

**Data-Driven and Machine Learning Methods for Improving Infrastructure Performance and Health Assessments under Complex Environmental and Operational Conditions**

by

Omid Bahrami

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Civil Engineering)  
in the University of Michigan  
2021

Doctoral Committee:

Professor Jerome P. Lynch, Co-Chair  
Assistant Research Scientist Wentao Wang, Co-Chair  
Associate Professor Eunshin Byon  
Associate Professor Branko Kerkez  
Associate Professor Jeffrey T. Scruggs

Omid Bahrami

omidb@umich.edu

ORCID iD: 0000-0002-0013-5728

© Omid Bahrami 2021

## **Dedication**

To my father, my mother and my sister.

## **Acknowledgements**

First and foremost, I would like to express my gratitude towards Professor Jerome P. Lynch for his supervision, mentorship and support over the past few years. He played a key role in shaping my character as a researcher and I owe my academic success to him. As one of the best researchers in our field, he selflessly shared his vast technical knowledge and valuable vision with me. In addition to being a fantastic advisor, Professor Lynch has been a great mentor to me during the course of my graduate studies. He patiently guided me and helped me navigate through the challenges that I faced in the past few years. As a student in his research group, I truly felt that my growth was the top priority, which speaks volumes about his outstanding approach to advising and mentoring students. From him, I learned to be patient, open-minded and compassionate when working in professional settings and I will carry what I learned from him with myself to my future career.

Next, I express my sincerest gratitude towards my co-advisor Dr. Wentao Wang for his supervision and mentorship. Dr. Wang played a key role in the growth of my academic character. While our relationship started as student and supervisor, our friendship extended far beyond and I now view him as my older brother. His motivation, passion, academic integrity and hard work has always been an inspiration to me and I have constantly attempted to replicate it in my graduate studies. Beyond an outstanding researcher, he is a great supervisor and mentor who was always there for me and helped me like an angel.



Next, I would like to acknowledge the great role that my committee members played in my growth. The valuable feedback provided by the committee, through both the group meetings and individual meetings that I had with them, was essential in increasing the academic rigor of my dissertation and contributed significantly to the quality of my research. Beyond the invaluable feedback and recommendations, each member uniquely contributed to the growth of my character as a scholar. I met very frequently with Dr. Byon and the meetings with her expanded my vision significantly. Dr. Kerkez is perhaps the nicest instructor I have had and his great course on Infrastructure Sensing was extremely useful in my research. Finally, I served as a Graduate Student Instructor for CEE 375 that Dr. Scruggs taught in Winter 2021, where I had the privilege of observing his level of hard work and dedication closely. Dr. Scruggs' work ethic has been truly inspiring and I have tried to replicate that ethic in the course of my studies and I will do so in the next stages of my career. In addition to my committee members, I would like to thank Professor Peter Adriaens for allowing me to take his valuable course on Infrastructure Finance, which played a key role in helping me determine my future career path and also for his support.

Next, I express my appreciations towards my friends and colleagues for their friendship and support. In the past six years, I managed to cultivate a rich network of extremely talented and kind individuals, which I will carry with myself beyond Ann Arbor. Specifically, I would like to thank my dear friends Dr. Azad Ghaffari and his kind wife Farinaz Nezampasandarbabi and Dr. Nephi Johnson and his kind wife Ashley Johnson. I am extremely grateful for my wonderful lab mates Dr. Sean O' Connor, Dr. Yilan Zhang, Mr. Mitsuhiro Hirose, Dr. Andrew Burton, Dr. Peng Sun, Dr. Rui Hou, Dr. Hao Zhou, Dr. Katherine Flanigan, Mr. Kidus Admassu and Mr. Gabriel Draughon. Additionally, I am extremely grateful for having the opportunity to meet and collaborate with my colleague Dr. Stavroula Tsiapoki.

I am thankful to the wonderful staff at the department of Civil and Environmental Engineering and the staff at the International Student Center who worked extremely hard to ensure that I can devote my entire attention to my research and took care of all administrative issues. I am also thankful to the crew of the Michigan Department of Transportation who helped me a lot during our field tests.

I also acknowledge the financial support provided by the National Science Foundation, Graham Institute of Sustainability, Great Lakes Water Authority and Rackham Graduate School.

Above and beyond all, I will forever be thankful to my father, my mother and my sister for all their help and support, both emotionally and financially, for putting my wishes and aspirations ahead of theirs and sacrificing significantly to enable me to pursue my dreams.

## Table of Contents

<b>Dedication .....</b>	<b>ii</b>
<b>Acknowledgements .....</b>	<b>iii</b>
<b>List of Tables .....</b>	<b>x</b>
<b>List of Figures.....</b>	<b>xii</b>
<b>Abstract.....</b>	<b>xvi</b>
<b>Chapter 1. Introduction .....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 “Intelligent” Infrastructure Systems .....	5
1.2.1 Sensors in Infrastructure Systems.....	5
1.2.2 Advances in Sensing Technology.....	6
1.2.3 SHM.....	8
1.2.4 Cyber-Physical Systems.....	11
1.3 Knowledge Gaps and Technical Challenge of Intelligent Infrastructure Systems .....	13
1.3.1 The Role of EOCs on Data-Driven Decision Processes for SHM.....	13
1.3.2 Assumptions of Independence and Identical Distribution .....	18
1.3.3 CPS-Enabled Infrastructure Monitoring and Assessment .....	19
1.4 Research Goal and Objectives .....	23

1.5	Dissertation Outline .....	26
<b>Chapter 2. Two Methods for Better EOCs Data Normalization for SHM: Use of EOCs-Sensitive Features and Soft Assignment .....</b>		
<b>30</b>		
2.1	Introduction.....	30
2.2	Theory and Methodology.....	32
2.2.1	Damage Detection Framework .....	33
2.2.2	Soft Assignment for Data Normalization .....	40
2.3	Description of the Dataset.....	43
2.3.1	Experimental Dataset .....	43
2.3.2	Simulation Dataset .....	45
2.4	Results and Discussions .....	48
2.4.1	EOC-Sensitive Features .....	49
2.4.2	SHM Framework Application.....	53
2.5	Conclusion .....	70
<b>Chapter 3. Hidden Markov Models for Sequential Damage Detection of Bridges.....</b>		
<b>71</b>		
3.1	Introduction.....	71
3.2	Theory .....	73
3.2.1	Hidden Markov Models .....	73
3.2.2	Damage Detection Using HMM.....	75
3.3	Description of the Dataset.....	76
3.4	Analysis.....	78

3.4.1	Three Tier Damage Detection Framework .....	78
3.4.2	Results and Discussions .....	82
3.5	Conclusion .....	85
<b>Chapter 4. Deep Learning Based Bridge Weigh-in-Motion Using Encoders.....</b>		<b>87</b>
4.1	Introduction.....	87
4.2	Theoretical Background.....	88
4.2.1	RNN-Based Encoders .....	89
4.2.2	Hidden State Aggregation.....	91
4.3	Description of the Datasets .....	92
4.3.1	I-275 Monitoring System .....	93
4.3.2	Data Acquisition Process .....	96
4.3.3	Description of the Finite Element Model.....	99
4.4	Results and Discussions .....	102
4.5	Conclusions.....	105
<b>Chapter 5. A Sequence-to-Sequence Model for Joint Bridge Response Forecasting .....</b>		<b>107</b>
5.1	Introduction.....	107
5.2	Description of the Dataset.....	110
5.2.1	Baseline Removal .....	112
5.2.2	Measurement Noise Removal .....	113
5.2.3	Signal Decomposition.....	114
5.3	Theory .....	115

5.3.1	Seq2Seq Model .....	116
5.3.2	Training the Seq2Seq Model .....	119
5.3.3	ARX Model.....	121
5.3.4	Regression-based Baseline.....	123
5.3.5	A Note on the Baseline Models .....	125
5.3.6	Error Metrics .....	126
5.4	Results and Discussions .....	128
5.5	Conclusion .....	141
<b>Chapter 6. Conclusion and Future Work.....</b>		<b>142</b>
6.1	Summary .....	142
6.1.1	Key Intellectual Contributions .....	144
6.2	Future Work .....	150
6.2.1	Research Questions.....	150
6.2.2	Broader Impact.....	151
<b>Bibliography .....</b>		<b>153</b>

## List of Tables

Table 2-1. AUC values for DSF-1 for different clustering approaches and different number of clusters for LANL dataset. ....	54
Table 2-2. AUC values for DSF 2 for different clustering approaches and different number of clusters for LANL dataset. ....	55
Table 2-3. AUC values for DSF-1 for different clustering approaches and different number of clusters for simulation dataset with weak damage. ....	60
Table 2-4. AUC values for DSF-1 for different clustering approaches and different number of clusters for simulation dataset with moderate damage. ....	61
Table 2-5. AUC values for DSF-1 for different clustering approaches and different number of clusters for simulation dataset with strong damage. ....	62
Table 2-6. AUC values for DSF-2 for different clustering approaches and different number of clusters for simulation dataset with weak damage. ....	63
Table 2-7. AUC values for DSF-2 for different clustering approaches and different number of clusters for simulation dataset with moderate damage. ....	64
Table 2-8. Table 8. AUC values for DSF-2 for different clustering approaches and different number of clusters for simulation dataset with strong damage. ....	65
Table 3-1. AUC values for HMM and baseline methods. ....	83
Table 4-1. Test results for two types of hidden state aggregation. ....	104

Table 4-2. Testing results for the proposed Encoder model with training labels being corrupted with Gaussian noise. ....	105
Table 5-1. Frequency of re-identified vehicles for different truck lanes combinations on the I-275 highway.....	111
Table 5-2. Static response forecasting RMSEs for forecasting TRB using NRB.....	129
Table 5-3. Dynamic response forecasting RMSEs for forecasting TRB using NRB. ....	129
Table 5-4. Combined TRB forecasting RMSEs when static and dynamic components are forecasted separately. ....	130
Table 5-5. Combined TRB forecasting RMSE when the actual signal is used without separating the static and dynamic components. ....	130
Table 5-6. Model performance on the test dataset when forecasting TRB's response. ....	131
Table 5-7. Description of properties for sample trucks used. ....	132
Table 5-8. Combined forecasting results for using the response of TRB to forecast the response of NRB. ....	133
Table 5-9. Model performance on the test dataset when forecasting NRB's response. ....	133
Table 5-10. Average nRMSE values for class 9 trucks for models trained with the entire data and data from class 9 only. ....	140



## List of Figures

Figure 1-1. Description of the scope of the dissertation. ....	4
Figure 1-2. Thesis Outline. ....	28
Figure 2-1. The Whisper 500 wind turbine with the locations of sensor nodes and the damage scenario. ....	44
Figure 2-2. Description of the methodology of generating simulation datasets. ....	44
Figure 2-3. (a) Sample acceleration spectrogram. (b) Rotor angular velocity for the same dataset. ....	50
Figure 2-4. Average peak frequencies and rotor angular velocity. ....	51
Figure 2-5. Average signal energy ratio an average yaw angle. ....	51
Figure 2-6. Damage EOCSFs against the corresponding EOCs. (a) Average peak frequencies and (b) average signal energy ratio. ....	52
Figure 2-7. Statistics of AUC values for a) DSF-1 and b) DSF-2 for different realizations of the experimental data. ....	56
Figure 2-8. ROC curves for experimental dataset for a) DSF-1 and b) DSF-2 for 5, 6 and 7 EOC clusters. ....	57
Figure 2-9. ROC curves for simulation dataset for DSF-1 for a) 8 and b) 13 EOC clusters for weak, moderate and strong levels of structural damage. ....	66
Figure 2-10. ROC curves for simulation dataset for DSF-1 for a) 16 and b) 21 EOC clusters for weak, moderate and strong levels of structural damage. ....	67

Figure 2-11. ROC curves for simulation dataset for DSF-2 for a) 8 and b) 13 EOC clusters for weak, moderate and strong levels of structural damage. ....	68
Figure 2-12. ROC curves for simulation dataset for DSF-2 for a)16 and b) 21 EOC clusters for weak, moderate and strong levels of structural damage. ....	69
Figure 3-1. Graphical representation of the HMM. The observed variable (gray circles) are emitted from states that are not observable (hidden). The hidden states possess Markov property.....	74
Figure 3-2. The side view of the Z-24 bridge. ....	77
Figure 3-3. Web temperature values for the Z-24 dataset. ....	79
Figure 3-4. ROC curves showing the performance of the three AD methods for four different number of hidden states/clusters. ....	82
Figure 3-5. HMM AUC values for different sequence lengths. ....	84
Figure 4-1. Schematics of the LSTM cell. ....	90
Figure 4-2. Schematic of the Encoder utilizing bi-directional LSTM cells and an aggregation module. The red cells represent the forward LSTMs and the blue cells represent the backward LSTMs. ....	91
Figure 4-3. Schematics of the aggregation module for a) fully connected ANN b) Max Pooling layer and fully connected aggregation modules.....	92
Figure 4-4. Components of the monitoring system in a) base station b) camera monitoring system. ....	95
Figure 4-5. Images of a sample truck identified and re-identified across the I-275 monitoring testbed. ....	99
Figure 4-6. a) Schematic drawing for the NRB model, b) sensor locations for the bridge weigh-in-motion. ....	100

Figure 5-1. Signal pre-processing steps for joint bridge response forecasting. .... 111

Figure 5-2. Strain measurement locations on a) TRB and b) NRB. .... 113

Figure 5-3. Sample signal decomposition. a) represents the observed TRB response (after noise removal) along with the static response signal b) the dynamic response signal..... 116

Figure 5-4. Seq2Seq model architecture. The Encoder condenses the information from the input signal into the context vector and the Decoder uses this vector to forecast the output response. .... 118

Figure 5-5. Schematic representation of the GRU cell. .... 118

Figure 5-6. Schematic representation of the regression-based baseline model. The information content of each signal is captured in regression coefficients and another regression model is used to related these coefficient vectors..... 124

Figure 5-7. TRB forecasting training and validation errors (MSE) for the Seq2Seq model for GRU (left) and LSTM (right) cell types..... 130

