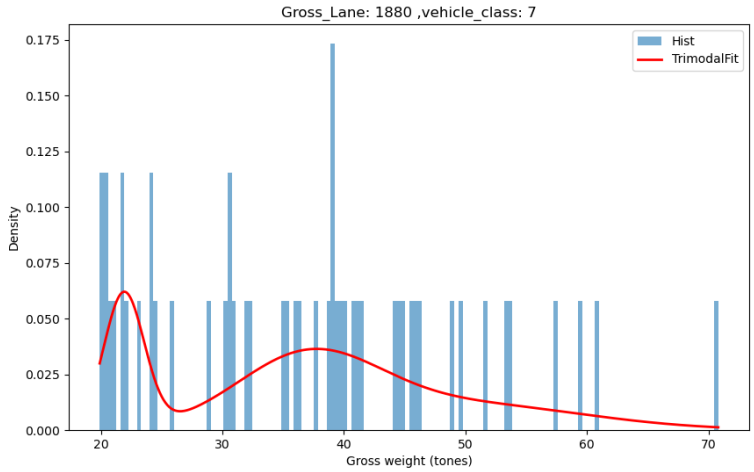


Figure A.6: Gross weight trimodal distribution for class 7 on lane 1880

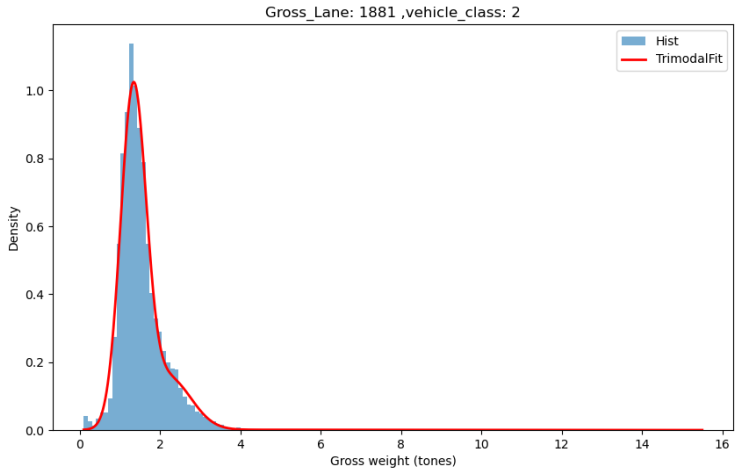


Figure A.7: Gross weight trimodal distribution for class 2 on lane 1881

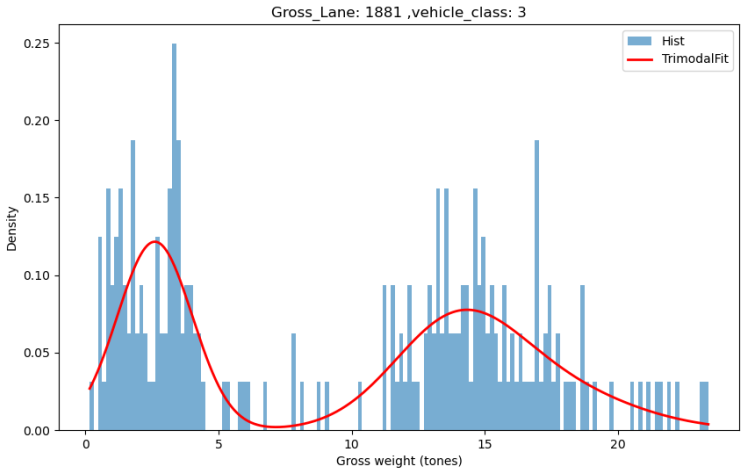


Figure A.8: Gross weight trimodal distribution for class 3 on lane 1881

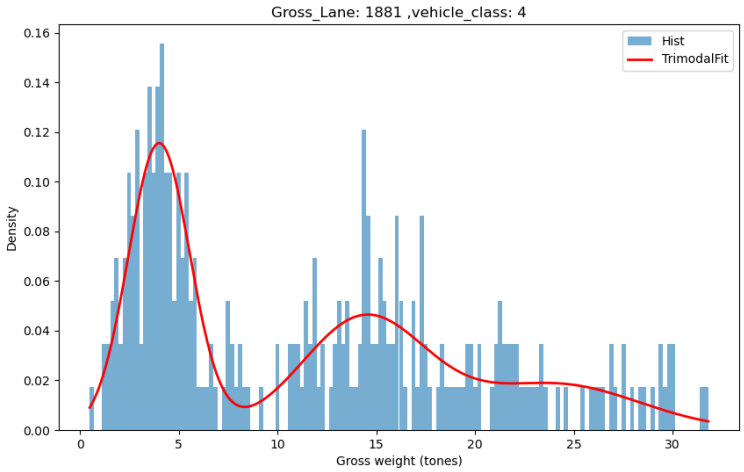


Figure A.9: Gross weight trimodal distribution for class 4 on lane 1881

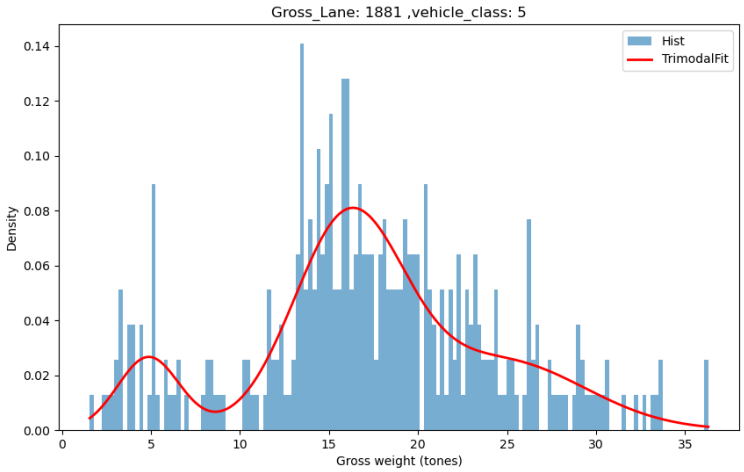


Figure A.10: Gross weight trimodal distribution for class 5 on lane 1881

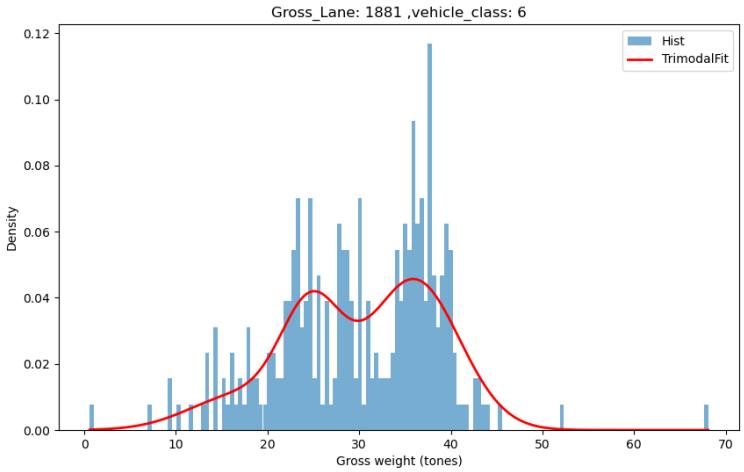


Figure A.11: Gross weight trimodal distribution for class 6 on lane 1881

Figure A.12: Gross weight trimodal distribution for class 7 on lane 1881

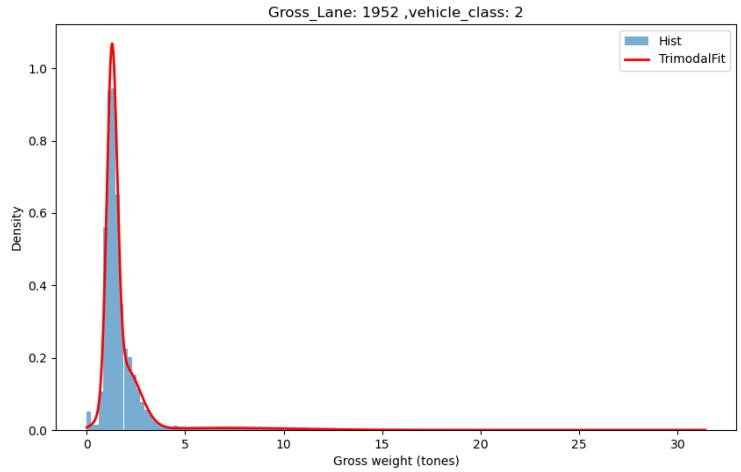


Figure A.13: Gross weight trimodal distribution for class 2 on lane 1952

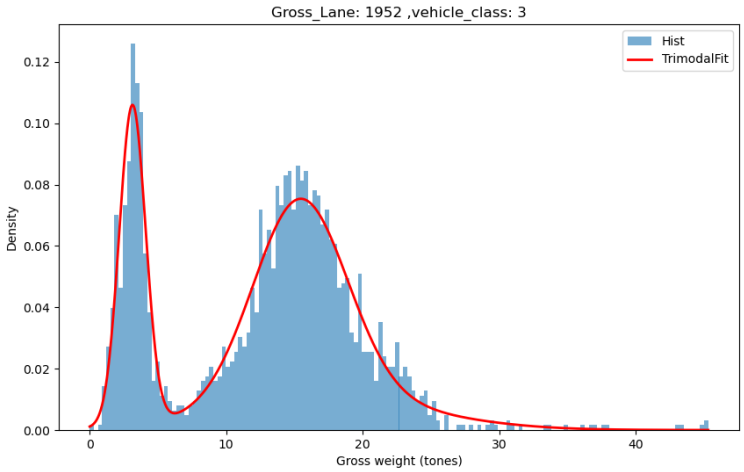


Figure A.14: Gross weight trimodal distribution for class 3 on lane 1952

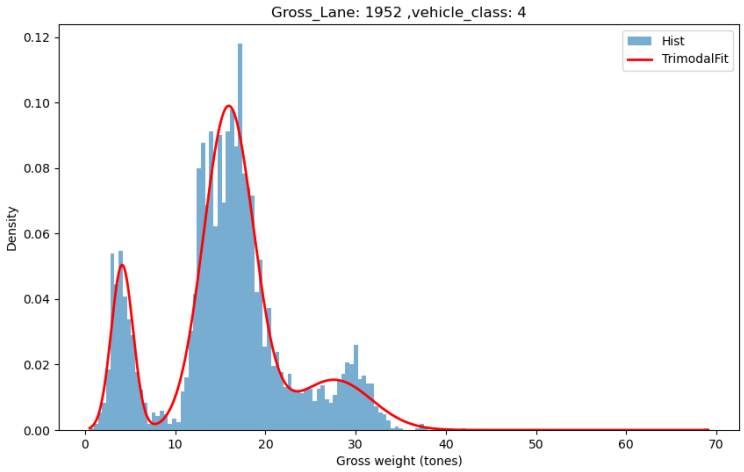


Figure A.15: Gross weight trimodal distribution for class 4 on lane 1952

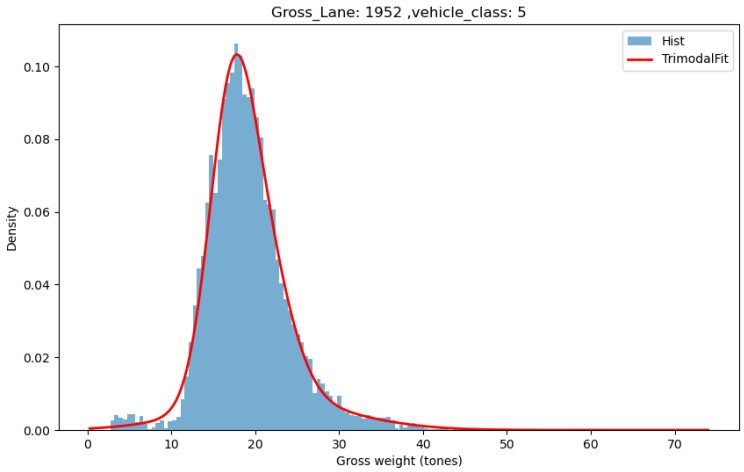


Figure A.16: Gross weight trimodal distribution for class 5 on lane 1952

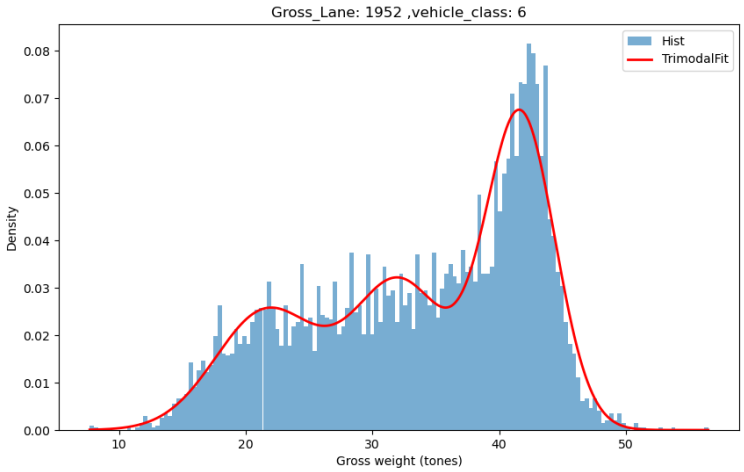


Figure A.17: Gross weight trimodal distribution for class 6 on lane 1952

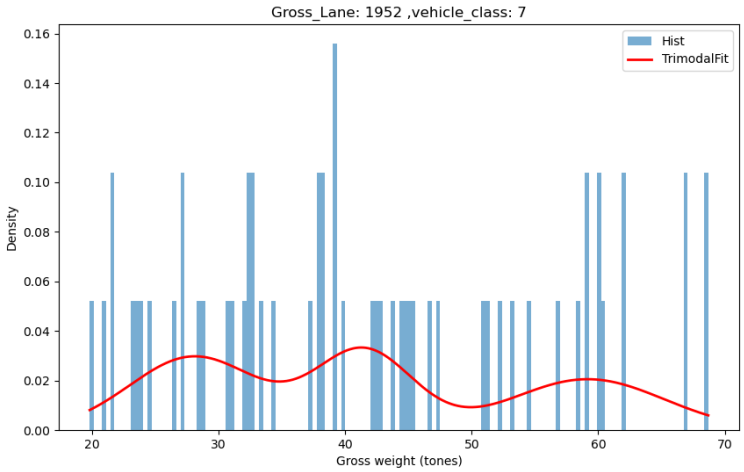


Figure A.18: Gross weight trimodal distribution for class 7 on lane 1952

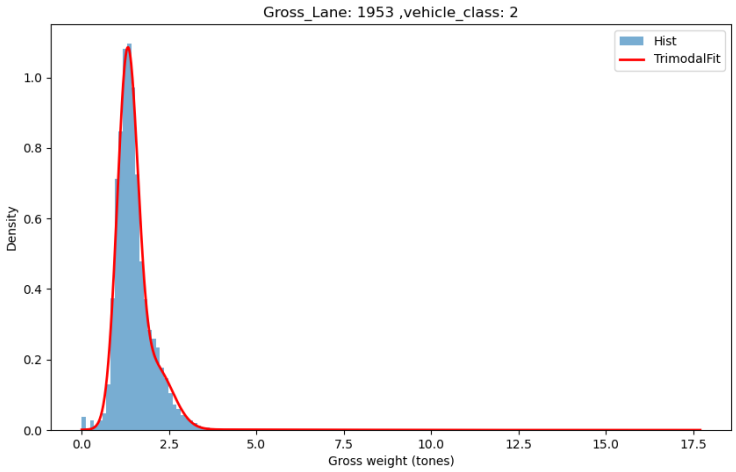


Figure A.19: Gross weight trimodal distribution for class 2 on lane 1953

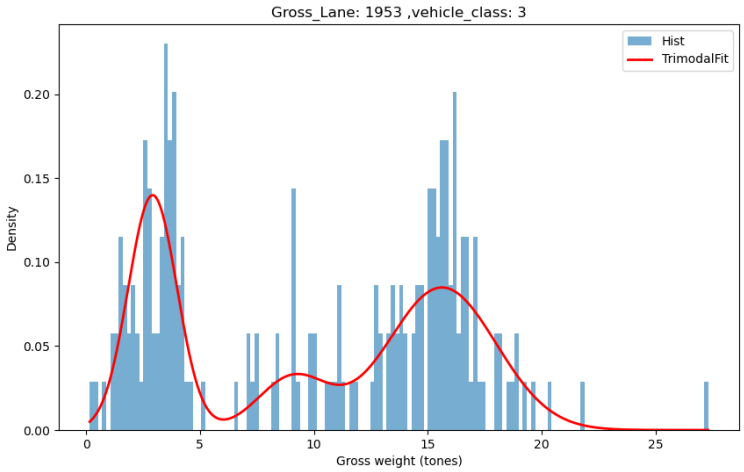


Figure A.20: Gross weight trimodal distribution for class 3 on lane 1953

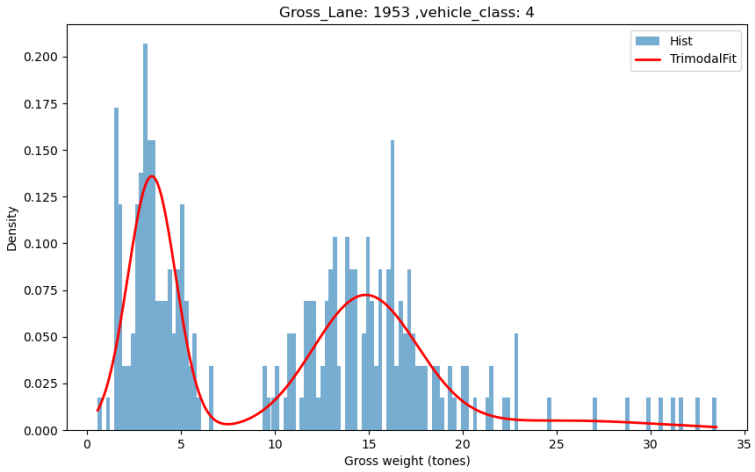


Figure A.21: Gross weight trimodal distribution for class 4 on lane 1953

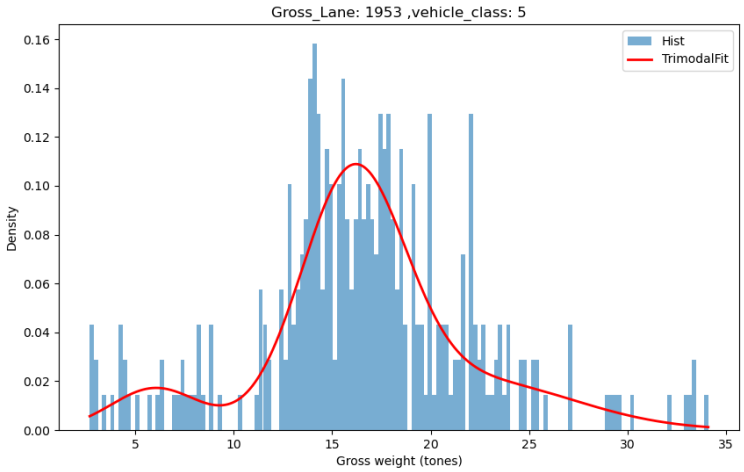


Figure A.22: Gross weight trimodal distribution for class 5 on lane 1953

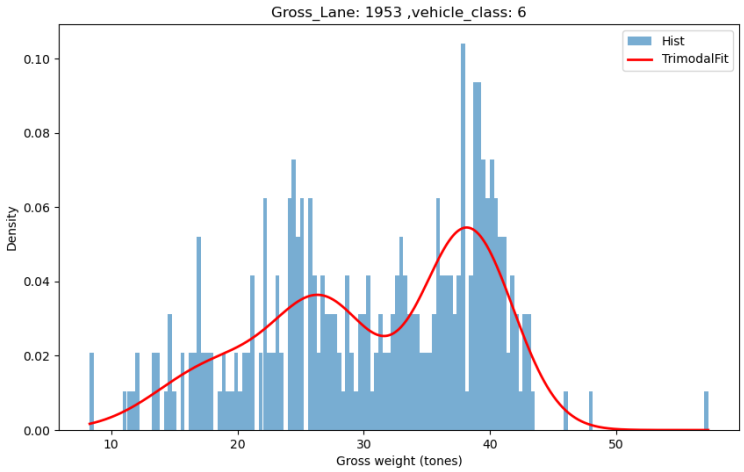


Figure A.23: Gross weight trimodal distribution for class 6 on lane 1953

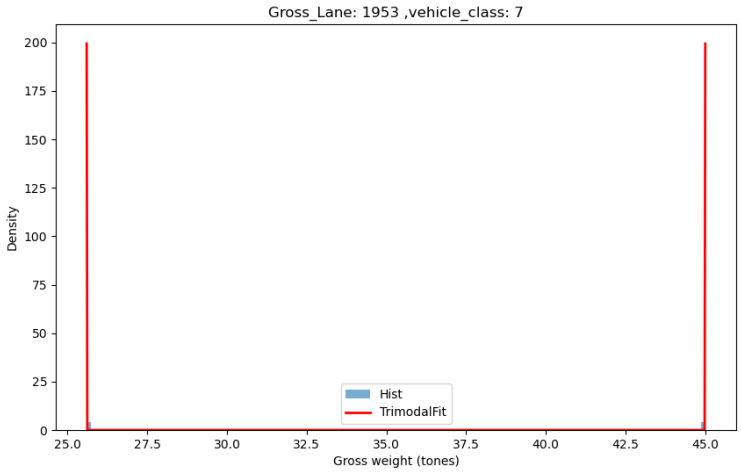


Figure A.24: Gross weight trimodal distribution for class 7 on lane 1953

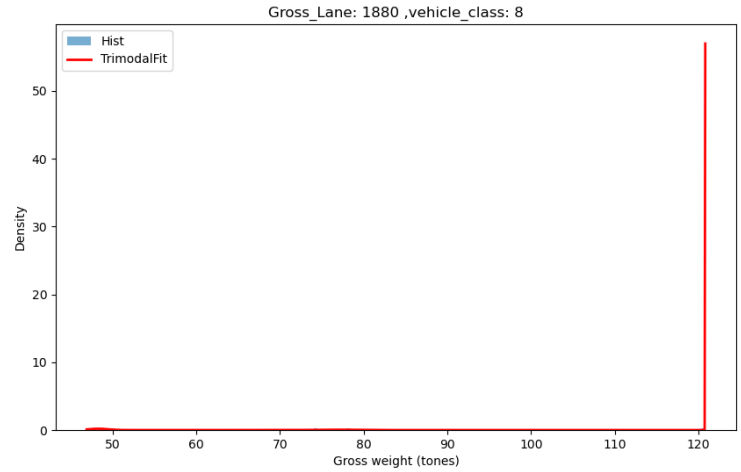


Figure A.25: Gross weight trimodal distribution for class 8 on lane 1880

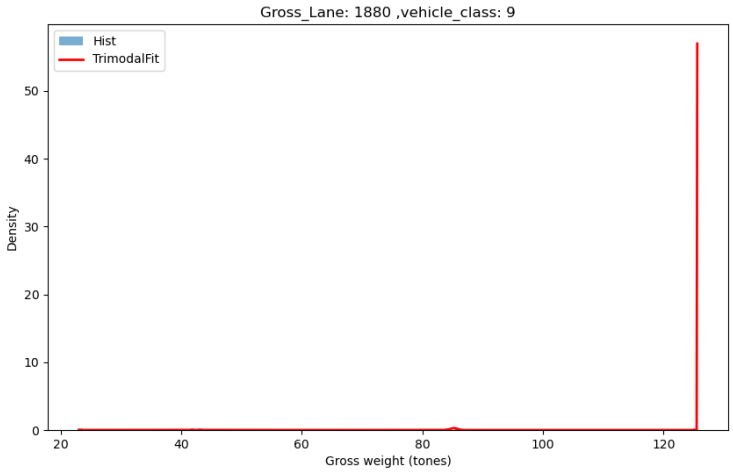


Figure A.26: Gross weight trimodal distribution for class 9 on lane 1880

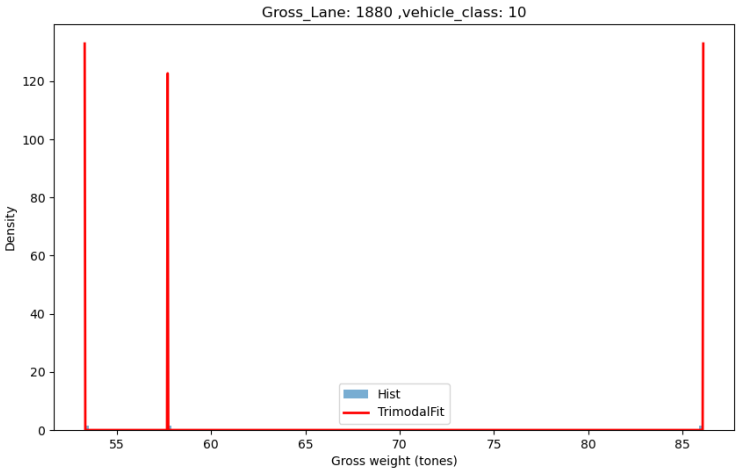


Figure A.27: Gross weight trimodal distribution for class 10 on lane 1880

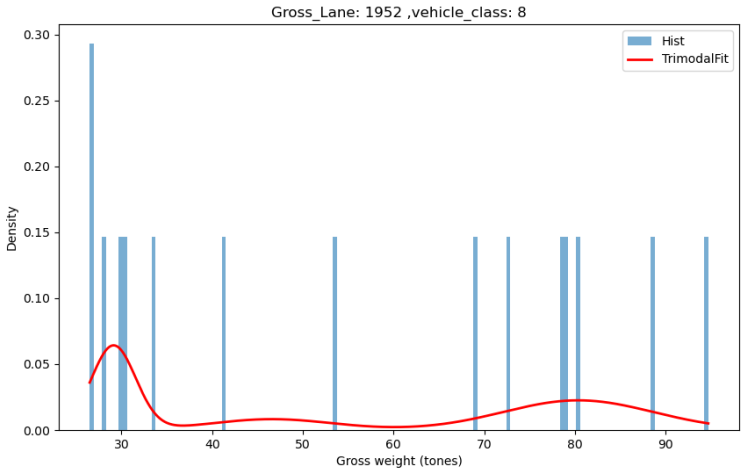


Figure A.28: Gross weight trimodal distribution for class 8 on lane 1952

Figure A.29: Gross weight trimodal distribution for class 9 on lane 1952

Figure A.30: Gross weight trimodal distribution for class 10 on lane 1952

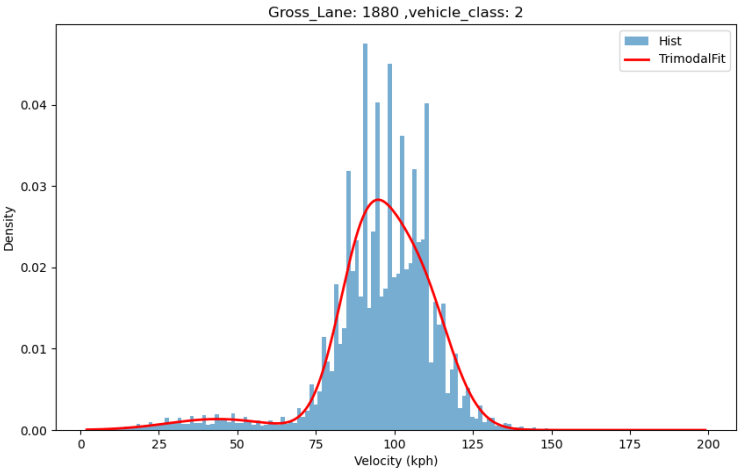


Figure A.31: Speed trimodal distribution for class 2 on lane 1880

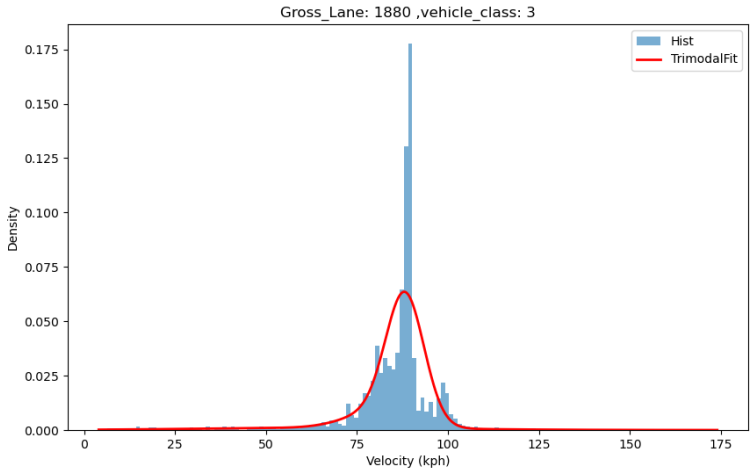


Figure A.32: Speed trimodal distribution for class 3 on lane 1880

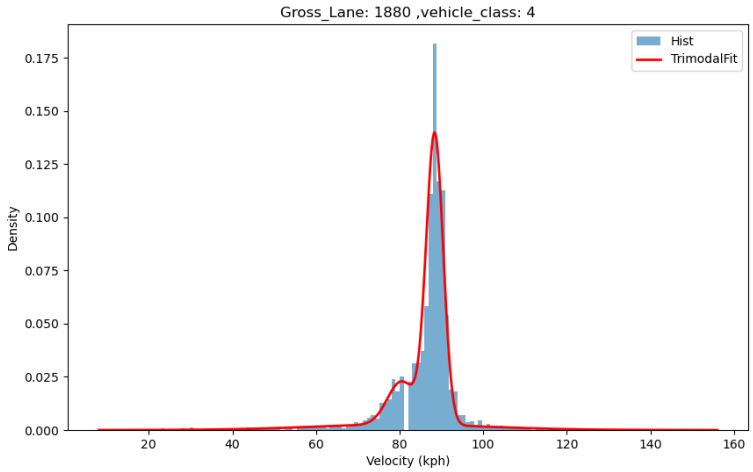


Figure A.33: Speed trimodal distribution for class 4 on lane 1880

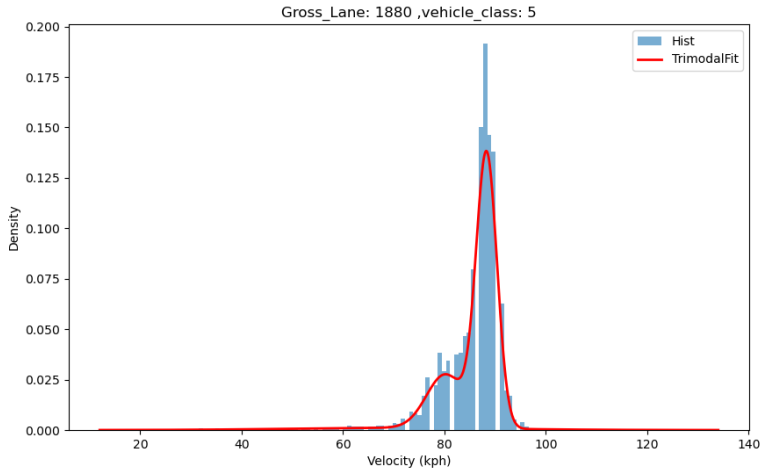