Figure 5-8. Sample trucks used to evaluate the performance of the Seq2Seq model. Figures a) to d) represent samples 1 to 4 respectively. .... 132

Figure 5-9. TRB forecasting results for 4 sample observations..... 135

Figure 5-10. NRB forecasting training and validation errors (MSE) for the Seq2Seq model for GRU (left) and LSTM (right) cell types. .... 136

Figure 5-11. NRB forecasting results for 4 sample observations. .... 137

Figure 5-12. Truck frequency in the dataset for a) different axle numbers and b) different FHWA vehicle classes. .... 138

Figure 5-13. Description of highway vehicle classification by FHWA (image source: FHWA). .... 139

Figure 5-14. Seq2Seq model performance for different truck classes..... 140

## **Abstract**

Infrastructure systems (IS) play a vital role in supporting the well-being of our society. A grand challenge confronting IS managers is existing asset management methods are falling short in ensuring safe and reliable IS components. Recent technological advances in the field of sensing and information technology have created opportunities to explore new approaches to managing IS components based on the use of data as quantitative evidence of structural performance and health. In tandem, advances in data science and machine learning (ML) have resulted in new data-driven analytical tools for efficiently processing large volumes of data. This dissertation explores the creation of data-driven analytical frameworks that extract information from the data generated by structural monitoring systems to help make better asset management decisions centered on structural performance and health. A challenge of assessing performance and health of IS components is the large variability such systems have in their environmental and operational conditions (EOCs). Hence, the overarching goal of the dissertation is to develop data-driven analytical frameworks that identify EOC for data normalization that improve structural performance assessments. First, the thesis explores new approaches to handling EOCs during the data normalization stage of structural health monitoring (SHM) algorithmic frameworks. The thesis proposes the extraction of EOC Sensitive Features (EOCSFs) from structural response data with EOCSFs used for data normalization. Unsupervised clustering of EOCSFs are used to establish EOC clusters during training to ensure damage sensitive features (DSFs) extracted from response data of the structure in an unknown state are fairly compared to DSFs of the healthy

structure operating in the same EOC state. To normalize test data, a novel soft assignment approach is also proposed to account for the uncertainties associated with assigning an EOCSF to a given EOC cluster. These innovations are shown to outperform traditional hard assignment using EOCs inferred from measurements taken independently of the structure. Wind turbines that experience wide EOC variability are used as an illustrative example. Second, the dissertation challenges the assumption of independence and identical distribution traditionally applied to condition assessments of a structure over prolonged observation periods. A novel approach to training Hidden Markov Models (HMM) to track structural deterioration with state dependencies under varying EOCs is proposed. The Z24 Bridge SHM testbed is adopted to validate the efficacy of the method to assessing the condition of a structure based on past observations of structural condition. Finally, in the last part of this dissertation, two ML-based frameworks are applied for EOCs identification and response assessment of highway bridges. First, an encoder is trained to extract truck weight characteristics from bridge response to truck traffic using training data collected from a cyber-physical system (CPS) architecture that links bridge responses with measured vehicular weights. Next, a sequence-to-sequence (Seq2Seq) model is used to forecast the response of one bridge to a truck given the response of another bridge to the same load. The Seq2Seq model enables the estimation of bridge responses in a highway network by using the response of limited number of instrumented bridges. It is shown that by normalizing the input observations based on vehicle load type, the predictive performance of the Seq2Seq model is increased. In summary, the thesis breaks new ground in advancing data-driven frameworks that can automate the conversion of IS monitoring data into valuable information for a plethora of IS applications.

# **Chapter 1.**

## **Introduction**

### **1.1 Introduction**

Infrastructure systems (IS) play a vital role in the well-being of any society and serve as the backbone of a healthy economy (Tomer et al., 2021). IS constitute a broad range of inter-connected subsystems including transportation systems, energy utilities, water and wastewater systems, and even by most recent definitions, telecommunication (e.g. broadband) systems. To ensure that IS can support the development and well-being of society, approaches to maintaining the physical well-being of IS are of the needed including vigilant inspection by visual means. Although transformative changes are occurring in terms of new technologies and methods useful for efficient management of IS, management methods in practice today have been slow to evolve at the same pace (Tang et al., 2018). The challenge that confronts IS stakeholders (e.g. engineers, stakeholders, managers, policy-makers) is that the infrastructure management strategies of the twentieth century are failing to ensure adequate maintenance of IS jeopardizing their ability to meet the demands society imposes on them. For instance, the American Society of Civil Engineers (ASCE) in 2021 has given a grade of “C-” to the United States infrastructure in its latest Report Card for America’s Infrastructure (ASCE, 2021). According to this report for example, more than 46,000 of America’s bridges are considered structurally deficient and an estimated \$125 billion dollars is needed for bridge repairs. Other alerting statistics is that water mains break once every two minutes, while

56% of the nation's ports are rated as in poor to mediocre conditions; similar statistics are faced in almost all other categories evaluated by ASCE. According to McKinsey & Company, there is a backlog of more than \$2 trillion dollars on public-infrastructure investing needed to improve the condition of the nation's IS (McKinsey Global Institute, 2020). As such, tremendous attention has been devoted by the civil engineering community to re-envisioning the management methods used for IS over the past few years.

Visual inspection methods to IS management have been used, both informally and formally, over hundreds of years. Visual inspection is defined as a method of human inspectors visually looking at the surfaces of a structure looking for obvious signs of deterioration (e.g., rust, cracks, spalling). Inspector judgement is then used to interpret the impact such deterioration might have on the performance and safety of the structure. The bridge reengineering community has one of the more advanced visual inspection frameworks (Agdas et al., 2016). However, recent studies focused on assessing the accuracy of visual inspection methods have focused wide variability in rating scores given by well-trained inspectors (e.g. 95% of rating scores vary by 2 points out of 9) when evaluating the same structure (Agrawal et al., 2021; FHWA, 2001). While visual inspection has a good track record, these findings underscore the subjectivity introduced by the inspector. The rapid progression of sensing technology in the mid- to late-twentieth century created new approaches to quantitatively measuring structural condition by non-destructive testing (NDT) methods including acoustic emissions, ultrasonic, Eddy currents, and radiography, just to name a few (Maierhofer et al., 2010). Such methods are excellent at providing a detailed and accurate assessment of structural condition but require expensive equipment and trained technicians to operate. They are also local inspection methods making the decision of where to apply often challenging with high uncertainty. Alternatively, structural health monitoring (SHM) emerged in



the 1990's and continues to be an area of active research. SHM defers from NDT in that it aims to install permanent sensors in a structure; it also aims to automate the processing of data to ensure limited human subjectivity can be introduced in its processes. SHM is already finding practical use in a variety of structures including bridges (Seo et al., 2016) and wind turbines (Ciang et al., 2008).

Second, innovative approaches are being explored to provide additional resources for the construction and maintenance of IS. While technologies like SHM provide an approach to assess structural conditions, financing innovations is still needed to close financing gaps in IS as pointed out by ASCE (ASCE, 2021). Public-Private Partnerships (PPPs) are one such example wherein public entities, as the owners and custodians of the IS, join private entities to collaborate on IS projects. The private sector agent often finances and/or plays the role of IS maintainer. Despite doubts, PPPs are becoming a prevalent practice in North America and are expected to result in increased efficiency and value-for-money, potentials for sharing project risks between public and private entities, as well as reduced delays and cost overruns (Verweij et al., 2017). Moreover, research efforts are being extended to explore even more novel and innovative approaches to financing real-world IS projects. For instance, Ashuri et. al. (2012) proposed an innovative approach of using options theory to price a common form of implementing PPPs for highway projects which are referred to as build-operate-transfer projects (Ashuri et al., 2012). In addition to such valuable application-oriented efforts, fundamental research has also been conducted to adjust the existing pricing models for IS projects. By considering market and non-market risks along with the fact that investors are loss-averse, Espinoza et. al. (2020) developed a more appropriate framework for assessing value of IS projects (Espinoza et al., 2020). Many of the emerging approaches to financing infrastructure projects are embracing data in their design. One

stark example is using structural response data to assess user fees based on the extent of consumed life a user has imposed on the structure. Such methods are gaining traction in highway bridges and roads as a new approach to tolling traffic for their use of highway IS asset (Gungor et al., 2018, 2019).

Condition assessment and financing of IS using data are just one example of what is possible for improving the management of IS using data. While IS data can be used for a wide variety of other purposes (e.g. control of IS performance), the focus of this dissertation will be on monitoring and assessing the performance of structural components of IS. By taking advantage of recent advances in infrastructure monitoring technologies, along with recent advances in artificial intelligence (AI) and Machine Learning (ML), this dissertation aims to contribute data-driven solutions to addressing the most pressing challenges for structural assessment and management. Figure 1–1 indicates the scope of the thesis. In the remainder of this chapter, the advances in the

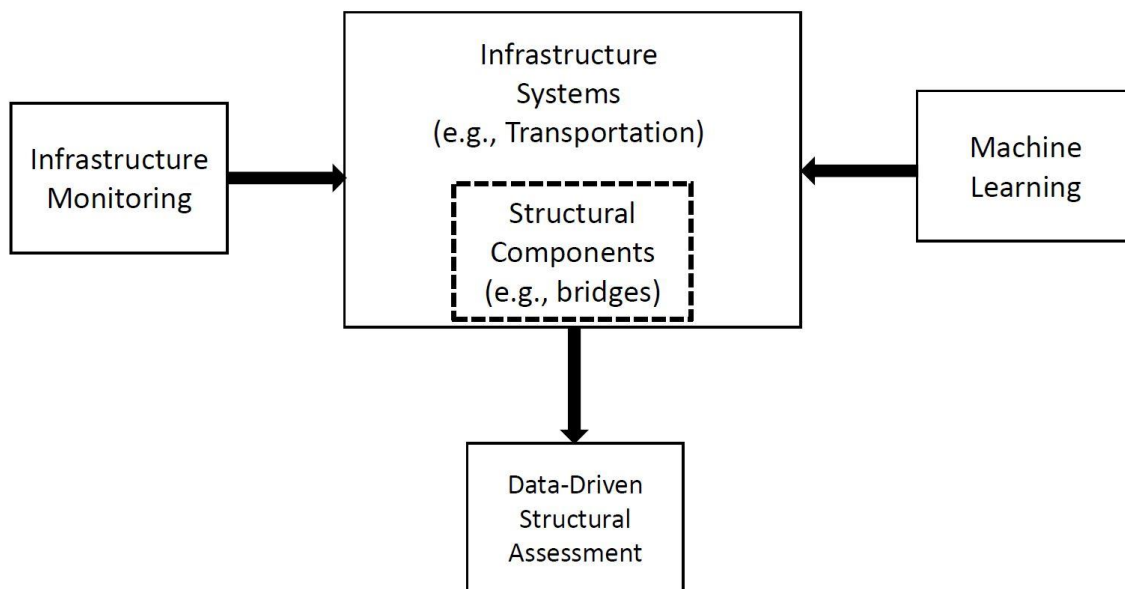


Figure 1-1. Description of the scope of the dissertation.

field of SHM are first described in greater detail. Second, the goal and objectives of the dissertation are described in detail, with the chapter concluding with an outline of the dissertation.

## **1.2 “Intelligent” Infrastructure Systems**

### **1.2.1 Sensors in Infrastructure Systems**

The emergence of “intelligent” IS dates back to 1940s when early accelerometers were first used to monitor the vibration of critical structures like long span bridges (e.g. Golden Gate Bridge) (Vincent, 1958). The intent of instrumenting IS components with monitoring systems was to gain better understanding of the behavior of structures exposed to extreme loads such as earthquakes and strong winds. Instrumentation of long span bridges occurred following the collapse of the Tacoma Narrows Bridge in 1940 where the problem of wind-induced vibration of suspension bridges became self-evident (Abdel-Ghaffar & Scalan, 1985). The seismic monitoring of structures in particular was a strong driver of the adoption of sensors for monitoring structural assets (Wang et al., 2014). Such monitoring systems are based on trigger-based data acquisition systems that record when a large seismic event is detected. Today through California’s Strong Motion Instrumentation Program, 650 ground-response stations have been installed in California with over 170 buildings, 60 bridges and 20 dams being instrumented since 1972 (CSMIP, 2021). The monitoring of IS under extreme loading conditions is not limited to seismic events. Numerous buildings and bridges have been also instrumented for monitoring the structural response to extreme wind loads. For example, the Akashi Kaikyo Bridge in Japan was instrumented during its construction in the early 1990’s to monitor the bridge response to wind loads given the massive size of the bridge (i.e., it was the largest suspension bridge in the world at the time of its construction) (Kashima et al., 2001). Another example is the Wind and Structural Health

Monitoring Systems (WASHMS) installed in Hong Kong suspension bridges beginning with the Tsing Ma Bridge in 1999 and later the Kap Shui Mun Bridge and Ting Kay Bridge (Ni & Wong, 2012; K.-Y. Wong & Ni, 2009; K. Y. Wong, 2004). These bridge monitoring systems have been vital to deciding when to close the bridges due to extreme wind responses during typhoons. Similarly, offshore structures have been instrumented to assess their response to wave load (Fugro, 2021). The primary motivator for most of these monitoring systems was to collect response data from large-scale and critical structures under extreme loads. The cost and complexity of installing these monitoring systems (Lynch & Loh, 2006) were justified by the importance of the structure and the value of the data to be collected. However, by the late 1990's, sensing technologies would begin to go through a transformation making high performance sensors cheaper and easier to install. These changes would ultimately make sensors available to a wide range of structures (i.e., not just large or critical structures) and for a wide range of purposes (i.e., SHM).

### **1.2.2 Advances in Sensing Technology**

A key factor preventing earlier adoption of permanent monitoring systems for civil IS has been their high costs (Wang et al., 2014). Costs on the order of thousands of dollars per channel placed many opportunities to deploy monitoring systems out of reach. These costs had been driven by expensive sensors (e.g. force balance accelerometers), wired communication systems, and multi-channel data acquisition systems. By the late 1990's, the landscape was changing quickly and dramatically with sensing technologies rapidly evolving into lower cost yet high performance solutions. These trends have continued to the modern day there by opening structural monitoring to a wide array of IS applications including SHM and “smart”, cyber-enabled infrastructure (e.g., connected vehicles interacting with road-side sensors).

Perhaps the most significant of the advances in sensing to occur in the 1990's was the ability to fabricate sensors using the same methods as those used to fabricate integrated electronics. Silicon was recognized as a potential material enabling the development of cost-effective, batch-fabricated and high-performance sensors and actuators (Petersen, 1982). The development of microelectromechanical systems (MEMS) was key to sensor miniaturization while driving sensor costs per unit to a low level (Liu, 2012). For instance, studies in the literature have shown that MEMS-based accelerometers are an order of magnitude cheaper when compared to traditional piezoelectric and force-balanced accelerometers (Bedon et al., 2018; Evans et al., 2014). MEMS sensors are also smaller than non-MEMS counterparts. Most importantly, their performance has compared well to traditional sensors. As a result, MEMS sensors are found in a wide variety of applications including sensing in cars (e.g., accelerometers for airbag deployments), cell phones (e.g., accelerometers to assess the phone orientation), etc. Similarly, MEMS sensors are increasingly common in IS monitoring applications.

The development of digital data acquisition (DAQ) systems was another key factor enabling the adoption of sensors. Compared to the Golden Gate Bridge monitoring system of the 1940's and 1950's where Hall accelerometers utilized pen and a rotating drum to record the sensor readings (Wang et al., 2014), the introduction of digital electronics and computers provided an essential tool for efficient data collection and storage. By designing DAQs with digital electronics and microprocessors, the DAQ industry could ride the trends of the broader electronics industry including use of faster processors (as defined by Moore's Law) with reducing cost. Today, standard DAQ solutions are very accessible to a broad range of users including IS stakeholders. While DAQ technology has improved, the vast majority of DAQs use wires to communicate sensor

data to the external DAQ platform. It has been these wires that have kept monitoring solutions a high cost alternative to traditional visual inspection of IS.

To address the limitations imposed on IS monitoring by wired DAQ architectures, the use of wireless telemetry was introduced to SHM systems in the late 1990s by Straser et. al. (1998) (Straser et al., 1998). Wireless sensor networks (WSNs) were proposed to reduce the instrumentation cost through eliminating the need for wiring and making the monitoring system deployment process less labor intensive (and thereby lower cost). WSNs enabled the deployment of dense arrays of network and showed potentials for distributed actuation and control. By the early 2000's there were numerous WSNs deployed in IS for SHM applications (Lynch & Loh, 2006; Lynch, 2007). A wide variety of innovative wireless sensors have been proposed for SHM and validation of these wireless sensors have been done at full-scale using traditional IS including bridges, buildings, wind turbines, and ships (Häckell et al., 2016; Jang et al., 2010; Johnson et al., 2018; Kurata et al., 2005; Pakzad, 2010; Pakzad et al., 2008). With low system costs and easy-to-deploy, modular installations, the ability to instrument infrastructure and to acquire a large volume of data. Today, the challenge of extracting value from IS monitoring data remains. Especially, how to use IS response data to infer the condition and performance of infrastructure remains an open research question in the SHM field.

### **1.2.3 SHM**

Preserving integrity of physical components of IS is of utmost importance when it comes to their safe and cost-efficient management. In recent years, there has been increasing demand on preserving IS assets and gaining the most usage out of them rather than building new infrastructure components due to growing finance gaps in infrastructure renewal in most mature economies like

the United States (Soga & Schooling, 2016). The need to “do more with less” is demanding new approaches to how civil engineers manage IS assets. One approach is structural health monitoring where sensor data is used to assess IS asset performance and structural conditions.

The rapid advancement and adoption of sensing technology in IS applications has led to the creation of the field of structural health monitoring (SHM). Formally, SHM is defined as the practice of strategies implemented for damage identification within structural systems (Worden et al., 2007). In this context, damage is defined as changes within the structural system (i.e. materials, geometry, and/or boundary conditions), that adversely affects both the local and global performance of the system (Worden et al., 2007). SHM can be performed on five different hierarchical levels: 1) damage detection, 2) localization, 3) classification, 4) assessment and 5) prediction (Farrar & Worden, 2012; Rytter, 1993). As is self-evident, each of these levels requires the output of all the previous ones. For instance, to locate damage in the structure, first the damage needs to be detected. Consequently, SHM algorithms remain underdeveloped primarily for the first few levels including detection and localization of its occurrence. Guiding the development of SHM algorithms, irrespective of the level considered are a set of seven fundamental axioms as first formalized by Worden et. al (2007). Perhaps the most significant of them all is Axiom II. Axiom II of SHM states that for damage assessment, a comparison between two system states is needed (Worden et al., 2007). Consequently, the comparison performed by SHM can be done either by using model-based approaches which require high fidelity physical models of the structure, or data-driven approaches which rely on statistical models built from measured structure response data (Farrar & Worden, 2012; Tibaduiza et al., 2013).

The earliest SHM algorithms grew out of the modal analysis field and largely relied on modal properties of structures within a model-based approach. Modal methods that compared

modal characteristics (e.g., frequency, damping, mode shape) were convenient due to the readily accessible accelerometers that could easily record structural vibration. An excellent literature review of vibration-based SHM was offered by Sohn et. al. (Sohn et al., 2003) but some notable examples are provided here. Williams and Messina (1999) developed an SHM framework by analyzing a correlation coefficient that compared changes in a structure's modal frequencies with predictions obtained from a frequency-sensitive finite element model (Williams & Messina, 1999). Chaudhari and Maiti (1999) used the Frobenius technique to develop a mechanical model for computing the transverse vibration of a geometrically segmented beam with and without cracks along the beam's normal axis (Chaudhari & Maiti, 1999). The cracked section was modeled as a rotational spring and the model showed high accuracy for both locating the crack and estimating the size of the crack. Leutenegger et al. (1999) attempted to locate and assess the length of fatigue cracks in transversely vibrating beams using the relationships between crack length and resonant frequencies evaluated by a mechanical model of the cracked beam (Leutenegger et al., 1999).

Model-based approaches are dependent on a structure's geometry and material composition, and possess the additional challenge of modeling uncertainties within their operational environment (Ying et al., 2013). An especially challenging aspect of model-based SHM methods is that their properties are often sensitive to both damage and the environment, making such methods difficult to apply in real-worlds settings. Consequently, data-driven SHM was proposed in response to the challenges and today is well established as the main approach to SHM. Earlier implementations of data-driven SHM consisted of two key steps following data acquisition. First, damage sensitive features (DSFs) were extracted from structural response data and then statistical models were built to discriminate between the response from undamaged and damaged structures. Sohn et al. (2003) have gathered a valuable and insightful summary of the



SHM literature by the early 2000's when the pivot to data-driven SHM was occurring (Sohn et al., 2003). Early implementations of data-driven SHM relied on statistical pattern recognition techniques for making a decision on the state of the structure (Sohn et al., 2001). Statistical process control charts (Sohn et al., 2000) and the well-known F-test (Sohn & Farrar, 2001) are amongst other decision rules used in early data-driven SHM applications. The portfolio of data-driven methods has only expanded over the past two decades with a wide range of algorithmic approaches emerging including the use of machine learning (ML). While these methods are improving the capabilities of SHM methods, such methods can still be challenges by changes in the operational environment that affect the features used in structural health assessments.

The SHM community was quick to realize the challenging nature of SHM in that the extent of complicated interactions between different structural components and the structure and the environment render well-known elementary pattern recognition techniques ineffective for SHM. As a result, SHM practitioners resorted to more complicated Signal Processing (SP) and Machine Learning (ML) techniques for data-driven SHM. Needless to say that the rapid progress, which is still on-going, in the fields of signal processing and artificial intelligence (AI) has been a major contributor to development of sophisticated ML-based SHM frameworks. In the next subsection, a more detailed review of the more recent ML-based SHM frameworks is presented. However, prior to doing so, another important concept that has had revolutionary impact on IS monitoring and management, will be introduced in the next subsection.

#### **1.2.4 Cyber-Physical Systems**

The current phase of “intelligent” IS is the concept of Cyber-Physical System (CPS) were first formalized by the National Science Foundation (NSF) in 2006 (Lee & Seshia, 2016). Later in

2007, it was recognized as a paramount and promising research direction in the field of networking and information technologies by President's Council of Advisors on Science and Technology (PCAST) (Gunes et al., 2014; Krogh et al., 2008). Rajkumar et al. (2010) formally define CPS as “physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core” (Rajkumar et al., 2010). Although this is one definition for CPS, it must be mentioned that other definitions of CPS exist within the scientific community (Gunes et al., 2014). Clearly, monitoring and control systems have been in existence for decades but these systems would not be considered as CPS. To be a CPS solution, accessible computing that can be scaled up to a large volume of data is an essential design feature. The current base of computing resources now available upon demand in “clouds” is driving CPS solutions. Specifically, through utilizing cloud computing, CPS solutions in IS applications can be created to perform data-driven monitoring and control services in real-time. Consequently, IS managers, owners and users can benefit significantly by embracing CPS architectures. The integration of communication and cloud computing have already had transformative benefits for IS stakeholders.

Perhaps the most prominent example of using CPS architecture for enhancing IS services are connected and autonomous vehicles (CAVs). CAVs possess large computational and storage power and utilize vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications to enhance their operations (Rawat et al., 2015). Sensors in the CAV and roadside transmit their data to each other and the roadside infrastructure. As higher levels of autonomy are sought, the use of cloud computing will increase to analyze large volumes of CAV data. Placing these computational services into the control loops that control CAVs in effect close the loop within the CPS architecture. In addition to CAVs, the transportation infrastructure (e.g. bridges) have been transformed by embracing CPS solutions. Intelligent transportation systems augment CPS to

existing transportation infrastructure to achieve real-time (or near real-time) traffic monitoring and control (Moller & Vakilzadian, 2016; Zeng et al., 2019). Other CPS examples beyond transportation infrastructure include smart energy grids (Khaitan et al., 2015; Seshia et al., 2017), smart water networks (Eggimann et al., 2017; Kerkez et al., 2016) and smart heating, ventilation and air-conditioning systems (Schmidt & Åhlund, 2018).

### **1.3 Knowledge Gaps and Technical Challenge of Intelligent Infrastructure Systems**

As data-driven structural assessment methods grew in popularity there are a number of notable challenges that emerge. This section will highlight a number of notable challenge that exist in intelligent infrastructure that affect their implementations in real-world IS. First, the role of environmental and operational conditions (EOCs) in the SHM paradigm is presented; EOCs often manifest is structural monitoring data in a manner that can mask change in the data associated with structural deterioration. Second, the underlying assumption of observation independence of many data-driven assessment methods is explored. Specially, such assumptions may limit the applicability of such approaches when deterioration and damage are a one way process (i.e., structures generally do not “heal” or get better).

#### **1.3.1 The Role of EOCs on Data-Driven Decision Processes for SHM**

Civil IS are typically massive and complex engineered systems that are exposed to very large loads including gravity and lateral loads. They also are exposed to their natural, ambient environments which can change the temperature of the structure.

An important challenge in the application of data-driven approaches to SHM for structures operating in the real-world conditions is the fact that benign variations in environmental and

operational conditions (EOCs) under which the structure is operating can influence structure behavior (Cross et al., 2012). EOC variation often result in changes in a structural system behavior that would be considered “normal” (i.e., still part of the range of response of a healthy structure). Given the aforementioned Axiom II of SHM (Worden et al., 2007), this presents challenges when comparing a structure at two separate instances in time. A change in feature observed between these two points of observation could be due to damage, EOC variations alone (i.e., structure is undamaged), or both. For example, the variations in EOCs could alter the response of a healthy structure in a manner that response features exceed levels which engineers consider normal conditions (under static EOC scenario) and hence lead to false positive alarms. Alternatively, variations in EOCs could mask the effects of structure damage and render the data-driven approach less sensitive (or in the worst case, insensitive) to structural damage states which results in false negative situations (Cross et al., 2012). This calls for removing the influences of EOCs from structural response data by a process known as EOCs data normalization (Sohn, 2007). An important challenge that remains as SHM transitions from research to practice is the development of robust EOCs data normalization methods (Figueiredo et al., 2014; Figueiredo et al., 2011). As a result, this issue has received significant attention within the SHM community over the past two decades.

Prior to discussing the different methods for EOCs data normalization, it is worth taking a deeper look at the philosophical significance of EOCs data normalization. A key point that must be stated is that at the very core, EOCs affect the structural system by either affecting the load, or more commonly how the system responds to a given load. The three pillars of solid mechanics are the equations that define systems equilibrium, compatibility of displacements (and strains), and the constitutive relations of the materials used in the structure that map stress to strain (Vekstein,

2020). Variations in EOCs manifest both in the equilibrium equations and constitutive relationships. For example, for a wind turbine tower, wind conditions (an EOC that is often characterized by wind speed and direction) affect the drag force applied to the tower. Additionally, by exciting the blades and power transmission systems within the nacelle, wind conditions also affect the load that is applied to the tower by the nacelle. These EOC-structure relationships would naturally manifest in the equilibrium equations used to model the turbine system. Another vivid example for how EOCs could affect the constitutive relations is through affecting the modulus of elasticity. For concrete, which is a frequently used material within civil engineering structures, the modulus of elasticity is dependent on temperature (Fanella, 2016). Moreover, temperature variations can affect the global response characteristics of a structure; for example, bridge modal frequencies are very well documented to be temperature dependent (Zhou & Yi, 2014). Finally, variations in temperature, can indirectly result in a mechanical loading on many statically indeterminate structures.

Consequently, by applying EOCs-based data normalization techniques for data-driven structural assessment methods like SHM, the practitioner is in essence attempting to account for the fact that the system is uniquely responding under different loading conditions. By normalizing the data with respect to EOCs, the goal is to compare newly observed responses to a specific baseline response in the same EOC state, such that any variations between the two responses can be attributed to structural damage (and not EOCs). It must be re-iterated that what distinguishes the varied response caused by change in EOCs from those of damage, is the fact that variations caused by EOCs are reversible.