Figure A.34: Speed trimodal distribution for class 5 on lane 1880

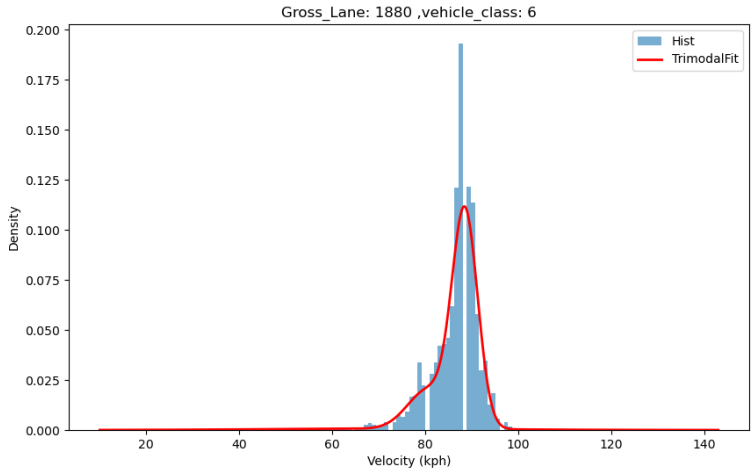


Figure A.35: Speed trimodal distribution for class 6 on lane 1880

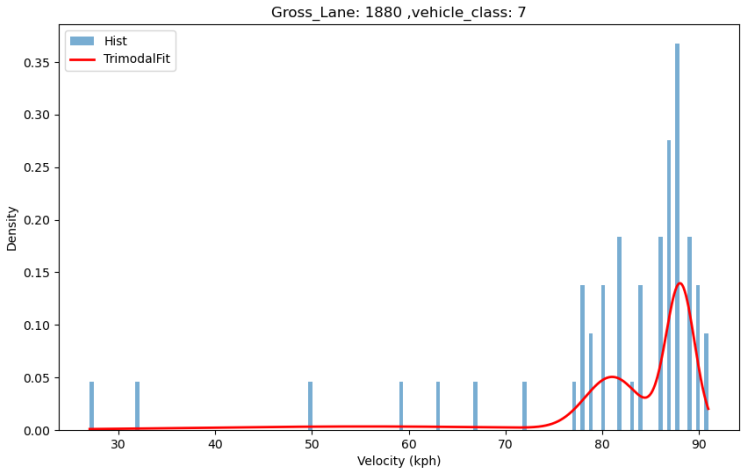


Figure A.36: Speed trimodal distribution for class 7 on lane 1880

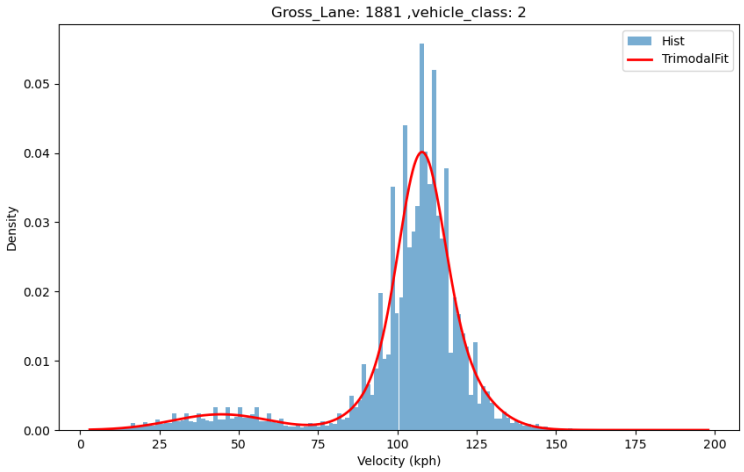


Figure A.37: Speed trimodal distribution for class 2 on lane 1881

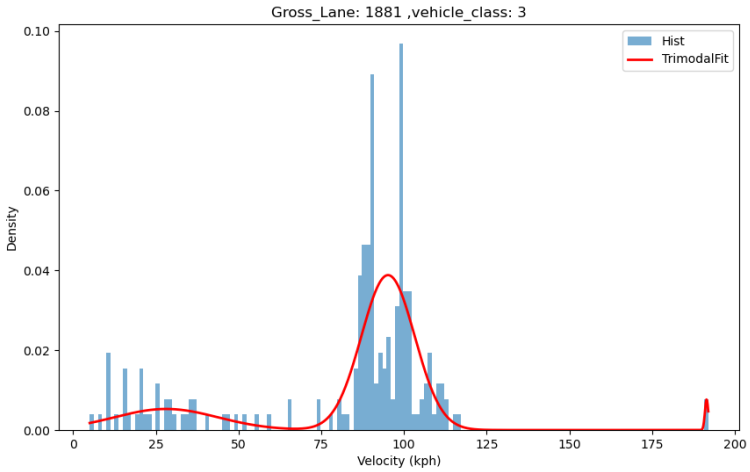


Figure A.38: Speed trimodal distribution for class 3 on lane 1881

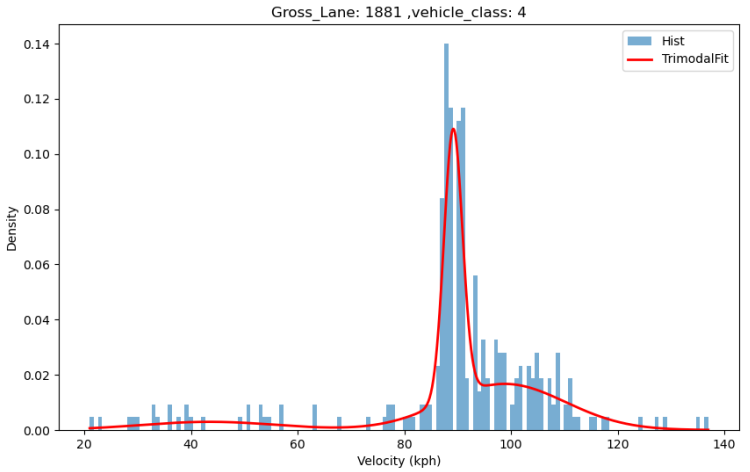


Figure A.39: Speed trimodal distribution for class 4 on lane 1881

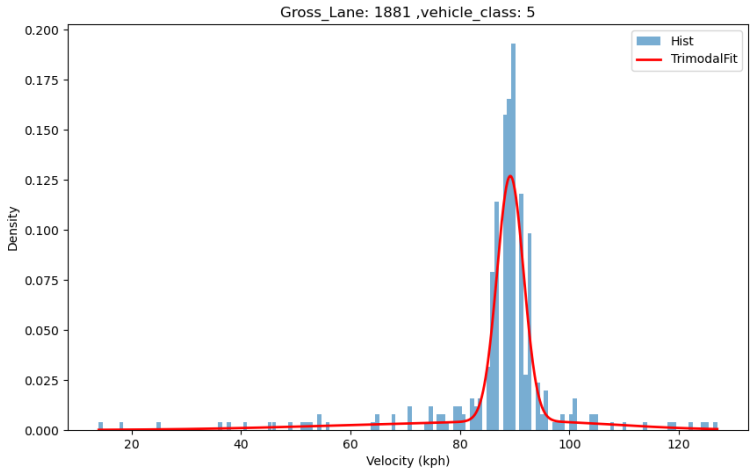


Figure A.40: Speed trimodal distribution for class 5 on lane 1881

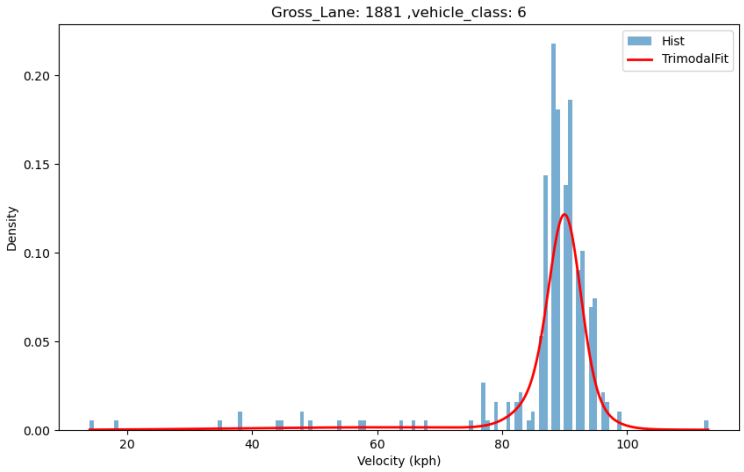


Figure A.41: Speed trimodal distribution for class 6 on lane 1881

Figure A.42: Speed trimodal distribution for class 7 on lane 1881

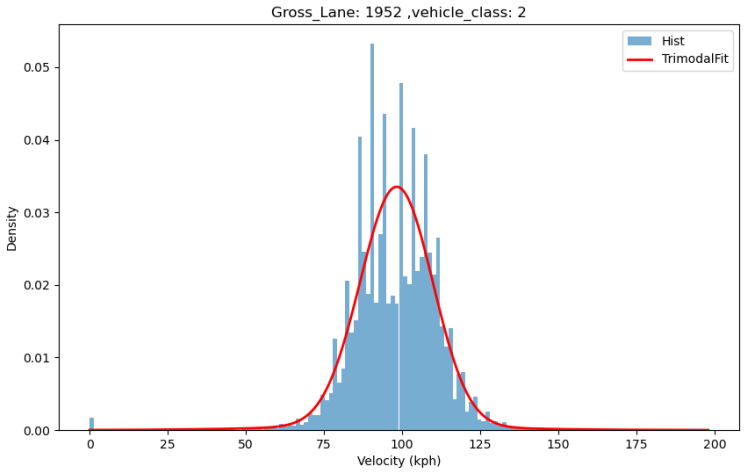


Figure A.43: Speed trimodal distribution for class 2 on lane 1952

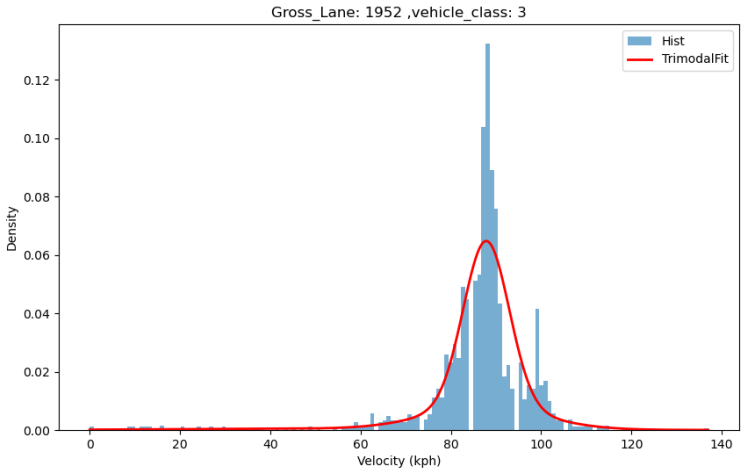


Figure A.44: Speed trimodal distribution for class 3 on lane 1952

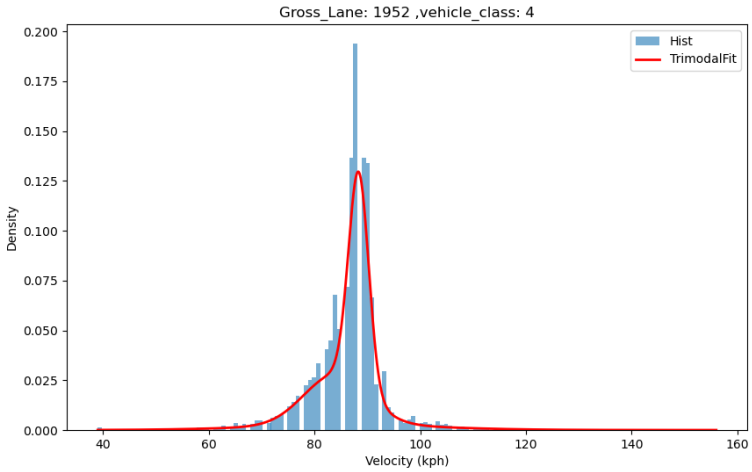


Figure A.45: Speed trimodal distribution for class 4 on lane 1952

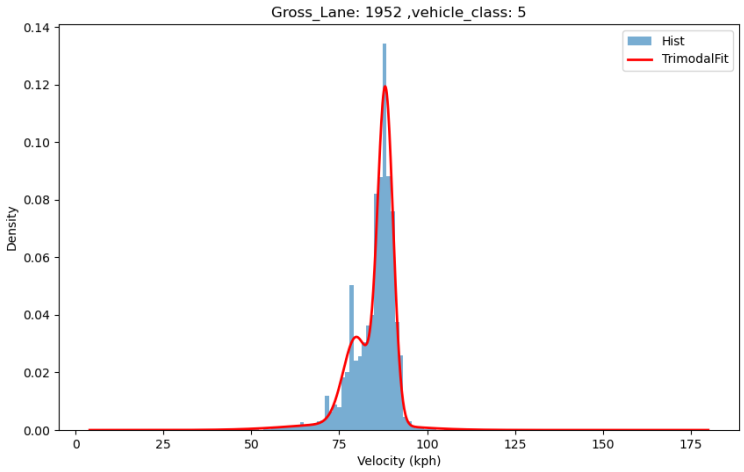


Figure A.46: Speed trimodal distribution for class 5 on lane 1952

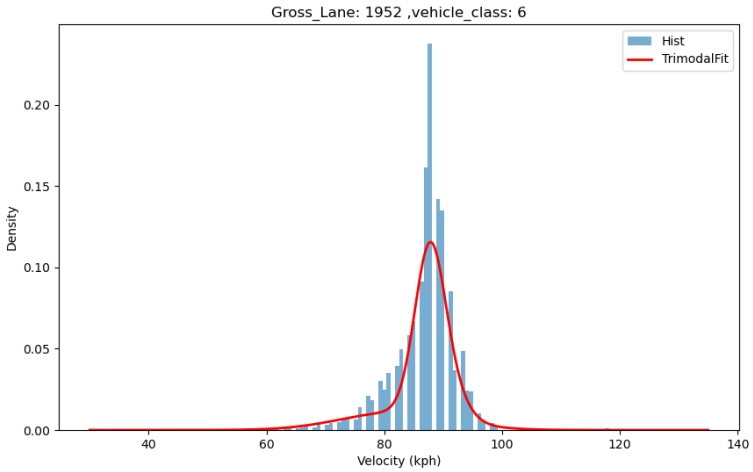


Figure A.47: Speed trimodal distribution for class 6 on lane 1952

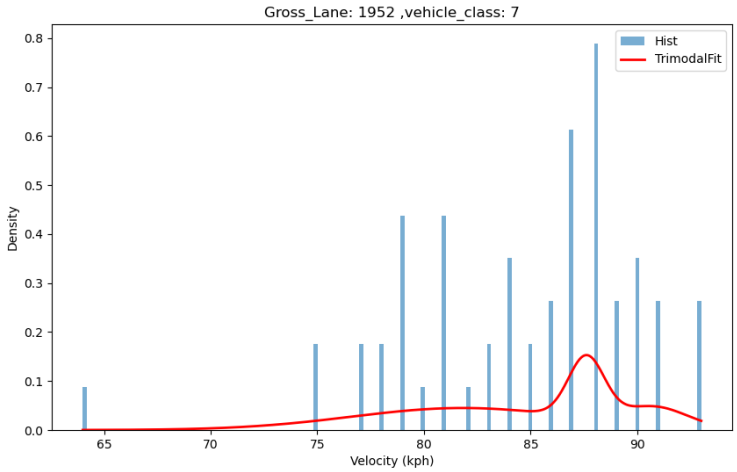


Figure A.48: Speed trimodal distribution for class 7 on lane 1952