One way to classify EOC data normalization methods is based on the extent that these methodologies use measured EOCs (Sohn, 2007). A general approach for cases were full

knowledge of EOCs is available is to regress damage sensitive features (DSFs) against EOCs to establish a relationship between these two parameter sets. Incoming datasets are then compared to the relationship and deviations from normal conditions can be evaluated base on the EOCs value. This approach requires an accurate and complete set of EOCs measurements to be available for the regression approach. Nonetheless, this approach has been implemented on structures such as wind turbines (Yampikulsakul et al., 2014) and bridges (Peeters & De Roeck, 2001; Steenackers & Guillaume, 2005).

Unfortunately, for most SHM applications, exact knowledge of EOCs is not readily available. In most field applications of SHM, traditionally, disproportionate attention is often given to selecting sensors that provide structural response data with sensors for EOCs data often receiving secondary attention. For example, there are cases were measurement of the complete set of EOCs is impractical. Also, the measured set of EOCs (if even measured) might not accurately capture the full range of EOCs influencing the response of the structure. In some other cases, the EOC data available might be those published by the site operator or weather stations which typically report EOCs in a statistical fashion with EOC parameters averaged over extended observation periods (e.g., minutes, hours) that are too large compared to the dynamics of the structure.

A growing number of methods have emerged to perform EOCs data normalization without the use of EOCs measurements and are solely reliant on structural response data. To remove the effects of EOCs variations, methods within this class use a diverse set of techniques and tools such as principal component analysis (Cross et al., 2012; Oh & Sohn, 2009; Tibaduiza et al., 2013), auto-associative neural networks (Figueiredo et al., 2011; Sohn, Worden, et al., 2001), singular value decomposition (Gomez Gonzalez, 2016; Figueiredo et al., 2011), factor analysis (Gomez

Gonzalez, 2016), robust regression and outlier detection (Dervilis et al., 2015; Dervilis et al., 2015) and cointegration (Cross et al., 2012; Cross et al., 2011; Cross & Worden, 2012; Worden et al., 2014; Zolna et al., 2015). Fundamentally speaking, these methods attempt to compensate for variations of EOCs by: 1) projecting the data to space where changes in DSFs caused by EOCs variations are clearly distinguishable from those caused by structural damage alone, or 2) selecting DSFs that are established to be insensitive to EOCs variations (Sohn, 2007).

A third class of methodologies aim at clustering the available dataset into smaller groups with each group having similar EOCs characteristics. Such methodologies incorporate an additional knowledge of EOCs for data normalization and are suitable for scenarios where the SHM practitioner deems the EOCs dataset, albeit incomplete and imperfect, worthy of integration into the broader SHM analysis framework. An important and well-established method within this class, is the Three-Tier Modular Framework developed by Häckell (Häckell, 2015). After its establishment, this framework has been applied for EOCs data normalization to numerous structures including multi-degree of freedom structures, small-scale research wind turbines, and utility grade wind turbines and their various subcomponents (Bahrami et al., 2017; Häckell, 2015; Häckell et al., 2016; Tsiapoki et al., 2018). It must be emphasized that there exists several other methodologies that attempt to group datasets with similar EOCs together, by using certain features of structural response data (Sohn & Farrar, 2001; Sohn et al., 2001) or in some other cases, by clustering in the DSFs space rather than clustering in the EOCs space (Figueiredo et al., 2014; Silva et al., 2016). Since these methods do not use EOCs data directly, they fall under the second class of methods introduced previously. Needless to say, one could combine the two spaces and perform clustering in a hybrid space consisting of both the EOCs and DSFs if necessary. Each of the previously mentioned classes of methods have their own advantages and disadvantages.

### **1.3.2 Assumptions of Independence and Identical Distribution**

The underlying assumption that dominates data-driven SHM applications is that the observations of the system (i.e. structural response records) are independent and identically distributed (i.i.d.). This stems from the fact that many of the data-driven methods (e.g., pattern classification and machine learning) used for SHM applications have been developed outside of the SHM arena. In most cases, these powerful models have gained a reputation from showing superb performance on a set of applications not necessarily related to structural engineering and dynamics. For many such applications, the assumption that the observations are i.i.d. is indeed valid. However, it is obvious that for SHM applications, the i.i.d. assumption is an idealization of the problem that jeopardizes the performance of algorithms based on it. To be more specific, the statistical independence in time is an over-simplified assumption for SHM applications as both the state of the structure under consideration and the variations in EOCs poses sequential relationships. An attractive tool for modeling the sequential nature of observations has been the Hidden Markov Models (Bishop, 2006; Zaidi et al., 2011). As for the dependence of states of a structure, in the absence of any maintenance actions, the state of the structure is only expected to deteriorate. However, for most SHM applications, a large enough dataset of data from the damaged state is often not available as most structural damage is repaired at the onset of detection in the interest of safety and cost considerations. The lack of data of the structure in a damaged state can render ML-based approaches to learning sequential relationships very difficult due to lack of training data. Clearly needed are methods of data-driven modeling of the sequential nature of structural deterioration based on a limited set of observations of the structure in its damaged state.



### **1.3.3 CPS-Enabled Infrastructure Monitoring and Assessment**

The potential of IS monitoring data extend far beyond serving as the basis of sophisticated computational models for structural assessment. The field is blessed with shrewd implementations and innovations that provide opportunities for unlocking the potentials of data-driven methods for a wide range of IS-relevant applications. Specifically, the augmentation of communication and computation with physical structures through CPS frameworks has enabled exciting new approaches to monitoring (and even controlling) system performance. Due to the availability of computational resources that scale to the data generated by sensors in a CPS framework, novel data sources that require processing to extract information can be now included. In IS applications, tremendous excitement surrounds the use of cameras to observe the loads imposed on some structures or to assess the structural conditions of the system. Computer vision (CV) algorithms are rapidly evolving including a new generation of ML-based algorithms ideally suited for object detection. The majority of CV-based techniques this are designed to automatically process large volumes of pictures and/or video feeds to extract valuable information to be used later. Specific to IS applications, CV has been applied to detect damage, measure structural displacement, and to assess human use of IS (Spencer et al., 2019). Other uses of CV in IS applications include its use for flood mapping and monitoring (Arshad et al., 2019), to track vehicles in intelligent traffic management systems (Osman et al., 2017) and to inspect wind turbines using drones (Shihavuddin et al., 2019).

An important utility of automated image processing capabilities of CV is to extract a set of information for IS application that was unavailable before. One such example is the use of CV to extract information about a structure's EOCs, especially the loads imposed on a structure. This is a major advancement because in some cases, traditional sensing technologies (e.g., discrete sensors

like accelerometers) are not applicable. One such example is the work of Hou et. al. (2019) where a cyber-physical system along a highway corridor includes a CV system that can automatically identify and re-identify vehicles crossing bridges monitored by SHM systems (Hou, 2020; Hou et al., 2019). A single-stage object detector called YOLO is embedded into road-side camera systems with graphical processing units (GPUs) included to automate the detection of large trucks. The bridge SHM systems are triggered to collect bridge response data when a truck approaching a bridge is detected. Also, a second level convolutional neural network (CNN) is trained to identify (so called reidentification) the same truck at multiple bridges to link one bridge response to another from the same truck load. As a result of this work, a unique dataset is created consisting of pairs of bridge responses to the same truck. Furthermore, by reidentifying vehicles at a weigh-in-motion (WIM) station in the same corridor, Hou. et. al. (2019) managed to build a large dataset consisting of bridge responses to key vehicle weight and geometric features such as vehicle axle spacing and axle weight distributions. Such datasets are extremely valuable as, to the best of the author's knowledge, it is one of the very few (if not only) dataset that contains pairs of bridge responses along with the corresponding load information derived from truck properties (including their weight) collected under routine operations. It is anticipated that with the increase of deployed monitoring systems on highway corridors, similar systems will be employed resulting in similar datasets that includes structure responses with the loads inducing them. With increased attention being paid to self-driving trucks and cars with wireless connectivity, vehicle-to-infrastructure (V2I) interactions will be possible offering an alternative approach to CV for vehicle load monitoring (Elliott et al., 2019).

Numerous motivations exist for focusing on highway corridors when exploring deployed CPS architecture that can collect both structural response data but also EOC data. From the

viewpoint of analyzing characteristics of traffic flow, CPS-enabled highway corridors are frequently explored (D'Angelo et al., 1999; Song et al., 2018; Yao et al., 2017). Modeling the flow of traffic has direct relevance on structural assessments by guarantying the number and speed of vehicles loading on asset like a bridge. Additionally, when outsourcing highway maintenance operations to private entities, highway corridors are also one of the most common IS types being managed by private contractors under PPP contracts (Duncan et al., 2014).

The introduction of cameras and CV into CPS-enabled highway corridors enables the opportunity to more broadly apply ML- and deep learning (DL)-based data processing techniques for solving existing challenges with highway corridor management. A key challenge is to forecast the complex interactions between truck loads and highway bridges which dominate the strain response of bridges to moving loads. Accurate response forecasting is key to enabling better understanding of vehicle-bridge dynamic interactions (VBI). VBI has received significant attention over the past few years and the body of literature allocated to VBI has been ever increasing since the widespread adoption of numerical models (Ding et al., 2010; Elhattab et al., 2016; González et al., 2008; Zhang et al., 2018; Zhao & Uddin, 2014). While VBI remains an active area of research, it is widely recognized that numerical models have limitations. Data from operational bridges could be invaluable for advancing new VBI modeling approaches.

With a high number of bridges now instrumented to measure their response to traffic loads, use of the bridge response data for estimating truck properties has also received significant attention; this problem is commonly known as bridge weigh-in-motion (BWIM). State Departments of Transportation (DOTs) in the US operate weigh-in-motion (WIM) stations across their highway networks to assess traffic volumes and total carried weightage on their roads. WIM stations measure valuable traffic information that can be used for future highway design and

planning. More importantly, WIM station information is also used to determine the share of funding that each state receives from the federal government. WIM station sensors are buried underneath the highway pavement and record truck weight and geometry characteristics as they pass over the embedded sensor. Consequently, WIM stations are expensive to install and often challenging to maintain. This has been the main motivation for considering BWIM as bridges are already being instrumented for structural monitoring. Thus, by developing BWIM solutions, researchers hope to take full advantage of existing instrumentation systems to derive more value from the data collected. Additionally, bridge sensors are often easier to maintain and repair compared to WIM sensors that have accessibility issues due to being buried underneath the road surface. Consequently, over the past decade, BWIM solutions have evolved towards using simpler bridge instrumentation by using more complex data analytics tools. Early BWIM solutions deployed axle detectors (such as laser sensor) on bridge road surfaces (Hou, 2020). However, recent BWIM methods rely on sensors measuring bridge responses alone. Many such methods use sophisticated models relying on a broad range of tools ranging from the fields of signal processing to optimization and DL (Pan et al., 2018; Wu et al., 2020; Y. Zhou et al., 2021).

With the existing opportunities for applying DL-based techniques to recently available bridge response data, two major gaps within the literature are prevalent; first is the bridge response forecasting over an entire truck population and second, implementation of BWIM models using real-world data. In this context, the diverse set of trucks traversing the bridges act as a broad EOC set dominating the response of bridges. Consequently, the application of DL-based models must be done with diligent focus devoted to the effects of EOCs on the performance of the data-driven models themselves. This would be akin to the aforementioned data normalization methods previously described for SHM applications.

## 1.4 Research Goal and Objectives

The overarching goal of this dissertation is to contribute to solving challenges in applying data-driven techniques for structural monitoring and assessment of IS assets. The key concept linking all these contributions is the focus on EOCs data normalization when applying data-driven techniques in this application domain. The dissertation aims to develop data-driven analytical frameworks that explicitly extract EOCs parameters to improve structural performance assessments. To accomplish this goal, the dissertation has three main objectives. First, it proposes a methodology for estimating features that heavily correlate with EOCs (also known as EOC sensitive features, or EOCSFs) from measured structural responses for IS scenarios where proper knowledge of EOCs are not available. Second, by employing the basic laws of probability and by building on previously proposed damage detection frameworks, new frameworks are explored to overcome the shortfalls of prior SHM frameworks and increase the damage detection performance. Finally, by using the unique dataset constructed by Hou et. al. (2019) and using the computation capabilities of DL-based time series analysis models, two solutions are proposed for better highway corridor management using bridge monitoring data: bridge-to-bridge response forecasting and BWIM. These methods consider the effects of EOCs on the performance of the DL-based models adopted. Two types of IS assets will be explored in the dissertation: wind turbines and highway bridges (in a CPS architecture). Both see major loads as part of their EOC profile yet there are challenges to fully measuring these EOCs as will be described later. The following paragraphs highlight key research objectives inherent to the dissertation.

The first objective of this dissertation is to propose a framework for systematically searching for EOCSFs when EOCs data is not available at the IS asset location. The term EOCs Sensitive Features (EOCSFs) is intentionally given to highlight the contrast between these features

and previously described Damage Sensitive Features (DSFs). To determine EOCSFs, a mechanical model of a structure (in this dissertation's case, a wind turbine) is built to enable simulating the response of the structure under varying EOCs. The benefit of conducting simulations is having complete control over the variations of EOCs. Regression models are used to explore the relationships between potential EOCSFs and EOCs using the simulated data and the degree to which these features correlate to EOCs. It is desirable for EOCSFs to be as insensitive to structural damage as possible since the EOCs data normalization stage must not depend on the structural status of the structure. Hence, the sensitivity of the EOCSFs to structural damage will be evaluated. To do so, the real-world damage scenario must be mimicked in the simulated model and the EOCSFs-EOCs relations are tested again using the regression model. It is recommended to search for alternative EOCSFs should a potential EOCSF show high sensitivity to structural damage.

The second objective of this dissertation is to propose EOC-aware damage detection frameworks for enhancing the performance of the existing SHM systems for real-world applications. Specifically, two innovative damage detection techniques are explored based on relaxing the limitations and idealized assumptions underlying the previously used damage detection frameworks. First, a Gaussian Mixture Model (GMM)-based probabilistic assignment approach is considered for EOC data normalization. Rather than considering the observations in one cluster and comparing the DSFs of that observation with the DSFs of the same EOC class, the DSFs of each observation is compared to all other EOCs classes. However, when combining the results, higher weight is given to sample if they are from EOC classes that the particular observation has a higher likelihood of belonging to. Compared to previously used data normalization approaches, the proposed method will take advantage of the entirety of the training

dataset while consciously favoring the results from “more similar” EOCs classes. The efficacy of this approach will be shown on wind turbine systems.

The second damage detection technique proposed is to relax the i.i.d. assumption existing within the majority of ML-based damage detection frameworks in practice. As such, a sequential damage detection framework using Hidden Markov Models (HMM) is proposed under varying EOCs. Rather than looking at individual observations, the HMM-based damage detection framework takes in sequences of structural responses measured at different instances. The HMM assigns likelihoods of observation to each sequence and all observations in a sequence that have a significantly low likelihood are labeled as damaged. Conversely, if the likelihood of observation for that sequence isn't low, the entire observation set is then labeled as undamaged. To evaluate the capabilities of the proposed SHM framework, its performance is compared with other damage detection frameworks that rely on the i.i.d. assumption. This damage detection technique will be explored using a bridge system with a controlled damage evolution process.

The third and final objective of this dissertation is to develop exclusive data-driven models for solving existing challenges within the practice of highway infrastructure asset management by utilizing DL models. Specifically, the novelty of this part of the work is to expand beyond the current unit for analysis, which is a single bridge, and utilize the unique and state-of-the-art dataset collected by Hou et. al. (2019) to propose models for corridor monitoring applications. Two models are developed, one for joint bridge response modeling (i.e., prediction) and the other for BWIM applications. There are two reasons for resorting to DL-based models given these datasets. First, the carefully designed complexities and nonlinear transformations imbedded within these models result in superior performance compared to their shallower counterparts. Second, the

unique architecture of these DL models makes them suitable for expanding to offer corridor monitoring applications with ease.

The first DL model to be explored, namely a sequence-to-sequence (Seq2Seq) model which is used to enable joint bridge response forecasting. This is done by forecasting the response of a downstream bridge to a given truck by using the response of an upstream bridge that has responded to the same truck. Such a solution has two goals: the first is to enable bridge response forecasting with limited instrumentation. Specifically, the approach is well suited for permanently instrumenting one bridge and deploying temporary instrumentations on other bridges across the corridor to gather enough data for training the DL-based predictive models. By doing so, an important step is taken towards transitioning from the existing highway financing models that mainly rely on fuel tax to models that quantify how much each vehicular user consumes of the infrastructure's life and charge for that consumed life. The second goal is to use the response of the bridge to previous similar trucks for controlling connected and automated trucks by prescribing trajectories so as to minimize the bridge consumed life and maximize the life span of the existing bridges. The key operational condition influencing the performance of the proposed data-driven model is the characteristics of the vehicle passing through the corridor. Next, using the bridge response data from the corridor, along with corresponding WIM station information for the truck, a DL-based model is developed for the BWIM problem.

## **1.5 Dissertation Outline**

Figure 1–2 shows the outline of this dissertation. In Chapter 2, the framework for estimating EOCSFs is developed. The structure of interest is the tower of a small scale experimental wind turbine located at the Los Alamos National Laboratory (LANL). To simulate the turbine's



response, three pieces of software are used to develop a mechanical model for the LANL turbine. First, a model is developed in the FAST (Jonkman & Buhl Jr., 2005) simulation platform to model the dynamic response of the wind turbine to wind input. Realistic response scenarios are simulated by generating wind profiles using TurbSim (Jonkman & Kilcher, 2012). Finally, a model is developed in MATLAB to simulate the response of the tower to the nacelle loads as evaluated by FAST. While FAST has a built-in feature to compute the tower response, a MATLAB model is developed for higher accuracy (by considering more structural modes than FAST's default number of modes) and higher flexibility for implementing different damage scenarios. Gaussian Process Regression is used to evaluate the relationships between the proposed EOCSFs and EOCs under undamaged and damaged towers. Furthermore, the GMM-based damage detection framework is developed in this chapter and the performance of the EOCSFs, as well as the proposed damage detection framework, is tested on both the simulated dataset and the experimental dataset from the LANL wind turbine.

Chapter 3 of this dissertation explores the sequential relations amongst observations for SHM applications. A sequential outlier detection framework using Hidden Markov Models is proposed for damage detection under varying EOCs. To evaluate the capabilities of the proposed SHM framework, the performance of the proposed framework is compared with other damage detection frameworks that rely on the i.i.d. assumption. The proposed frameworks are implemented on the well-known Z-24 bridge dataset. This dataset is extremely valuable for SHM applications as it is one of the very few available that contain measurements from the bridge while

realistic and common damage scenarios are physically implemented on numerous structural components of the bridge over a prolonged period of observation time.

In Chapter 4, an Auto-Encoder is used for BWIM. A Finite Element model for simulating

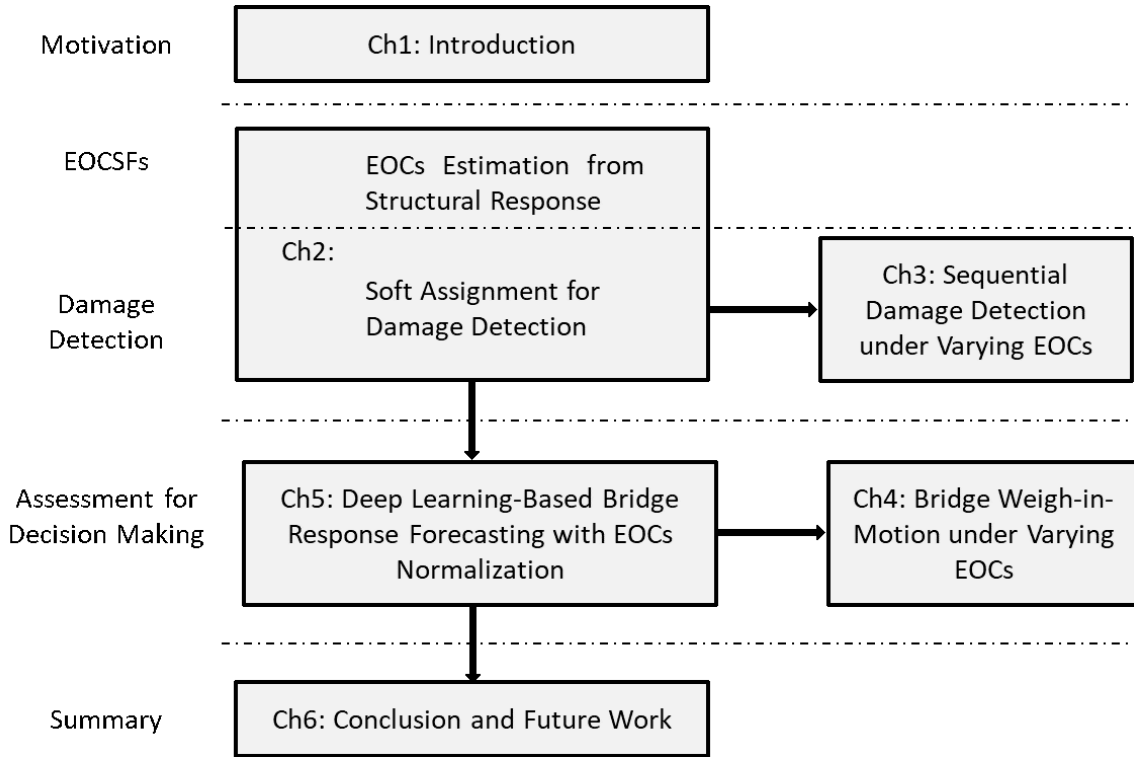


Figure 1-2. Thesis Outline.

the response of the bridge to truck traffic is developed to first enable analyzing the performance of the Encoder under ideal scenarios. The same approach is used for BWIM on the experimental data collected by Hou et. al. (2019). This data is from the I-275 highway corridor located in southeast Michigan. The measurement system consists of two instrumented bridges, namely the Newburg Road Bridge (NRB) and the Telegraph Road Bridge (TRB), a WIM station and four roadside traffic cameras.

In Chapter 5, a Seq2Seq model is trained for joint bridge response forecasting. Specifically, the response of a bridge to the same truck is used to forecast the response of another bridge to that identical. The experimental dataset from Chapter 4 is used to train the Seq2Seq model. Two

distinct problems are solved; the first one, the NRB response is used as the input to the model to forecast the TRB response and in the other one, the direction of forecasting is reversed. The detailed description of the signal pre-processing techniques for removing noise, isolating the bridge free vibration and isolating observations with a single truck is included in this chapter. In this chapter. It will be observed that by focusing on one vehicle class only, the performance of the forecasting framework is enhanced. This suggests that the operational condition of interest that has the most significant impact on the model performance is the type of vehicle travelling over the bridges.

This dissertation concludes by listing the key findings, along with the directions of future work in Chapter 6.

## **Chapter 2.**

### **Two Methods for Better EOCs Data Normalization for SHM: Use of EOCs-Sensitive Features and Soft Assignment**

#### **2.1 Introduction**

Wind power is amongst the fastest growing renewable energy sources. An important step in enhancing the marketability of wind power is optimizing the operation and maintenance costs of wind turbines (Byon & Ding, 2010). A key tool for optimizing the maintenance costs of wind turbines is the implementation of a comprehensive condition-monitoring system, capable of providing diagnostic information on the health of turbine subsystems and informing maintenance crew to trends that could potentially lead to turbine failure (Walford, 2006). Consequently, wind turbine condition monitoring systems and strategies have received significant attention over the past years. Reviews of such strategies can be found in the literature (Ciang et al., 2008; García Márquez et al., 2012). Of primary importance amongst all turbine monitoring systems are the structural health monitoring (SHM) systems as structural damage is capable of inducing catastrophic damage to the integrity of the wind turbines (Ciang et al., 2008).

Variations in EOCs play a significant role in the structural response of a wind turbine. Specifically speaking, wind speed and wind direction in particular, are the main drivers of the turbine blades which in turn run the turbine in the nacelle. As a result, the response of each component within the turbine is significantly influenced by EOCs. In chapter 1 of this dissertation, the importance of EOCs data normalization for SHM along with a review of existing approaches to EOCs data normalization were

thoroughly discussed. In this chapter, two solutions for better EOCs data normalization for wind turbines is pursued.

First, for the case of imperfect EOCs, we seek features from structure's response that enhance the efficiency of data normalization. The basic premise of this approach is that for SHM applications of interest, one, the measured structure response data has much higher quality compared to the available EOCs data and two, EOCs variations leave a trace in the structure response data. This would enable exploring features from structure response data that are sensitive to EOCs (as opposed to structural damage), which from here on, would be referred to as EOC sensitive features or in short, EOCSFs. The first assumption is valid for most practical field measurements used for SHM as discussed before. The second assumption is reasonable as well since if the variations in EOCs are capable of adversely influencing the performance of the SHM practice, they are expected to have meaningful signatures in the measured structure response. The EOCSFs used for data normalization must be as insensitive to structural damage as possible because the data normalization procedure used for SHM must be independent of the damage status of structure. Our approach to find EOCSFs is first to build a mechanical model for the structure and use the simulation environment, where full knowledge of and control over EOCs is available, to extract EOCSFs from structure response data. Next, the sensitivity of EOCSFs to structural damage are evaluated to ensure that data normalization stage would not suffer from the dependence of EOCSFs on structural damage.

The second contribution of this chapter is to introduce a novel data normalization approach to enable all clusters to contribute when deciding on the status of an observation. Partitioning the entire dataset into a finite number of smaller clusters has two short-comings inherent in its nature. First, many datasets could lie within the boundaries of two clusters. In the traditionally used clustering approach, they are forced to be assigned to one and only one cluster despite having high probability

of belonging to another clusters as well. This could cause errors both within the training and testing stages. The second issue is that within the traditional clustering approach, only a single cluster is used to decide whether an observation is damaged or not. Thus, this approach tends to limit itself in the decision making process to smaller subset of data available and hence appears to be sub-optimal as it does not take full advantage of the entire valuable training dataset available. These two issues especially become important when the size of dataset is relatively large and samples are well-spread over the entire span of observation. Our solution for overcoming these two limitations of clustering approach, is to design a new EOCs data normalization methodology that enables all clusters to play a role in deciding on the status of an observation.

The efficiency of the two proposed solutions of this research are evaluated using an experimental dataset measured from a small-scale wind turbine located at LANL and a simulated dataset for the same turbine. For the purpose of this work, small scale wind turbines are perfect structures as their structural response is driven by loads that depend heavily on EOCs (i.e. wind properties) and they do not possess the large inertia of utility grade wind turbines within their dynamic response.

The rest of this chapter is organized as follows: next section presents the methodology and theoretical background of this work. Section 3 introduces the experimental dataset and familiarizes the reader with the shortcomings of this dataset collected under typical field situations. Additionally, it introduces the simulation datasets and why and how they were generated. Finally, in section 4 the results of implementing the damage detection framework with the two proposed solutions are shown. The chapter concludes by summarizing key findings.

## **2.2 Theory and Methodology**

This section of the chapter is devoted to describing the research methodology and the theoretical background.

In this chapter, two assignment approaches of data normalization for structural health monitoring are presented. The first approach is hard assignment approach which uses features from structural response to help yield more accurate estimates of the effects of EOCs on the structural response and overcome the lack of high quality EOCs data. The second approach is soft assignment approach for data normalization.