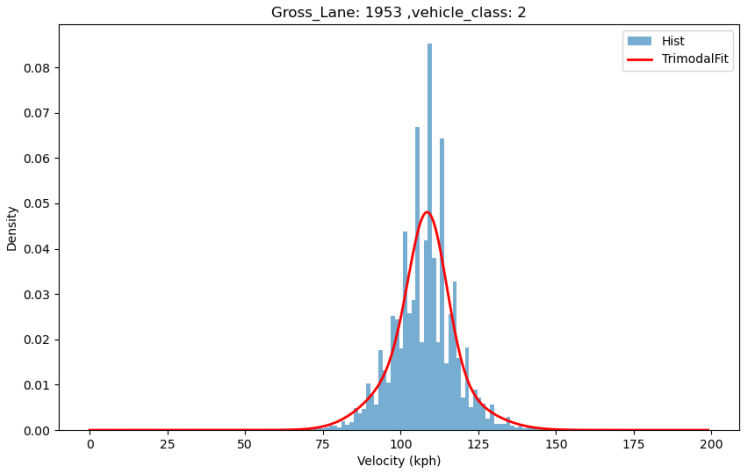


Figure A.49: Speed trimodal distribution for class 2 on lane 1953

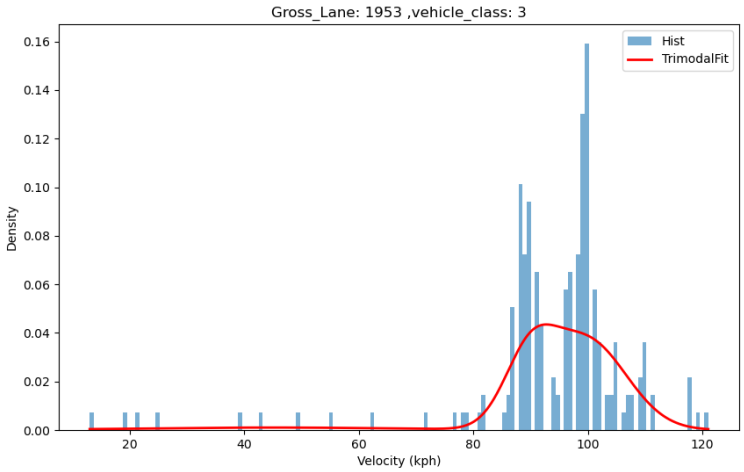


Figure A.50: Speed trimodal distribution for class 3 on lane 1953

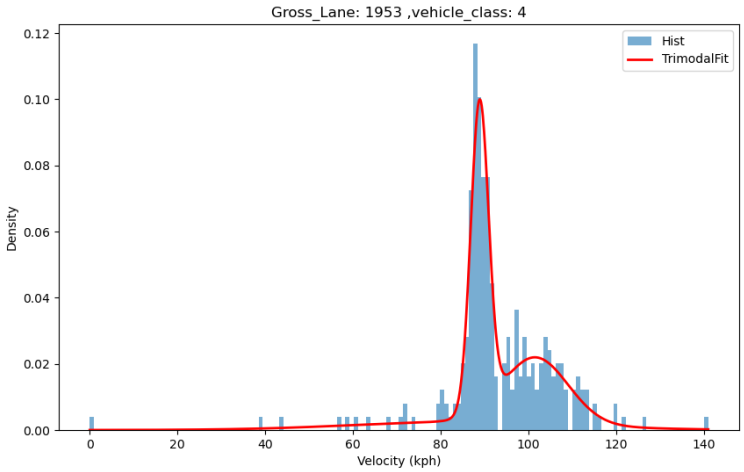


Figure A.51: Speed trimodal distribution for class 4 on lane 1953

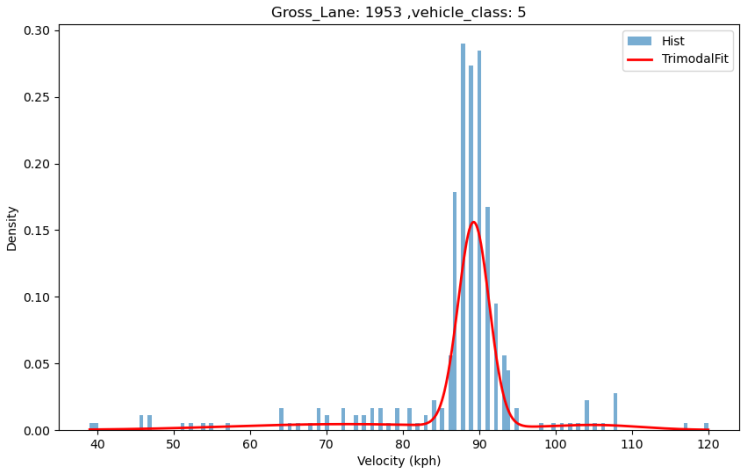


Figure A.52: Speed trimodal distribution for class 5 on lane 1953

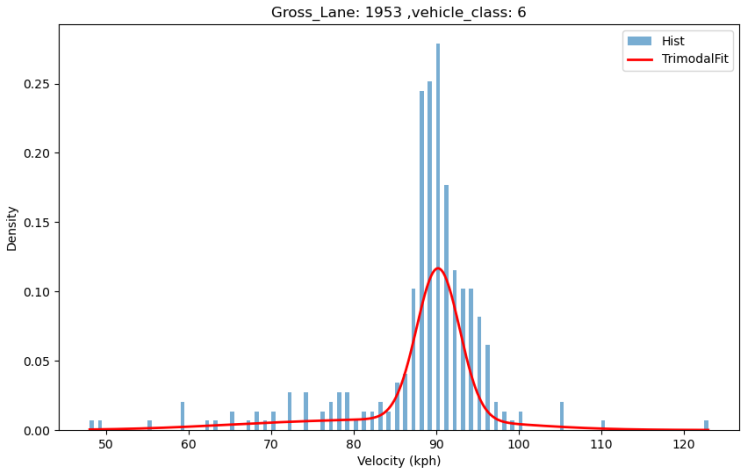


Figure A.53: Speed trimodal distribution for class 6 on lane 1953

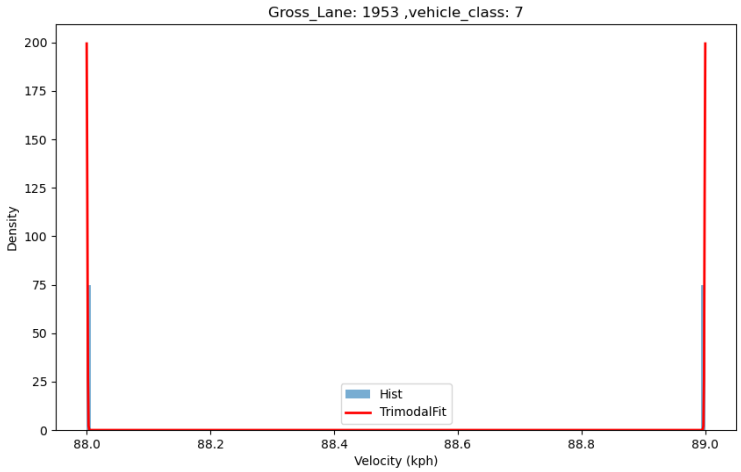


Figure A.54: Speed trimodal distribution for class 7 on lane 1953

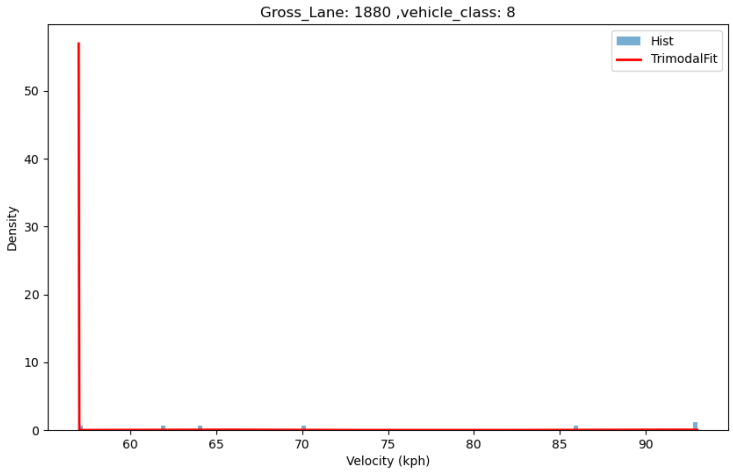


Figure A.55: Speed trimodal distribution for class 8 on lane 1880

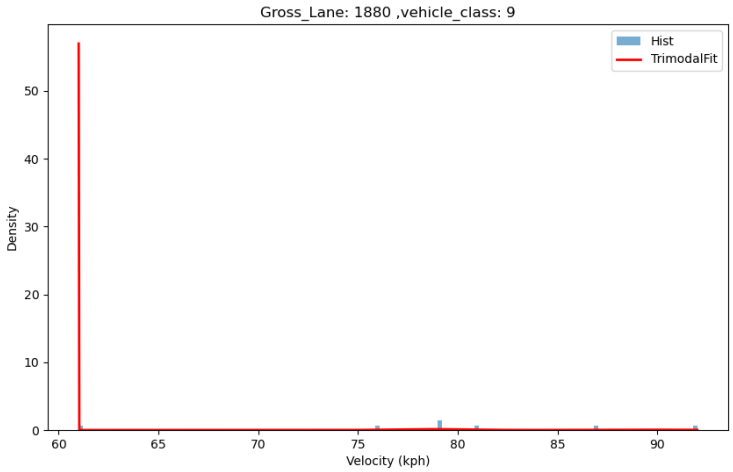


Figure A.56: Speed trimodal distribution for class 9 on lane 1880

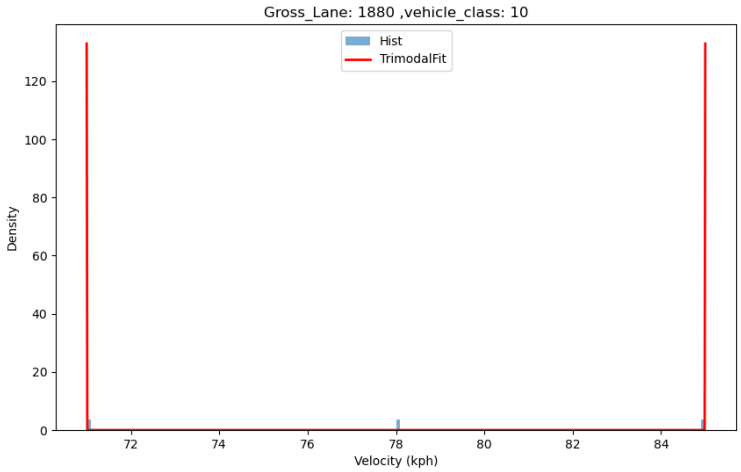


Figure A.57: Speed trimodal distribution for class 10 on lane 1880

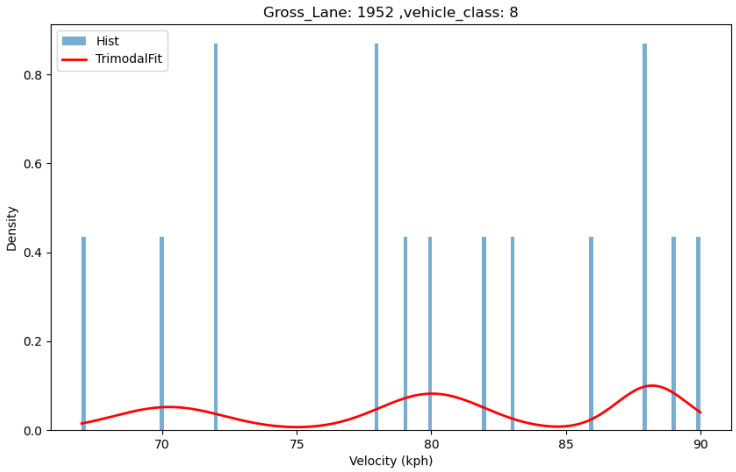


Figure A.58: Speed trimodal distribution for class 8 on lane 1952

Figure A.59: Speed trimodal distribution for class 9 on lane 1952

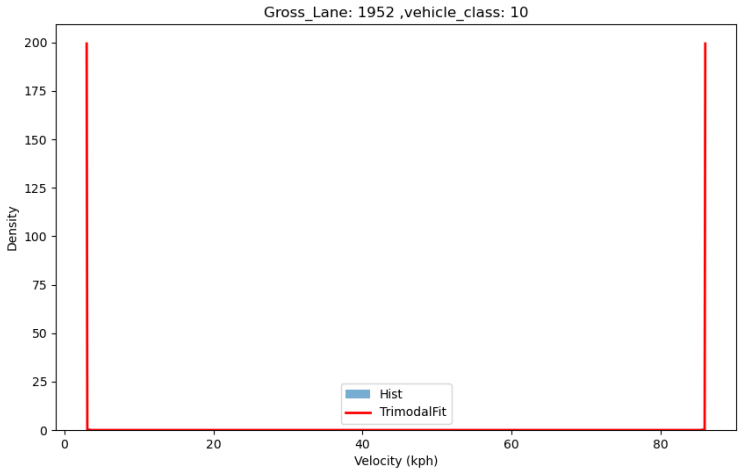


Figure A.60: Speed trimodal distribution for class 10 on lane 1952

Table A.1: Trimodal distributions parameters for gross weight

lane	axles	μ_1	σ_1	$w1$	μ_2	σ_2	w_2	μ_3	σ_3	w_3
1880	2	1.314	0.320	0.717	7.184	3.722	0.061	2.330	0.610	0.223
	3	16.599	3.517	0.490	3.181	0.934	0.187	14.514	3.013	0.323
	4	21.312	5.132	0.525	13.868	2.201	0.351	4.151	1.053	0.124
	5	22.242	2.680	0.441	14.722	2.011	0.412	21.351	8.891	0.147