### **2.2.1 Damage Detection Framework**

The damage detection framework used within this study is called the three-tier modular framework (Häckell, 2015; Häckell et al., 2016). As it appears by the name, the three-tier framework consists of three main tiers. At the training stage, the training dataset is normalized with respect to the EOCs in the first tier. This is done by dividing the dataset into smaller classes such that the data instances within each class have similar EOCs. In the second tier, DSFs, also known as condition parameters (CPs), are extracted for each observation. Ultimately, in the third tier, decision boundaries are established for damage detection using pre-determined probabilistic models and significance levels. At the testing stage, in the data normalization tier (first tier), the incoming observations are assigned to the classes to which they have the most similar EOCs. DSFs are extracted in the second tier and compared to decision limits and labeled as damaged or undamaged in the last tier.

This three-tier framework has been utilized previously for various SHM applications including the LANL small scale wind turbine which is the subject of this study (Häckell et al., 2016). This proposed framework is pure data-driven machine learning which is only trained by undamaged data without any underlying mechanical model of the structure. Since it is a challenge to identify when a structure under service change its status, we emphasize that the framework is only trained by undamaged data to meet the needs in real-world, because, in general, the structures

under services can only provide undamaged information. Furthermore, it is desired that the framework could detect damaged observations without presenting those observations to the training stage in the proposed framework. In other words, all the training data are from the undamaged observations to train the framework with unsupervised learning implementation. In the testing stage, the trained framework is evaluated using a mixture of damaged and undamaged datasets to determine the existence of damage. It is worth mentioning that according to axiom III of SHM (Farrar & Worden, 2012; Rytter, 1993) further identifying the type of damage present and damage severity requires supervised learning.

In summary, this framework has two key advantages that makes it an outstanding option for this research. First, it introduces a systematic approach to EOCs data normalization utilizing novel machine learning technique within the context of SHM. Secondly, it is highly modular which provides the practice of SHM and damage detection with high level of flexibility. This framework enables the SHM engineer to put different pieces into place at each module to create a new damage detection system and the best combination can be selected for the specific application by evaluating different options.

### 2.2.1.1 *K-means Clustering*

K-means is a clustering algorithms that aims at partitioning an N observations  $\{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_N\}$  into a predetermined K number of clusters with the nearest mean by minimizing the following objective function (Bishop, 2006):

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{e}_n - \boldsymbol{\phi}_k\|^2 \quad \text{Eq. 2-1}$$

$$\boldsymbol{\phi}_k = \frac{\sum_n r_{nk} \mathbf{e}_n}{\sum_n r_{nk}} \quad \text{Eq. 2-2}$$



where  $\phi_k$  is the center of  $k^{\text{th}}$  cluster,  $\|\cdot\|$  represents the Euclidian distance between two points, and  $r_{nk}$  is a binary variable whose value is 1, if  $e_n$  belongs to the  $k^{\text{th}}$  cluster, otherwise, the value is zero (note that each observation must be assigned to one and only one cluster).

In general, an iterative approach is utilized to chase the best partition by alternating between two steps: (1) assignment step, initial  $\phi_k$  values are assumed and the objective function is minimized with respect to  $r_{nk}$ , by assigning each observation to the appropriate cluster; (2) updating step,  $r_{nk}$  values are assumed to be constant and the objective function is minimized with respect to  $\phi_k$ . For the first step, for the  $n^{\text{th}}$  observation, to minimize the objective function one has to assign that observation to cluster with smallest  $\|e_n - \phi_k\|$  value (i.e. cluster with the closest center). For the second step, by evaluating  $\partial J / \partial \phi_k$  and setting it to zero, it can be shown that the objective function is minimized by selecting (Bishop, 2006):  $\phi_k$  is effectively the mean of the observations assigned to the  $k^{\text{th}}$  cluster. For this reason, the algorithm is called *K*-means.

It is noted that different objective functions and different measures for distance of Equation 2-1 would result in a different formula for updating in Equation 2-2. These two steps of assigning observations to closest cluster and updating the cluster center correspond respectively to the expectation and maximization steps of an algorithm known as expectation-maximization (EM) (Dempster; N. M., 1977). This iterative procedure is continued until no further change in the cluster assignment is observed. Since after each iteration the value of the objective function decreases the procedure is guaranteed to converge (Bishop, 2006). In addition, multiple random initial values of  $\phi_k$  are generated to have a higher chance of reaching the global minimum instead of a local one.

Within this study, *K*-means clustering is carried out in the EOC space to group the observations with similar EOCs together. This is done either by clustering using features that are

direct measure of EOCs (such as wind speed and wind direction) or features that are shown to correlate with EOCs.

### 2.2.1.2 Vector-Autoregressive Models (VAR)

The DSFs used in this work are extracted from time-series of the vibration record based on residues of vector autoregressive (VAR) models. VAR model is an extension of the autoregressive (AR) model which enable the extraction of scalar DSFs from time series consisting of multiple measurements channels.

To extract the DSFs, a VAR model is fitted to an undamaged dataset. Then, the trained model is employed to describe a test dataset whose state is unknown. When a new dataset comes into the model, a decision is made compared to the distribution of the undamaged dataset. If the error is low, then the new dataset can be described by the undamaged model relatively accurately and hence the test data is concluded to be undamaged as well. In contrary, a large error implies the test dataset is from a damaged state because the undamaged model cannot describe the test dataset accurately.

In mathematical forms, suppose  $\mathbf{z}_1[t] \in \mathbb{R}^{M \times 1}$  ( $t = 1, 2, \dots, T_{z_1}$ ) is an  $M \times 1$  vector representing measurements from the  $M$  sensors installed on the structure in the reference undamaged state. Then the VAR model is fitted as:

$$\mathbf{z}_1[t] = \mathbf{w} + \sum_{p=1}^P \mathbf{A}_p \mathbf{z}_1[t-p] + \boldsymbol{\varepsilon}[t] \quad \text{Eq. 2-3}$$

where  $p + 1 \leq t \leq T_{z_1}$ ,  $\mathbf{A}_p \in \mathbb{R}^{M \times M}$  is the weighting coefficient matrix,  $\mathbf{w}$  is  $\mathbb{R}^{M \times 1}$  vector used to capture the bias of the model,  $\boldsymbol{\varepsilon}$  represents the error of model in describing the measurements, and  $P$  is the order of VAR model. Once the model order is selected, the remaining model parameters

are determined by a least-squares approach. The code developed by Schneider and Neumaier (Schneider & Neumaier, 2001) is used to fit the VAR models to the reference datasets.

Suppose  $\mathbf{z}_2[t]$  ( $t = 1, 2, \dots, T_{z_2}$ ) represents the time series of a test dataset. To find the DSFs, the same model is also used to describe its time series:

$$\hat{\mathbf{z}}_2[t] = \mathbf{w} + \sum_{p=1}^P \mathbf{A}_p \mathbf{z}_2[t-p] \quad \text{Eq. 2-4}$$

where  $\hat{\mathbf{z}}_2$  represents the estimate of the  $\mathbf{z}_2$  time series using the VAR model. The error,  $\boldsymbol{\epsilon}[t]$ , is the difference between the actual time series and the estimated time series:

$$\boldsymbol{\epsilon}[t] = \mathbf{z}_2[t] - \hat{\mathbf{z}}_2[t] \quad \text{Eq. 2-5}$$

It is noted that the length of time series from testing dataset,  $T_{z_2}$ , might be different from that of  $\mathbf{z}_1$ , but the size of these vectors must be the same (i.e.,  $\mathbf{z}_2[t] \in \mathbb{R}^{M \times 1}$  should be an  $M \times 1$  vector as well).

The coefficient of determination of the VAR model is termed (Copeland, 1997) :

$$R^2 = \frac{1}{M} \sum_{m=1}^M \left( 1 - \sqrt{\frac{\sum_{t=P+1}^{T_{z_2}} (z_{2m}[t] - \hat{z}_{2m}[t])^2}{\sum_{t=P+1}^{T_{z_2}} (z_{2m}[t] - \bar{z}_{2m})^2}} \right) \quad \text{Eq. 2-6}$$

where  $z_{2m}[t]$  represents the  $m^{\text{th}}$  component of the vector  $\mathbf{z}_2[t]$ . Additionally,  $\bar{z}_{2m}$  is the time average of the measurements  $z_{2m}[t]$  with  $t$  ranging from 1 to  $T_{z_2}$ . The first DSF of this work is based on  $R^2$ , which is defined as:

$$DSF^{R^2} = (1 - R^2) \frac{T_{z_2} - 1}{T_{z_2} - P - 1} \quad \text{Eq. 2-7}$$

It is noted that  $R^2$  approaches unity for perfect fit so  $DSF^{R^2}$  as defined above, attains small values for measurements from an undamaged structure and tends to be larger for damaged datasets.

The second DSF is based on the M-test by Box (Box & Box, 2011) and compares the consistency of the estimated covariance matrix of the fitted model to test dataset with that of a single, or potentially many reference datasets. The estimate of covariance matrix is defined as:

$$\hat{\Psi}_{z_2} = \frac{1}{T_{z_2} - P - 1} \sum_{t=P+1}^{T_{z_2}} \epsilon[t] \epsilon[t]^* \quad \text{Eq. 2-8}$$

where  $\epsilon[t]^*$  represents the Hermitian transpose of the  $\epsilon[t]$ . This DSF, termed  $DSF^M$  is defined as:

$$DSF^M = (\tilde{t} - G) \ln|\Psi| - \sum_{g=1}^G (\tilde{t}_g - 1) \ln|\Psi_g| \quad \text{Eq. 2-9}$$

where  $\tilde{t} = \sum_{g=1}^G \tilde{t}_g$ ,  $G$  is the number of reference datasets,  $\tilde{t}_g = T_g - P - 1$ ,  $T_g$  is the total number of time instances in the  $g^{\text{th}}$  dataset, and  $\Psi$  is the pooled covariance matrix of the test dataset and the other  $G - 1$  reference datasets given as:

$$\Psi = \frac{1}{\tilde{t} - G} \sum_{g=1}^G (\tilde{t}_g - 1) \Psi_g \quad \text{Eq. 2-10}$$

For a more detailed description of the DSFs and their derivation, the reader is referred to (Häckell, Moritz W.; Rolfes, 2013; Häckell, 2015).

### 2.2.1.3 Decision Boundaries and Hypothesis Testing

As mentioned previously, the three-tier framework is trained only using undamaged data. Hence this can be thought of as testing the null hypothesis that the observation is coming from a structure in undamaged conditions. For the damage detection problem, rejecting the null hypothesis would result in accepting the only alternative hypothesis which states the observation from damaged structure. The ultimate goal of training is to set decision boundaries on DSF values. If the DSF corresponding to an observation is within the boundaries, that observation is labeled as

undamaged; otherwise, it is labeled damaged. To set the decision boundaries, a known distribution is fitted to the training DSFs and boundaries are set by selecting a threshold (i.e., an  $\alpha$  value, also known as significance level). After completing the training stage, for each observation in testing stage the following procedure is conducted. Suppose  $s$  is the state of the structure associated with observation  $\mathbf{o}$  during hypothesis testing (which for the problem at hand, could be either undamaged or damaged). Then by comparing it to the distribution corresponding to undamaged training datasets, if  $P(s = UD) < \alpha$  then  $\mathbf{o}$  is labeled damaged. It is labeled undamaged if otherwise ( $P(s = UD) \geq \alpha$ ). Ideally, we seek to achieve a framework that correctly labels all damaged and undamaged observations in the testing stage. However, for most of the applications, when damage is not in critically high levels, that is not achievable and there would be false classification (e.g., labeling an observation as damaged when in fact it is undamaged).

It is emphasized that the threshold is selected by the engineer implementing the framework and is dependent on the consequences of misclassifying the observations. Misclassifying an undamaged observation as damaged would result in a structure being taken out of service for further inspection unnecessarily which results in loss of revenue and misclassifying a damaged observation could result in severe damage to the structure or loss of life (Farrar & Worden, 2012). It is up to the engineer to take into account these consequences when selecting the threshold for damage detection. For SHM applications, when structure is in good conditions, typically the threshold is selected so that most healthy datasets fall within the correct boundaries.

## 2.2.2 Soft Assignment for Data Normalization

### 2.2.2.1 Notion of Soft Assignment

The aforementioned approach for data classification has two inherent shortcomings: Firstly, by clustering the data and only comparing each new observation with the closest cluster, we cannot get the full benefits of the entirely valuable training dataset. Secondly, this all-or-none binary assignment cannot precisely describe the data near the boundaries, because a slightly change might cause the data point falling into another cluster.

To overcome these challenges, a soft assignment approach is proposed to generate the possibility of the observation belonging to each cluster instead of assigning it to one and only one cluster for define decision boundaries in the three-tier framework. Suppose the entire training dataset is broken into  $K$  mutually exclusive and collectively exhaustive clusters  $C_1, C_2, \dots, C_K$ , and  $\sum_{k=1}^K P(\mathbf{o} \in C_k) \equiv 1$ . When new observation,  $\mathbf{o}$ , with state,  $s$ , comes in the testing stage, the probability that  $\mathbf{h}$  is undamaged can be expressed as:

$$P(s = UD) = \sum_{k=1}^K P(s = UD, \mathbf{o} \in C_k) \quad \text{Eq. 2-11}$$

$$P(s = UD, \mathbf{o} \in C_k) = P(s = UD | \mathbf{o} \in C_k) P(\mathbf{o} \in C_k) \quad \text{Eq. 2-12}$$

Combining Equation 2-11 and Equation 2-12:

$$P(s = UD) = \sum_{k=1}^K P(s = UD | \mathbf{o} \in C_k) P(\mathbf{o} \in C_k) \quad \text{Eq. 2-13}$$

Equation 2-13 introduces the notion of soft assignments in data normalization. To label the observation,  $\mathbf{o}$ , the possibility of undamaged,  $P(s = UD)$ , is compared to the confidence interval,  $\alpha$ , similar to the aforementioned method of hypothesis testing. If  $P(s = UD) < \alpha$  then  $\mathbf{o}$  is labeled damaged, otherwise, labeled as undamaged.