	6	36.990	2.528	0.423	21.001	4.791	0.264	27.718	3.788	0.313
	7	21.902	1.660	0.252	49.713	9.936	0.290	36.984	5.886	0.458
	8	120.800	0.001	0.143	48.275	1.163	0.571	76.300	2.100	0.286
	9	125.600	0.001	0.143	36.067	9.253	0.429	85.233	0.602	0.429
	10	57.700	0.001	0.333	86.100	0.001	0.333	53.300	0.001	0.333
1881	2	1.334	0.317	0.777	2.212	0.563	0.215	4.439	3.684	0.008
	3	2.606	1.406	0.428	17.448	3.047	0.190	13.892	2.381	0.382
	4	4.004	1.557	0.450	24.057	4.236	0.197	14.396	3.127	0.353
	5	16.103	3.086	0.589	24.708	4.682	0.295	4.868	1.735	0.116
	6	16.986	5.486	0.144	35.981	4.953	0.565	24.822	3.351	0.290
	7	35.300	0.001	0.500	0.260	0.001	0.500	0.000	0.001	0.000
1952	2	1.956	0.769	0.325	1.285	0.261	0.622	7.097	3.659	0.053
	3	15.399	3.418	0.564	3.131	0.966	0.253	17.502	7.346	0.183
	4	27.592	4.169	0.160	4.110	1.207	0.153	15.916	2.775	0.687
	5	20.103	3.600	0.410	16.997	2.651	0.412	22.035	9.057	0.178
	6	32.139	3.693	0.288	21.668	4.041	0.255	41.664	2.740	0.456
	7	59.161	6.053	0.311	28.056	5.114	0.381	41.456	3.838	0.308
	8	46.586	6.754	0.140	80.363	8.283	0.468	29.128	2.447	0.392
	9	91.400	0.001	0.500	47.600	0.001	0.500	0.000	0.001	0.000
	10	47.000	0.001	0.500	2.590	0.001	0.500	0.000	0.001	0.000
1953	2	2.102	0.503	0.221	3.483	3.647	0.012	1.310	0.296	0.767
	3	15.618	2.375	0.506	2.922	1.068	0.375	9.083	1.524	0.119
	4	14.813	2.778	0.500	3.450	1.274	0.434	25.504	5.260	0.066
	5	6.018	2.236	0.096	16.048	2.625	0.657	22.363	4.956	0.247
	6	38.232	3.602	0.489	18.252	4.624	0.198	26.858	3.810	0.313
	7	25.600	0.001	0.500	45.000	0.001	0.500	0.000	0.001	0.000

Table A.2: Trimodal distributions parameters for speed

lane	axles	μ_1	σ_1	w_1	μ_2	σ_2	w_2	μ_3	σ_3	w_3
1880	2	106.873	10.253	0.470	43.711	16.061	0.054	90.568	8.822	0.475
	3	88.196	5.218	0.755	65.952	32.922	0.086	80.563	8.865	0.159
	4	88.390	2.035	0.695	80.303	21.027	0.133	80.454	3.372	0.172
	5	80.176	3.707	0.247	70.277	20.039	0.055	88.297	2.060	0.698
	6	88.513	2.761	0.741	80.654	4.453	0.218	69.681	24.684	0.041
	7	81.067	2.788	0.345	55.598	17.744	0.150	88.102	1.470	0.505
	8	57.000	0.001	0.143	90.667	3.300	0.429	65.333	3.399	0.429
	9	61.000	0.001	0.143	78.749	1.786	0.571	89.488	2.519	0.286
	10	85.000	0.001	0.333	71.000	0.001	0.333	78.000	0.001	0.333
1881	2	109.059	13.223	0.467	44.474	15.362	0.087	107.588	6.804	0.445
	3	28.112	15.596	0.207	95.190	8.051	0.784	191.500	0.500	0.010
	4	98.802	11.431	0.479	43.473	12.953	0.097	89.168	1.737	0.424
	5	75.528	24.268	0.177	89.191	2.430	0.745	95.908	13.754	0.077
	6	89.051	4.788	0.334	61.274	23.427	0.090	90.062	2.454	0.576
	7	88.000	0.001	0.500	3.000	0.001	0.500	0.000	0.001	0.000
1952	2	95.265	11.152	0.598	102.744	10.524	0.358	94.041	33.727	0.045
	3	87.395	12.721	0.241	87.832	5.046	0.720	52.390	32.927	0.039
	4	88.354	1.903	0.514	84.001	6.000	0.379	86.031	16.647	0.108
	5	79.812	3.678	0.277	88.132	2.205	0.637	77.587	15.359	0.086
	6	87.685	2.564	0.632	91.093	2.950	0.118	81.883	9.258	0.250
	7	87.614	0.845	0.263	81.824	5.223	0.589	90.835	1.540	0.147
	8	88.211	1.323	0.331	70.253	2.053	0.267	80.028	1.958	0.402
	9	56.000	0.001	0.500	79.000	0.001	0.500	0.000	0.001	0.000
	10	86.000	0.001	0.500	3.000	0.001	0.500	0.000	0.001	0.000
1953	2	108.787	6.004	0.527	101.188	11.494	0.294	116.170	12.329	0.179
	3	89.708	4.419	0.290	99.694	7.005	0.652	44.639	22.757	0.058
	4	85.052	25.209	0.154	88.884	2.029	0.473	101.608	7.439	0.373
	5	105.383	6.012	0.051	89.278	2.029	0.781	72.582	15.096	0.168
	6	91.175	10.264	0.141	74.051	11.508	0.144	90.248	2.605	0.714
	7	88.000	0.001	0.500	89.000	0.001	0.500	0.000	0.001	0.000

Appendix B

Papers & Awards

B.1 Papers

Xu, D., Chong, H., Main, I., Mineter, M., De Bold, R., Forde, M.C., Gair, C., Madden, P., Angus, E. and Ho, C., 2019. Using statistical models and machine learning techniques to process big data from the forth road bridge. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making* (pp. 411-419). ICE Publishing.

Xu, D., Xu, X., Forde, M.C. and Caballero, A., 2023. Concrete and steel bridge Structural Health Monitoring-Insight into choices for machine learning applications. In *Construction and Building Materials*, 402, p.132596.

Xu, X., Xu, D., Caballero, A., Ren, Y., Huang, Q., Chang, W. and Forde, M.C., 2023, August. Vehicle-induced deflection prediction using long short-term memory networks. In *Structures* (Vol. 54, pp. 596-606). Elsevier.

Xu X., Xu D., et al., 2023, Sensor Optimization Using Digital Twins Based on The Case Studies of Two Cable Stayed Bridges in the UK and China. In *Bridge Engineering*. ICE Publishing. (Under review)

B.2 Awards

Charles M. Vest Grand Challenge Scholarship, 2019 - National Academy of Engineering, USA; Royal Society of Engineering, UK; Massachusetts Institute of Technology, USA

References

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, 154–170. doi: 10.1016/j.jsv.2016.10.043

Administration, U. S. P. W. (1939). *America builds: The record of pwa. public works administration* (No. p. 96). U.S. Government Printing Office.

Alipour, M., Harris, D. K., & Miller, G. R. (2019). Robust pixel-level crack detection using deep fully convolutional neural networks. *Journal of Computing in Civil Engineering*, *33*(6), 04019040. doi: 10.1061/(ASCE)CP.1943-5487.0000854

Baidar, B., & Aftab, M. (2015). *Bridges: Analysis, Design, Structural Health Monitoring, and Rehabilitation*. Springer International Publishing.

Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, *18*(2), 401–421. doi: 10.1177/1475921718757405

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828. doi: 10.1109/SECON.2016.7506650

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166. doi: 10.1109/72.279181

Bham, G. H., & Benekohal, R. F. (2004). A high fidelity traffic simulation model based on cellular automata and car-following concepts. *Transportation Research Part C: Emerging Technologies*, *12*(1), 1–32. doi: 10.1016/j.trc.2002.05.001

Bianco, A. M., Garcia Ben, M., Martinez, E., & Yohai, V. J. (2001). Outlier detection in regression models with arima errors using robust estimates. *Journal of Forecasting*, *20*(8), 565–579. doi: 10.1002/for.768

Bishop, C. M., et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

Boller, C., & Meyendorf, N. (2008). State-of-the-art in structural health monitoring for aeronautics. In *Proceedings of the international symposium on NDT in aerospace* (pp. 1–8).

Brownjohn, J., Koo, K.-Y., Scullion, A., & List, D. (2015). Operational deformations in long-span bridges. *Structure and Infrastructure Engineering*, *11*(4), 556–574. doi: 10.1080/15732479.2014.951857

Brownjohn, J. M. W., Moyo, P., Omenzetter, P., & Lu, Y. (2003). Assessment of highway bridge upgrading by dynamic testing and finite-element model updating. *Journal of Bridge Engineering*, *8*(3), 162–172. doi: 10.1061/(ASCE)1084-0702(2003)8:3(162)

BSI. (2004). *Eurocode 1: Actions on structures. traffic loads on bridges*. Author.

BSI. (2006). Steel, concrete and composite bridges: Specification for loads. *BS 5400-2*.

Bu, G., Chanda, S., Guan, H., Jo, J., Blumenstein, M., & Loo, Y. (2015). Crack detection using a texture analysis-based technique for visual bridge inspection. *Electronic Journal of Structural Engineering*, *14*(1), 41–48. doi: 10.1109/ACCESS.2019.2961755

Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical mechanics and its applications*, *519*, 127–139. doi: 10.1016/j.physa.2018.11.061

Catbas, F. N., Celik, O., Avci, O., Abdeljaber, O., Gul, M., & Do, N. T. (2017). Sensing and monitoring for stadium structures: a review of recent advances and a forward look. *Frontiers in built environment*, *3*, 38. doi: 10.3389/fbuil.2017.00038

Çelebi, M. (2012). Golden gate bridge response: A study with low-amplitude data from three earthquakes. *Earthquake spectra*, *28*(2), 487–510. doi: 10.1193/1.4000018

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378. doi: 10.1111/mice.12263

Chang, C., Chang, T., & Zhang, Q. (2001). Ambient vibration of long-span cable-stayed bridge. *Journal of bridge engineering*, *6*(1), 46–53. doi: 10.1061/(ASCE)1084-0702(2001)6:1(46)

Chellapilla, K., Puri, S., & Simard, P. (2006). High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition.*

Chen, F., & Jahanshahi, M. (2017). NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. *IEEE Transactions on Industrial Electronics*, *65*(5), 4392–4400. doi: 10.1109/TIE.2017.2764844

Chen, S.-Z., Wu, G., & Feng, D.-C. (2019). Damage detection of highway bridges based on long-gauge strain response under stochastic traffic flow. *Mechanical Systems and Signal Processing*, *127*, 551–572. doi: 10.1016/j.ymssp.2019.03.022

Chen, Z., Li, H., Bao, Y., Li, N., & Jin, Y. (2016). Identification of spatio-temporal distribution of vehicle loads on long-span bridges using computer vision technology. *Structural Control and Health Monitoring*, *23*(3), 517–534. doi: 10.1002/stc.1780

Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems* (pp. 161–168). doi: 10.1145/3338840.3355641

Choi, M.-Y., & Kwon, I.-b. (2000). Damage detection system of a real steel truss bridge by neural networks. In *Smart structures and materials 2000: Smart systems for bridges, structures, and highways* (Vol. 3988, pp. 295–306). doi: 10.1117/12.383151

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, *22*(12), 3207–3220. doi: 10.1162/NECO_a_00052

CMOT. (2015). *General specifications for design of highway bridges and culverts*. China Communications Press.

Colford, B., & Cocksedge, C. (2008). *Forth Road Bridge - first internal inspection, strength evaluation, acoustic monitoring and dehumidification of the main cables.* `https://web.archive.org/web/20081218133741/http://www.feta.gov.uk/download/files/firstinsp.pdf`. ([Online; accessed 30-Aug-2023])

Colford, B., Jones, S., Timby, D., & Brown, K. (2009). The maintenance of the main expansion joints on the Forth Road Suspension Bridge, Scotland. In *Safety and Reliability of Bridge Structures (Ed. K. Mahmoud)* (pp. 353–366). CRC Press.

Concepcion, R. S., & Ilagan, L. C. (2019). Application of hybrid soft computing for classification of reinforced concrete bridge structural health based on thermal-vibration intelligent system parameters. In *2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 207–212). doi: 10.1109/CSPA.2019.8696007

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297. doi: 10.1007/BF00994018

Cousins, D. P., McAra, D., & Hill, C. (2022). Monitoring of the Queensferry Crossing, scotland. In *Proceedings of the institution of civil engineers-bridge engineering* (pp. 1–15). doi: 10.1680/jbren.22.00018

De Miranda, M. (2016). Chapter 15 - long-span bridges. In A. Pipinato (Ed.), *Innovative bridge design handbook* (p. 383-425). Boston: Butterworth-Heinemann. doi: 10.1016/B978-0-12-800058-8.00015-3

de Almeida Cardoso, R., Cury, A., & Barbosa, F. (2019). Automated real-time damage detection strategy using raw dynamic measurements. *Engineering Structures*, *196*, 109364. doi: 10.1016/j.engstruct.2019.109364

Deng, L., Wang, W., & Cai, C. (2017). Effect of pavement maintenance cycle on the fatigue reliability of simply-supported steel i-girder bridges under dynamic vehicle loading. *Engineering Structures*, *133*, 124–132. doi: 10.1016/j.engstruct.2016.12.022

Deng, L., Yu, Y., Zou, Q., & Cai, C. (2015). State-of-the-art review of dynamic impact factors of highway bridges. *Journal of Bridge Engineering*, *20*(5), 04014080. doi: 10.1061/(asce)be.1943-5592.0000672

Deng, Y., Li, A., Liu, Y., & Chen, S. (2018). Investigation of temperature actions on flat steel box girders of long-span bridges with temperature monitoring data. *Advances in Structural Engineering*, *21*(14), 2099–2113. doi: 10.1177/1369433218766946

Deng, Y., Zhang, M., Feng, D.-M., & Li, A.-Q. (2021). Predicting fatigue damage of highway suspension bridge hangers using weigh-in-motion data and machine learning. *Structure and Infrastructure Engineering*, *17*(2), 233–248. doi: 10.1080/15732479.2020.1734632

Dexter, R. J. (2002). *Performance testing for modular bridge joint systems* (Vol. 467). Transportation Research Board.