The advantage of using this proposed soft assignment is the fact that it resolves the previously mentioned shortcomings associated with hard assignment clustering approach. This is done by the term  $P(\mathbf{o} \in C_k)$  present on the right hand side of Equation 2-13. When performing hard assignment,  $P(\mathbf{o} \in C_k)$  has a one-out-of- $K$  representation meaning that it is one for exactly one of the values of  $1 \leq k \leq K$  and zero for all others. In other words, dataset  $\mathbf{o}$  is assigned to one and only one cluster and it is assumed that there is zero probability of  $\mathbf{o}$  belonging to other clusters. However, relaxing that constraint results in resolution of the previously mentioned shortcomings. The main drawback of this approach is that it is computationally more expensive compared to the hard assignment approach since for each observation, one has to compute  $P(s = UD|\mathbf{o} \in C_k)$  for all  $K$  clusters, while in the hard assignment approach, one has to compute  $P(s = UD|\mathbf{o} \in C_k)$  only for the cluster that  $\mathbf{o}$  is assigned to. Higher number of clusters would make this approach more expensive as well, however, typically for data normalization for SHM, the number of clusters is not considered to be very large. The term  $P(s = UD|\mathbf{o} \in C_k)$  is computed using the exact similar approach as in the hard assignment case, i.e. via assuming  $\mathbf{o}$  only belongs to the cluster under consideration.

Towards computing  $P(s = UD)$  using soft assignment, one could take different approach towards computing  $P(\mathbf{o} \in C_k)$  but the important point is that this term is solely computed using the EOCs. The only constraint from a data normalization perspective is that these terms must be valid probabilities. In this work, we use Mixture of Gaussians for computing  $P(\mathbf{o} \in C_k)$ . A more complete detail for training the Mixture of Gaussians can be found in (Bishop, 2006), however, in the following subsection, an abstract derivation is presented to provide the reader with necessary background.

### 2.2.2.2 Gaussian Mixture Models

Suppose an observation  $\mathbf{o}$  with an EOC feature vector of  $\mathbf{e}$  is available and there exist  $K$  distinct cluster groups  $C_1, C_2, \dots, C_K$  for the EOCs. Furthermore, the  $k^{\text{th}}$  cluster,  $C_k$  is assumed to follow a Gaussian distribution with mean vector,  $\boldsymbol{\mu}_k$ , and covariance matrix,  $\boldsymbol{\Psi}_k$  (Bishop, 2006). The probability of observation  $\mathbf{o}$  occurring can be expressed as:

$$P(\mathbf{o}) = \sum_{k=1}^K P(\mathbf{o}, \mathbf{o} \in C_k) = \sum_{k=1}^K P(\mathbf{o} \in C_k)P(\mathbf{o}|\mathbf{o} \in C_k) \quad \text{Eq. 2-14}$$

$$P(\mathbf{o}|\mathbf{o} \in C_k) = \mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k) \quad \text{Eq. 2-15}$$

Bayes' theorem is employed to calculate  $P(\mathbf{o} \in C_k|\mathbf{o})$  in Equation 2-15:

$$P(\mathbf{o} \in C_k|\mathbf{o}) = \frac{\pi_k \mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)} \quad \text{Eq. 2-16}$$

where  $\mathcal{N}(\mathbf{e}|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)$  represents the Gaussian mixture probability distribution function,  $\pi_k$  is the prior probability that the observation belongs to the  $k^{\text{th}}$  cluster,  $\mathbf{o} \in C_k$ , and  $P(\mathbf{o} \in C_k|\mathbf{o})$  is the posterior probability that  $\mathbf{o} \in C_k$  given the observation of  $\mathbf{o}$ .

So for a training dataset of  $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_N\}$  consisting of independent and identically distributed observation with EOC features  $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$ , the logarithm of likelihood function can be expressed as:

$$\ln P(\mathbf{O}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Psi}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{e}_n|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k) \right\} \quad \text{Eq. 2-17}$$

and for each of the observations:

$$P(\mathbf{o}_n \in C_k|\mathbf{O}) = \frac{\pi_k \mathcal{N}(\mathbf{e}_n|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{e}_n|\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)} \quad \text{Eq. 2-18}$$

the term  $P(\mathbf{o}_n \in C_k|\mathbf{O})$  is known as the responsibility that cluster  $C_k$  takes for observing  $\mathbf{o}_n$ .



## 2.3 Description of the Dataset

### 2.3.1 Experimental Dataset

The subject structure of this study is a small-scale research wind turbine tower located at Los Alamos National Lab (LANL, Los Alamos, NM, USA). The wind turbine is a Whisper 500, manufactured by Southwest Windpower, Inc., with a nominal power of 3 kW (10.5 m/s wind speed) and two blades whose rotor diameter is 4.5 m, as shown in Figure 2-1. The Whisper 500 wind turbine with the locations of sensor nodes and the damage scenario.. A 70 kg nacelle is placed at the top of the tower with a rotor whose nominal speed is 500 r/min (8.2 Hz). The wind turbine has a tail stabilizer which enables the turbine to follow the wind direction under normal operation and a furling mechanism that protects the turbine in high wind events by changing the rotor orientation. The EOCs data was recorded by a 46-m meteorological tower with four levels, TA-49 station, located on the Pajarito Plateau in an open meadow (N 35.8133°, W 106.2993°, elevation 214.58 m) . It records the environmental conditions every 15 minutes since June 24, 1987.

The tower of the turbine is 12.2m tall made of steel with varying cross-sections. The tower is supported by two pins: one is located at the bottom of the structure and the other one located at about 4 m height from the bottom (almost one third of the tower). The bottom pin can be removed to tilt the turbine for maintenance and inspection purposes. Since it is impossible to create a physically damage on this experimental wind turbine tower, the damage scenario is introduced by replacing the bottom pin with a spring to mimic the loss of stiffness to create damaged datasets.

Six wireless sensing nodes are mounted on the 0.72 m, 1.26 m, 2.86 m, 8.04 m, 9.18 m, and 10.23 m of the tower to measure triaxial accelerations to generate datasets. Each sensing node consists of a Martlet wireless unit, a tri-axial accelerometer, a 5-V power control board, and an accelerometer connecting base board (Kane, 2014). While a triaxial accelerometer is used in the

sensing node, only the accelerations in the horizontal plane (x- and y- directions) are recorded for

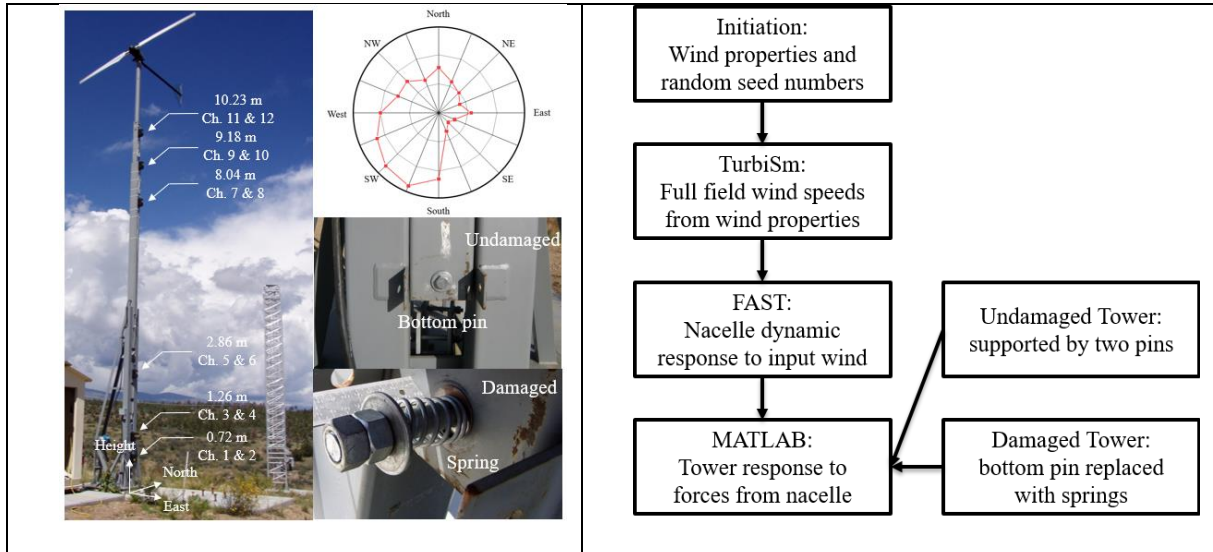


Figure 2-1. The Whisper 500 wind turbine with the locations of sensor nodes and the damage scenario.

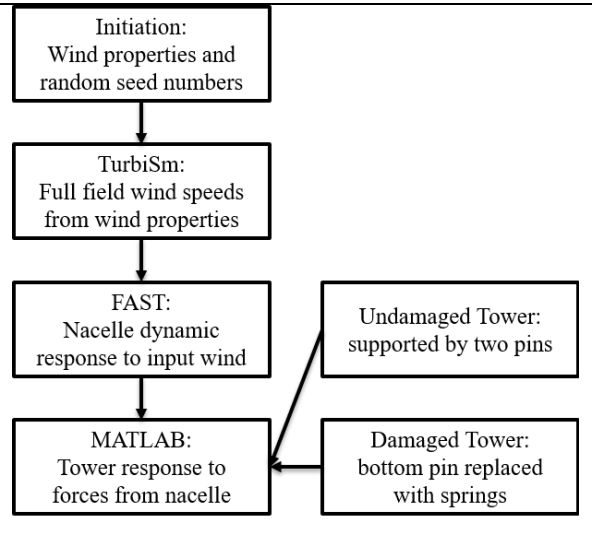


Figure 2-2. Description of the methodology of generating simulation datasets.

their significant values. In total, 12 channels of accelerations measured by six wireless nodes mounted at different locations along the tower are employed to describe the vibration of the wind turbine responded to the natural wind environment. The sampling frequency is 500 Hz, and the measured time duration is roughly 24 seconds. The measurements of all channels are transferred from each sensing node to a base station located closed to the turbine via 2.4 GHz IEEE 152. short-range radio. Both the sensing nodes and the base station are powered by a battery bank. The whole Martlet sensing platform is designed and deployed in June 2013. The experimental datasets were collected on September 16-18, 2013, and the main wind direction is south which make the tower instead of the bolt/spring lean against the wind flow excitation. Total of 325 undamaged datasets are collected approximately every 5 minutes because the 12 channels of measurements require time to completely transfer the data. After manually introducing the aforementioned damaged scenario, another 42 samples are collected as the damaged datasets, which make the entire dataset consist of 367 observations (325 undamaged, and 42 damaged samples).

While this dataset is a valuable experimental dataset for SHM, it also has two shortcomings: (a) The lack of appropriate information on EOCs for data normalization, because the EOCs data were provided by the meteorological tower with a lower rate (every 15 minutes) instead of directly measured by an embedded sensor. (b) the small size of the dataset (367 samples); a higher number of data samples would help establish more confidence in hypotheses developed.

These issues motivated the team to generate a simulated dataset for the LANL wind turbine. The numerical simulation could generate larger datasets and all parameters of interest (including EOCs) are fully known and available. In addition, the simulation has the capability of generating various damaged scenarios that are likely to happen but had not occurred during the three-day interval of experimentation in LANL.

### **2.3.2 Simulation Dataset**

To overcome the shortcomings in the experimental datasets, a mechanical model is built for the turbine using FAST (fatigue, aerodynamics, structures, and turbulence) (Jonkman, J. M.; Buhl Jr., 2005) software, a computer-aided engineering tool developed by national renewable energy lab (NREL) for simulating the coupled dynamic response of wind turbines. The FAST code is capable to simulate the coupled aerodynamic and structural dynamic model using controllable wind inflow data to analyze wind turbine response in the time domain. In addition, FAST can be linked to MATLAB enabling users to implement advanced controls and processing. The advantage of this model is that it enables the analysis of two- or three blade horizontal-axis wind turbines (HAWTs) within the virtual environment where all details of important parameters are available. The model is built by modifying a sample model for Test 17 in FAST version 7, which is also a small-scale

wind turbine similar to the Whisper 500 in LANL. Model updating is also implemented to align the model power curve as accurately as possible with that of the experimental turbine (Whisper 500) provided by the manufacturer via changing both generator properties and model parameters. To excite the turbine, TurbSim (Jonkman & Kilcher, 2012), a stochastic turbulent-wind simulator also developed by NREL, is employed to provide a full-field flow environment that contains coherent turbulence. It uses a statistical model to generate time series of wind speed vectors in the interested space to affect wind turbine aeroelastic response and loading. Numerous different wind speed and direction time histories are generated by changing fluid dynamic features such as average wind speed and parameters pertaining to the random phase realizations of the wind speed spectra. Wind fields generated by TurbSim were fed to FAST to compute the dynamic response of the turbine model.

Since the FAST is only capable of modeling cantilever wind turbine tower and the tower of LANL is supported by two pins, a mechanical model is developed in MATLAB to transfer the response of cantilever wind turbine to that of the two-pin tower for more accurate representation. The properties of the mechanical model in MATLAB is imported from the tower model built in FAST. Similar to the LANL experimental setup, the damage scenario is created by replacing the pin in the bottom to a spring. In addition, to minimize the potential effects of differences between these two models, model updating is also implemented to match their first modal frequencies. Considering the small motion of the tower, we assume that the dynamics of the tower does not affect the response of the nacelle to wind flow. Thus, the loads at the tower tip (nacelle) computed by FAST are fed to the MATLAB model to compute the response of the turbine tower supported by two pins. After the validation of the model created in MATLAB, the updated two-pin supported tower model is used in junction with FAST and TurbSim to generate the simulated dataset. A

variety of environmental conditions and key operational conditions such as rotor angular velocity and nacelle yaw angle are preset as input excitation. Structural response was computed using the combination of FAST and MATLAB models. The acceleration at the same locations as the sensors are recorded with a sampling frequency of 500 Hz (i.e., 0.002 s time interval as same as the experimental dataset).