Diez, A., Khoa, N. L. D., Makki Alamdari, M., Wang, Y., Chen, F., & Runcie, P. (2016). A clustering approach for structural health monitoring on bridges. *Journal of Civil Structural Health Monitoring*, *6*(3), 429–445. doi: 10.1007/s13349-016-0160-0

Ding, Y., Zhou, G., Li, A., & Wang, G. (2012). Thermal field characteristic analysis of steel box girder based on long-term measurement data. *International Journal of Steel Structures*, *12*, 219–232. doi: 10.1007/s13296-012-2006-x

Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. *Expert Systems with Applications*, *166*, 114060. doi: 10.1016/j.eswa.2020.114060

Dong, G., & Liu, H. (2018). *Feature engineering for machine learning and data analytics*. CRC Press.

Dubin, E. E., & Yanev, B. S. (2001). Managing the East River bridges in New York City. *7th International Symposium on Smart Structures and Materials*(August 2001), 60–74. doi: 10.1117/12.435629

Dung, C. V., et al. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*, 52–58. doi: 10.1016/j.autcon.2018.11.028

Eltouny, K., Gomaa, M., & Liang, X. (2023). Unsupervised learning methods for data-driven vibration-based structural health monitoring: A review. *Sensors*, *23*(6), 3290. doi: 10.3390/s23063290

Farrar, C., & Worden, K. (2012). *Structural health monitoring: a machine learning perspective*. John Wiley & Sons.

Farrar, C. R., Baker, W., Bell, T., Cone, K., Darling, T., Duffey, T., ... Migliori, A. (1994). *Dynamic characterization and damage detection in the i-40 bridge over the rio grande* (Tech. Rep.). Los Alamos National Lab., NM (United States).

Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 303–315. doi: 10.1098/rsta.2006.1928

Friswell, M., & Mottershead, J. E. (2013). *Finite element model updating in structural dynamics* (Vol. 38). Springer Science & Business Media.

Fujino, Y., Siringoringo, D. M., Ikeda, Y., Nagayama, T., & Mizutani, T. (2019). Research and implementations of structural monitoring for bridges and buildings in japan. *Engineering*, *5*(6), 1093–1119. doi: 10.1016/j.eng.2019.09.006

Getachew, A., & Obrien, E. J. (2007). Simplified site-specific traffic load models for bridge assessment. *Structure and infrastructure Engineering*, *3*(4), 303–311. doi: 10.1080/15732470500424245

Gordan, M., Razak, H. A., Ismail, Z., Ghaedi, K., Tan, Z. X., & Ghayeb, H. H. (2020). A hybrid ann-based imperial competitive algorithm methodology for structural damage identification of slab-on-girder bridge using data mining. *Applied Soft Computing*, *88*, 106013. doi: 10.1016/j.asoc.2019.106013

Görl, E., & Link, M. (2003). Damage identification using changes of eigenfrequencies and mode shapes. *Mechanical systems and signal processing*, *17*(1), 103–110. doi: 10.1006/mssp.2002.1545

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, *18*(5-6), 602–610.

Gudivada, V., Apon, A., & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, *10*(1), 1–20.

Gumbel, E. J. (1958). *Statistics of extremes*. Columbia university press.

Guo, A., Jiang, A., Lin, J., & Li, X. (2020). Data mining algorithms for bridge health monitoring: Kohonen clustering and lstm prediction approaches. *The Journal of Supercomputing*, *76*(2), 932–947. doi: 10.1007/s11227-019-03045-8

Guo, T., Liu, J., Zhang, Y., & Pan, S. (2015). Displacement monitoring and analysis of expansion joints of long-span steel bridges with viscous dampers. *Journal of Bridge Engineering*, *20*(9), 04014099. doi: 10.1061/(ASCE)BE.1943-5592.0000701

Hakim, S., & Razak, H. A. (2013). Structural damage detection of steel bridge girder using artificial neural networks and finite element models. *Steel Compos. Struct*, *14*(4), 367–377. doi: 10.12989/scs.2013.14.4.367

He, H.-X., & Yan, W.-m. (2007). Structural damage detection with wavelet support vector machine: introduction and applications. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, *14*(1), 162–176. doi: 10.1002/stc.150

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278–282). doi: 10.1109/ICDAR.1995.598994

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116. doi: 10.1142/S0218488598000094

Hochreiter, S., & Schmidhuber, J. (1997, nov). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Inaudi, D. (2010). Overview of 40 bridge structural health monitoring projects. In *International bridge conference, ibc* (pp. 15–17).

Jeong, S., Ferguson, M., Hou, R., Lynch, J. P., Sohn, H., & Law, K. H. (2019). Sensor data reconstruction using bidirectional recurrent neural network with application to bridge monitoring. *Advanced engineering informatics*, *42*, 100991. doi: 10.1016/j.aei.2019.100991

Johnson, T. (2023). *Traffic diversion only remedy to Queensferry Crossing's falling ice woes.* `https://www.newcivilengineer.com/latest/traffic-diversion-only-remedy-to-queensferry-crossings-falling-ice-woes-after-many-solutions-tested-03-07-2023/`. New Civil Engineer. ([Online; accessed 30-Aug-2023])

Kang, D., Ko, K., & Huh, J. (2015). Determination of extreme wind values using the gumbel distribution. *Energy*, *86*, 51–58. doi: 10.1016/j.energy.2015.03.126

Koo, K.-Y., Brownjohn, J., List, D., & Cole, R. (2013). Structural health monitoring of the tamar suspension bridge. *Structural Control and Health Monitoring*, *20*(4), 609–625. doi: 10.1002/stc.1481

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. doi: 10.1145/3065386

Kruachottikul, P., Cooharojananone, N., Phanomchoeng, G., Chavarnakul, T., Kovitanggoon, K., & Trakulwaranont, D. (2021). Deep learning-based visual defect-inspection system for reinforced concrete bridge substructure: a case of thailand's department of highways. *Journal of Civil Structural Health Monitoring*, *11*(4), 949–965. doi: 10.1007/s13349-021-00490-z

Kurata, S., & Nagatani, T. (2003). Spatio-temporal dynamics of jams in two-lane traffic flow with a blockage. *Physica A: Statistical Mechanics and its Applications*, *318*(3-4), 537–550. doi: 10.1016/S0378-4371(02)01376-6

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444. doi: 10.1038/nature14539

Lei, X., Sun, L., Xia, Y., & He, T. (2020). Vibration-based seismic damage states evaluation for regional concrete beam bridges using random forest method. *Sustainability*, *12*(12), 5106. doi: 10.3390/su12125106

Li, B., Ushiroda, K., Yang, L., Song, Q., & Xiao, J. (2017). Wall-climbing robot for non-destructive evaluation using impact-echo and metric learning svm. *International Journal of Intelligent Robotics and Applications*, *1*(3), 255–270. doi: 10.1007/s41315-017-0028-4

Li, L., Chen, B., Zhou, L., Xia, Q., Zhou, Y., Zhou, X., & Xia, Y. (2023). Thermal behaviors of bridges—a literature review. *Advances in Structural Engineering*, *26*(6), 985–1010. doi: 10.1177/13694332231153976

Li, S., Li, S., Laima, S., & Li, H. (2021). Data-driven modeling of bridge buffeting in the time domain using long short-term memory network based on structural health monitoring. *Structural Control and Health Monitoring*, *28*(8), e2772. doi: 10.1002/stc.2772

Li, S., Zhu, S., Xu, Y.-L., Chen, Z.-W., & Li, H. (2012). Long-term condition assessment of suspenders under traffic loads based on structural monitoring system: application to the tsing ma bridge. *Structural Control and Health Monitoring*, *19*(1), 82–101. doi: 10.1002/stc.427

Li, T., Wu, T., & Liu, Z. (2020). Nonlinear unsteady bridge aerodynamics: Reduced-order modeling based on deep lstm networks. *Journal of Wind Engineering and Industrial Aerodynamics*, *198*, 104116. doi: 10.1016/j.jweia.2020.104116

Liu, G., Niu, Y., Zhao, W., Duan, Y., & Shu, J. (2022). Data anomaly detection for structural health monitoring using a combination network of ganomaly and cnn. *Smart Structures and Systems*, *29*(1), 53–62. doi: 10.12989/sss.2022.29.1.053

Liu, M., Frangopol, D. M., & Kim, S. (2009). Bridge system performance assessment from structural health monitoring: A case study. *Journal of Structural Engineering*, *135*(6), 733–742. doi: 10.1061/(ASCE)ST.1943-541X.0000014

Liu, Y., Zhang, H., Liu, Y., Deng, Y., Jiang, N., & Lu, N. (2017). Fatigue reliability assessment for orthotropic steel deck details under traffic flow and temperature loading. *Engineering Failure Analysis*, *71*, 179–194. doi: 10.1016/j.engfailanal.2016.11.007

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, *32*(23), 17351–17360. doi: 10.1007/s00521-020-04867-x

Mahmoud, K. (2009). *Safety and reliability of bridge structures*. CRC Press.

Manuello, A., Niccolini, G., & Carpinteri, A. (2019). Ae monitoring of a concrete arch road tunnel: Damage evolution and localization. *Engineering Fracture Mechanics*, *210*, 279–287. doi: 10.1016/j.engfracmech.2018.07.029

Martino, N., Maser, K., Birken, R., & Wang, M. (2016). Quantifying bridge deck corrosion using ground penetrating radar. *Research in Nondestructive Evaluation*, *27*(2), 112–124. doi: 10.1080/09349847.2015.1067342

Matarazzo, T. J., Kondor, D., Milardo, S., Eshkevari, S. S., Santi, P., Pakzad, S. N., ... Ratti, C. (2022). Crowdsourcing bridge dynamic monitoring with smartphone vehicle trips. *Communications engineering*, *1*(1), 29. doi: 10.1038/s44172-022-00025-4

Mehrjoo, M., Khaji, N., Moharrami, H., & Bahreininejad, A. (2008). Damage detection of truss bridge joints using artificial neural networks. *Expert systems with applications*, *35*(3), 1122–1131. doi: 10.1016/j.eswa.2007.08.008

Miao, C., & Shi, C. (2013). Temperature gradient and its effect on flat steel box girder of long-span suspension bridge. *Science China Technological Sciences*, *56*, 1929–1939. doi: 10.1007/s11431-013-5280-8

Min, J., Park, S., & Yun, C.-B. (2010). Impedance-based structural health monitoring using neural networks for autonomous frequency range selection. *Smart Materials and Structures*, *19*(12), 125011. doi: 10.1088/0964-1726/19/12/125011

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, *42*(11), 30–36. doi: 10.1145/319382.319388

Modarres, C., Astorga, N., Droguett, E. L., & Meruane, V. (2018). Convolutional neural networks for automated damage recognition and damage type identification. *Structural Control and Health Monitoring*, *25*(10), e2230. doi: 10.1002/stc.2230

Moore, E. Z., Nichols, J. M., & Murphy, K. D. (2012). Model-based shm: Demonstration of identification of a crack in a thin plate using free vibration data. *Mechanical systems and signal processing*, *29*, 284–295. doi: 10.1016/j.ymssp.2011.09.022

Mousavi, M., & Gandomi, A. H. (2021). Structural health monitoring under environmental and operational variations using mcd prediction error. *Journal of Sound and Vibration*, *512*, 116370. doi: 10.1016/j.jsv.2021.116370

Munawar, H. S., Hammad, A. W., Haddad, A., Soares, C. A. P., & Waller, S. T. (2021). Image-based crack detection methods: A review. *Infrastructures*, *6*(8), 115. doi: 10.3390/infrastructures6080115

National Transportation Safety Board. (1989, June). *Aircraft accident report–aloha airlines, flight 243, boeing 737-200, n73711, near maui, hawaii, april 28, 1988* (Tech. Rep. No. NTSB/AAR-89/03). Washington, D.C. 20594: National Transportation Safety Board.

Neves, A., González, I., Leander, J., & Karoumi, R. (2017). Structural health monitoring of bridges: a model-free ann-based approach to damage detection. *Journal of Civil Structural Health Monitoring*, *7*(5), 689–702. doi: 10.1007/s13349-017-0252-5

Nguyen, H. D., Tran, H., Bui-Tien, T., De Roeck, G., & Abdel Wahab, M. (2020). Damage detection in truss bridges using transmissibility and machine learning algorithm: Application to nam o bridge. *Smart Structures and Systems*, *26*(1), 35–47. doi: 10.12989/sss.2020.26.1.035

Ni, F., Zhang, J., & Noori, M. N. (2019, dec). Deep learning for data anomaly detection and data compression of a long-span suspension bridge. *Computer-Aided Civil and Infrastructure Engineering*, *35*(7), 685–700. doi: 10.1111/mice.12528

Okasha, N. M., Frangopol, D. M., & Orcesi, A. D. (2012). Automated finite element updating using strain data for the lifetime reliability assessment of bridges. *Reliability Engineering & System Safety*, *99*, 139–150. doi: 10.1016/j.ress.2011.11.007

Oliveira, S., & Alegre, A. (2019). Seismic and structural health monitoring of dams in portugal. *Seismic structural health monitoring: from theory to successful applications*, 87–113. doi: 10.1007/978-3-030-13976-6_4

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. doi: 10.48550/arXiv.1609.03499

Palutikof, J. P., Brabson, B., Lister, D. H., & Adcock, S. (1999). A review of methods to calculate extreme wind speeds. *Meteorological applications*, *6*(2), 119–132. doi: 10.1017/S1350482799001103

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. doi: 10.1061/(ASCE)BE.1943-5592.0001199

Pan, Y., Wang, D., Dong, Y., & Peng, B. (2021). A novel vision-based framework for identifying dynamic vehicle loads on long-span bridges: A case study of jiangyin bridge, china. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 10441–10457. doi: 10.1109/TITS.2021.3094504

Pawar, P. M., Venkatesulu Reddy, K., & Ganguli, R. (2007). Damage detection in beams using spatial fourier analysis and neural networks. *Journal of Intelligent Material Systems and Structures*, *18*(4), 347–359. doi: 10.1177/1045389X06066292

Peeters, B., Maeck, J., & De Roeck, G. (2001). Vibration-based damage detection in civil engineering: excitation sources and temperature effects. *Smart materials and Structures*, *10*(3), 518. doi: 10.1088/0964-1726/10/3/314

Phares, B., Graybeal, B., Rolander, D., Moore, M., & Washer, G. (2001). Reliability and accuracy of routine inspection of highway bridges. *Transportation Research Record: Journal of the Transportation Research Board*(1749), 82–92. doi: 10.3141/1749-13

Pinheiro, E. C., & Ferrari, S. L. (2016). A comparative review of generalizations of the gumbel extreme value distribution with an application to wind speed data. *Journal of Statistical Computation and Simulation*, *86*(11), 2241–2261. doi: 10.1080/00949655.2015.1107909

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400. doi: 10.1002/tal.1400

Reynders, E., Teughels, A., & De Roeck, G. (2010). Finite element model updating and structural damage identification using omax data. *Mechanical Systems and Signal Processing*, *24*(5), 1306–1323. doi: 10.1016/j.ymssp.2010.03.014

Rigatti, S. J. (2017). Random forest. *Journal of Insurance Medicine*, *47*(1), 31–39. doi: 10.17849/insm-47-01-31-39.1

Rizzo, P., & Enshaeian, A. (2021). Challenges in bridge health monitoring: A review. *Sensors*, *21*(13), 4336. doi: 10.3390/s21134336

Rosales, M. B., Filipich, C. P., & Buezas, F. S. (2009). Crack detection in beam-like structures. *Engineering Structures*, *31*(10), 2257–2264. doi: 10.1016/j.engstruct.2009.04.007

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (Tech. Rep.). California Univ San Diego La Jolla Inst for Cognitive Science.

Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, *323*, 203–213. doi: 10.1016/j.neucom.2018.09.082

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*. doi: 10.48550/arXiv.1402.1128

Sakiyama, F. I. H., Lehmann, F., & Garrecht, H. (2021). Structural health monitoring of concrete structures using fibre-optic-based sensors: A review. *Magazine of concrete research*, *73*(4), 174–194. doi: 10.1680/jmacr.19.00185

Sanayei, M., Phelps, J. E., Sipple, J. D., Bell, E. S., & Brenner, B. R. (2011). Instrumentation, nondestructive testing, and finite-element model updating for bridge evaluation using strain measurements. *Journal of bridge engineering*, *17*(1), 130–138. doi: 10.1061/(ASCE)BE.1943-5592.0000228

Sarmadi, H., Entezami, A., Salar, M., & De Michele, C. (2021). Bridge health monitoring in environmental variability by new clustering and threshold estimation methods. *Journal of Civil Structural Health Monitoring*, *11*(3), 629–644. doi: 10.1007/s13349-021-00472-1

Schubert, E., Zimek, A., & Kriegel, H.-P. (2014). Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data mining and knowledge discovery*, *28*(1), 190–237. doi: 10.1007/s10618-012-0300-z

Schwenker, F., & Trentin, E. (2014). Pattern classification and clustering: A review of partially supervised learning approaches. *Pattern Recognition Letters*, *37*, 4–14. doi: 10.1016/j.patrec.2013.10.017

Shajihan, S. A. V., Wang, S., Zhai, G., & Spencer Jr, B. F. (2022). Cnn based data anomaly detection using multi-channel imagery for structural health monitoring. *Smart Structures and Systems*, *29*(1), 181–193. doi: 10.12989/sss.2022.29.1.181

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. doi: 10.1016/j.asoc.2019.105524

Siringoringo, D. M., & Fujino, Y. (2006). Experimental study of laser doppler vibrometer and ambient vibration for vibration-based damage detection. *Engineering Structures*, *28*(13), 1803–1815. doi: 10.1016/j.engstruct.2006.03.006

Sony, S., Gamage, S., Sadhu, A., & Samarabandu, J. (2022). Vibration-based multiclass damage detection and localization using long short-term memory networks. In *Structures* (Vol. 35, pp. 436–451). doi: 10.1016/j.istruc.2021.10.088

Soyoz, S., & Feng, M. Q. (2009). Long-term monitoring and identification of bridge structural parameters. *Computer-Aided Civil and Infrastructure Engineering*, *24*(2), 82–92. doi: 10.1111/j.1467-8667.2008.00572.x

Suh, G., & Cha, Y.-J. (2018). Deep faster r-cnn-based automated detection and localization of multiple types of damage. In *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2018* (Vol. 10598, pp. 197–204). doi: 10.1117/12.2295954

Taha, M. R., Noureldin, A., Osman, A., & El-Sheimy, N. (2004). Introduction to the use of wavelet multiresolution analysis for intelligent structural health monitoring. *Canadian Journal of Civil Engineering*, *31*(5), 719–731. doi: 10.1139/l04-022

Tang, Z., Chen, Z., Bao, Y., & Li, H. (2019). Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Structural Control and Health Monitoring*, *26*(1), e2296. doi: 10.1002/stc.2296

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, *3*(2), 334–340. doi: 10.1016/j.fcij.2018.10.003

Tran-Ngoc, H., Khatir, S., De Roeck, G., Bui-Tien, T., & Wahab, M. A. (2019). An efficient artificial neural network for damage detection in bridges and beam-like structures by improving training parameters using cuckoo search algorithm. *Engineering Structures*, *199*, 109637. doi: 10.1016/j.engstruct.2019.109637

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*. doi: 10.1155/2018/7068349

Wang, F.-Y., & Xu, Y.-L. (2019). Traffic load simulation for long-span suspension bridges. *Journal of Bridge Engineering*, *24*(5), 05019005. doi: 10.1061/(ASCE)BE.1943-5592.0001381

Wang, H., Li, A., Niu, J., Zong, Z., & Li, J. (2013). Long-term monitoring of wind characteristics at sutong bridge site. *Journal of Wind Engineering and Industrial Aerodynamics*, *115*, 39–47. doi: 10.1016/j.jweia.2013.01.006

Wu, J., & Li, Q. (2006). Structural parameter identification and damage detection for a steel structure using a two-stage finite element model updating method. *Journal of Constructional Steel Research*, *62*(3), 231–239. doi: 10.1016/j.jcsr.2005.07.003

Xia, Q., Zhang, J., Tian, Y., & Zhang, Y. (2017). Experimental study of thermal effects on a long-span suspension bridge. *Journal of Bridge Engineering*, *22*(7), 04017034. doi: 10.1061/(ASCE)BE.1943-5592.0001083

Xu, D., Chong, H., Main, I., Mineter, M., De Bold, R., Forde, M., ... Ho, C. (2019). Using statistical models and machine learning techniques to process big data from the forth road bridge. In *International conference on smart infrastructure and construction 2019 (icsic) driving data-informed decision-making* (pp. 411–419). doi: 10.1680/icsic.64669.411

Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., & Chen, X. (2019). Automatic bridge crack detection using a convolutional neural network. *Applied Sciences*, *9*(14), 2867. doi: 10.3390/app9142867

Xu, X., Forde, M. C., Ren, Y., & Huang, Q. (2021). A bayesian approach for site-specific extreme load prediction of large scale bridges. *Structure and Infrastructure Engineering*, 1–14. doi: 10.1080/15732479.2021.2021953

Xu, X., Huang, Q., Ren, Y., Zhao, D.-Y., Yang, J., & Zhang, D.-Y. (2019). Modeling and separation of thermal effects from cable-stayed bridge response. *Journal of Bridge Engineering*, *24*(5), 04019028. doi: 10.1061/(ASCE)BE.1943-5592.0001387

Xu, X., Ren, Y., Huang, Q., Zhao, D.-Y., Tong, Z.-J., & Chang, W.-J. (2020). Thermal response separation for bridge long-term monitoring systems using multi-resolution wavelet-based methodologies. *Journal of Civil Structural Health Monitoring*, *10*, 527–541. doi: 10.1007/s13349-020-00402-7

Xu, X., Xu, Y.-L., Ren, Y., & Huang, Q. (2021). Site-specific extreme load estimation of a long-span cable-stayed bridge. *Journal of Bridge Engineering*, *26*(4), 05021001. doi: 10.1061/(ASCE)BE.1943-5592.0001700

Xu, Y., Chen, B., Ng, C., Wong, K., & Chan, W. (2010). Monitoring temperature effect on a long suspension bridge. *structural control and health monitoring*, *17*(6), 632–653. doi: 10.1002/stc.340

Yang, J., Zhang, L., Chen, C., Li, Y., Li, R., Wang, G., ... Zeng, Z. (2020). A hierarchical deep convolutional neural network and gated recurrent unit framework for structural damage detection. *Information Sciences*, *540*, 117–130. doi: 10.1016/j.ins.2020.05.090

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1090–1109. doi: 10.1111/mice.12412

Yue, Z.-x., Ding, Y.-l., & Zhao, H.-w. (2021, jun). Deep learning-based minute-scale digital prediction model of temperature-induced deflection of a cable-stayed bridge: Case study. *Journal of Bridge Engineering*, *26*(6), 05021004. doi: 10.1061/(asce)be.1943-5592.0001716

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. In *2016 ieee international conference on image processing (icip)* (pp. 3708–3712). doi: 10.1109/ICIP.2016.7533052

Zhao, D., Ren, Y., Huang, Q., & Xu, X. (2019). Analysis of temperature-induced deflection of cable-stayed bridge based on bp neural network. In *Iop conference series: Earth and environmental science* (Vol. 242, p. 062075). doi: 10.1088/1755-1315/242/6/062075

Zhao, H., Ding, Y., Li, A., Sheng, W., & Geng, F. (2020, jul). Digital modeling on the nonlinear mapping between multi-source monitoring data of in-service bridges. *Structural Control and Health Monitoring*, *27*(11). doi: 10.1002/stc.2618

Zhao, J., Ivan, J. N., & DeWolf, J. T. (1998). Structural damage detection using artificial neural networks. *Journal of Infrastructure Systems*, *4*(3), 93–101. doi: 10.1061/(ASCE)1076-0342(1998)4:3(93)

Zhao, R., Yan, R., Wang, J., & Mao, K. (2017). Learning to monitor machine health with convolutional bi-directional lstm networks. *Sensors*, *17*(2), 273. doi: 10.3390/s17020273

Zhao, X., Li, S., Su, H., Zhou, L., & Loh, K. J. (2018). Image-based comprehensive maintenance and inspection method for bridges using deep learning. In *Smart materials, adaptive structures and intelligent systems* (Vol. 51951, p. V002T05A017). doi: 10.1115/SMASIS2018-8268

Zhou, G.-D., Yi, T.-H., Chen, B., & Zhang, H. (2017). A generalized pareto distribution–based extreme value model of thermal gradients in a long-span bridge combining parameter updating. *Advances in Structural Engineering*, *20*(2), 202–213. doi: 10.1177/1369433216660010

Zhu, J., & Song, J. (2020). An intelligent classification model for surface defects on cement concrete bridges. *Applied Sciences*, *10*(3), 972. doi: 10.3390/app10030972