We emphasize that numerical models are not dedicated to a flawless expression nor a mirror image of experimental turbine. And the major mismatch should be contributed to the main rotor bearing of the LANL turbine (Chipka, J. B. et. al., 2013). While the inherent error exists, there are two main purposes for developing a numerical mechanical model: (a) objective is to find certain features from structures response that correlate with EOCs and the other one is to validate the hypotheses of this research on a large and diverse enough dataset, which might be very difficult to acquire when using experimental setups; (b) the mechanical model can be treated as a separate structure from that of LANL since the hypotheses are not specific to LANL wind turbine, but rather generalizable to all structures. As for the first objective, the model is used only to inform which potential structural response features do have strong correlations with EOCs. The inaccuracies associated with the model are within acceptable limits for this purpose, especially since there was a prior belief that these features would actually correlate with EOCs. If a mechanical model was to be used for an application that required the accurate representation of the system, such as for control purposes or estimating the remaining life of the structure, a more accurate model would have been necessary.

Based on the mechanical model of LANL wind turbine, the relationships between the key EOCs and the undamaged structural response features are first studied in the simulations to pursue EOC sensitive parameters for assignments. Total of 1809 simulations are generated to simulate the

response of the undamaged tower for 80 seconds, with uniform distribution of wind speed and direction. Considering the initial disturbing caused by wind flow, the first 20 seconds are discarded to keep the remaining as stable responses to form the desired data set. According to the time duration measured in the experiments, each simulation is divided into two data samples with a 30-second duration resulting in a total of 3618 data samples.

After the study of exploring the EOC-sensitive features in the simulations, various wind speeds and directions are generated by the TurbSim based on the meteorological records from the TA-49 station to implement the damage detection framework. For the training stage, only the undamaged data samples are employed to train the three-tier framework. The total number of undamaged tower data samples for training is 1800 with a time duration of 30 seconds. For the testing stage, 1200 undamaged and 1200 damaged data samples are used to validate the proposed strategy. We emphasize that the 1200 damaged data samples are generated by setting three stiffness values of the springs in the tower as weak, moderate and strong to simulate different supporting from the pins in the tower of LANL wind turbine. The relationships of EOCs and structural response features from the undamaged/damaged datasets are also compared to evaluate the dependency of EOC sensitive features on the structural damages.

## **2.4 Results and Discussions**

To extract the damage-sensitive features for clustering, large data sets should be normalized according to the EOCs. Thus, to obtain a better normalization, the most important procedure is to pursue EOCSFs which are ideally only related to EOCs and not depending on structural damaged states.

### 2.4.1 EOC-Sensitive Features

There are two major EOCs influence the response of the wind turbine tower: rotor angular velocity and nacelle yaw angle which directly related to the wind speeds and wind directions according to the previous research (Bahrami, O. et. al., 2017). Thus, peak frequencies resulted from the rotor angular velocity and the acceleration energy ratio in the x- and y- directions affected by the nacelle yaw angle are two major EOCSFs.

We emphasize that the EOCs are not limited by only two, more various EOCs could be considered for the same operating structure. The selections of influential EOCs from the infinitely combinations are based on the engineering judgement for the purposes of SHM. In this study, for this small-scale turbine in LANL that lack sophisticated control mechanisms used in utility grade wind turbines, rotor angular velocity strongly correlates with wind speed and nacelle yaw angle strongly correlates with wind direction. These operational conditions do represent the environmental conditions that influence the turbine as well.

#### 2.4.1.1 Peak Frequencies in Spectrograms

Short-time Fourier transform (STFT) is used to build the spectrogram of acceleration in  $x$ -direction reveal the relationship between the rotor angular velocity and the transient peak frequencies. A 1.024 s window (512 data points with a sample rate of 500 Hz) with a time step of 0.024 s is employed to capture the variations in the frequency content of the signals as time evolves. Since the transient peaks in all twelve acceleration channels are similar, only the spectrogram of CH. 11 data (Sensor 6,  $x$ -direction acceleration response at 10.23 m) is shown in Figure 2-3 for the purpose of demonstration. Two general sets of frequency peaks were observed, one constant in time and the other time varying. The steady peaks are located at lower frequencies attributed to structural modal frequencies. As we can see in Figure 2-3 (a), the first peak frequency and the second peak

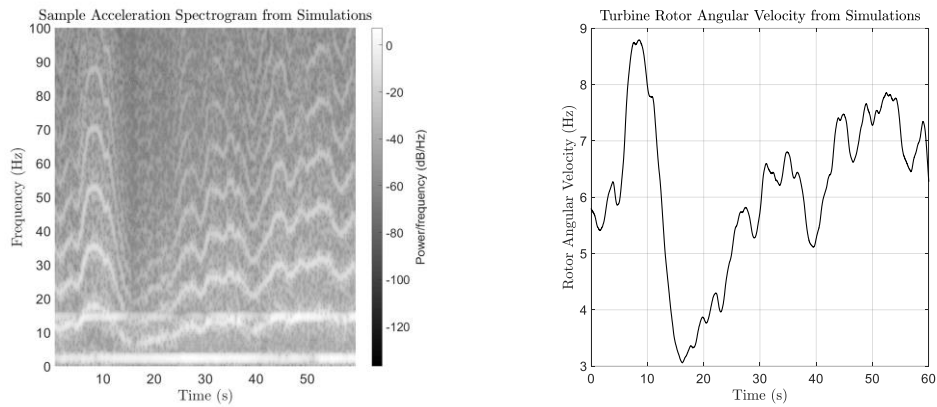


Figure 2-3. (a) Sample acceleration spectrogram. (b) Rotor angular velocity for the same dataset.

frequency around 2 Hz and 14 Hz, respectively, is independent to the rotor angular velocity which represent the natural modal vibration of the wind turbine tower. And the time varying peaks are higher frequencies related to the rotor angular velocity as shown in Figure 2-3 (b). The lowest three EOC-sensitive transient peak frequencies (second, third, and fourth peak frequency of transient peaks) are extracted from the spectrogram and averaged for the 30-second long acceleration records as shown in Figure 2-4. These two sets of frequency peaks can also be observed in the acceleration spectrogram of LANL wind turbine in the previous researches(Chipka, J. B. et al, 2013). Thus, to create the EOCSF corresponding to rotor angular velocity, only the lowest average EOC-sensitive peak frequency is calculated for the EOC-normalization. In addition, for the experimental data sets, the EOCSF is simply calculated by the highest amplitude in the power spectral density (Fourier transform) of the acceleration records.

#### 2.4.1.2 Signal Energy Ratio

As we mentioned, the nacelle orientation (nacelle yaw angle) is another important EOC which directly related to the wind direction. Considering the wind direction can cause different vibrations



in two measured directions, the natural logarithm of the ratio of acceleration energy in  $x$ - and  $y$ -directions is proposed to build an EOCSF to describe the nacelle orientation. The signal energy ratios in all six measurement locations was averaged over the time span to compute the final energy ratio ( $ER$  as defined in Equation 2-19):

$$ER = \ln\left(\frac{1}{6} \sum_{i=1}^6 \frac{E_{x_i}}{E_{y_i}}\right) \quad \text{Eq. 2-19}$$

$$E_{x_i} = \sum |x_i[t]|^2, E_{y_i} = \sum |y_i[t]|^2 \quad \text{Eq. 2-20}$$

where  $E_{x_i}$  and  $E_{y_i}$  represent signal energy of the acceleration in  $x$ - and  $y$ -direction at the  $i^{\text{th}}$  measurement location ( $x_i[t]$ , and  $y_i[t]$ ) along the tower respectively (Poularikas & Ramadan, 1998). Taking the average amongst all 6 measurement locations can reduce the variance to benefit the GPR-mean of the data points. Figure 2-5 shows the relationship between  $ER$  and average yaw angle for the simulation dataset.

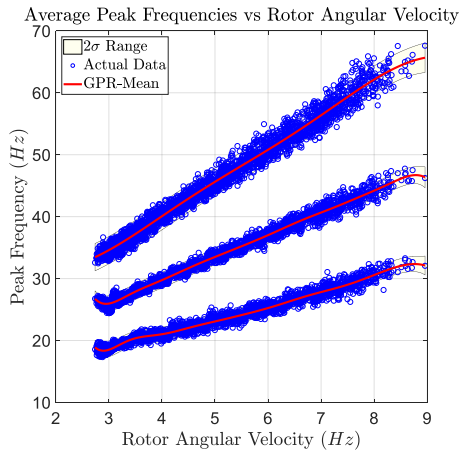


Figure 2-4. Average peak frequencies and rotor angular velocity.

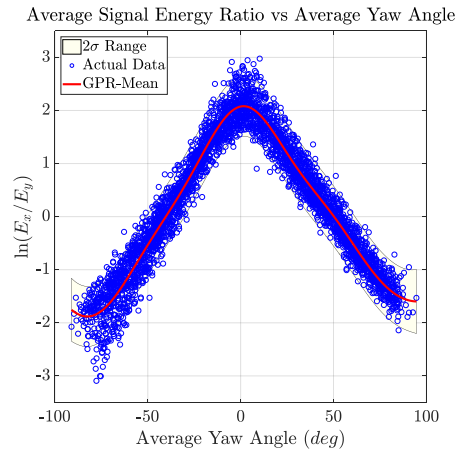


Figure 2-5. Average signal energy ratio an average yaw angle.

### 2.4.1.3 Features' Sensitivity to Structural Damage

Ideally, the EOCSFs are insensitive to structural damage for EOCs data normalization in damage detection framework. If structural damage varies EOCSFs, then it also affects the EOCs data

normalization process that uses EOCSFs as the basis for data normalization. This could result in assignment of dataset to a group with different EOCs and potentially lead to false classification. To evaluate the sensitivity of the proposed EOCSFs to structural damage in the tower, the synthetic damaged datasets including three damaged levels were used. The aforementioned two features (peak frequencies and  $ER$ ) are extracted from the damaged datasets. Meanwhile, GPR is employed to perform their mean values and  $2\sigma$  range in Figure 2-6 compared the previous undamaged case. The GPR calculations of EOCSFs from damaged dataset is shown in red, and undamaged ones are shown in green (only colorful online). As it can be seen, the damaged mean value has good agreement with the undamaged ones, and even the  $2\sigma$  ranges match each other as well. That indicates the proposed EOCSFs (peak frequencies and  $ER$ ) remain insensitive to damage scenario, which can be used to reliably present the wind speed and wind direction.

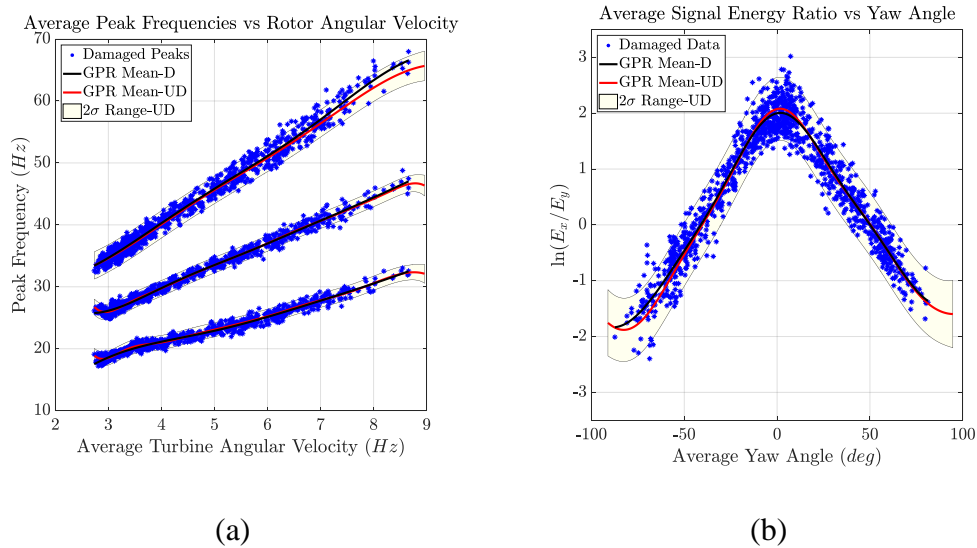


Figure 2-6. Damage EOCSFs against the corresponding EOCs. (a) Average peak frequencies and (b) average signal energy ratio.

It is emphasized that in this study we only focus on the damage of the wind turbine tower instead of the damage in the blade or nacelle, etc., which might introduce more complex damaged scenarios causing instability of the EOCSFs. In the case of assuming that the damage only occurred

on the tower, the results above have validated the proposed EOCSFs are insensitive to various damage levels in the wind turbine tower.

## **2.4.2 SHM Framework Application**

### *2.4.2.1 LANL Dataset*

The three-tier modular framework was implemented on the LANL dataset for damage detection. For EOCs data normalization, two sets of features were available. The first one was the wind statistics over the 15-minute intervals from the meteorological tower at LANL, which is labeled as long-averaged EOCs. The second EOCSF is labeled as structural features introduced in the previous sub-section which are extracted from structural response data with an approximately 30-second length.

217 undamaged datasets were randomly selected for training, and the remaining 108 undamaged datasets and 42 damaged ones were used in the testing stage. In the EOCs data normalization stage, five to seven clusters were used for both the hard and soft assignment cases. When using long average EOCs, clustering was performed on a two-dimensional feature space consisting of averaged wind speed and averaged wind direction. When using structural features, the feature space for clustering consisted of the two previously introduced EOCSFs.

In this chapter, receiver-operating-characteristics curves (ROC curves) are employed to evaluate the performance of the framework. ROC curves are plots of true positive (TP) rate versus false positive (FP) rate whose significance level is varied from 0 to 1. TP rate is the ratio of correctly classified damaged observations to the total number of damaged observations in the testing phase and the FP rate is the ratio of falsely classified undamaged observations to the total number of undamaged observation in the training stage. It is obvious that these rates are between

0 and 1 indicating that the ROC curve would be contained within a unit square. In the extreme case when  $\alpha=0$ , all observations are labeled as undamaged which resulting in both TP and FP rates to be 0. Conversely, when  $\alpha=1$  all observations are labeled as damaged resulting in both TP and FP rates to be 1. Hence, the ROC curve begins from (0,0) and ends at (1,1). An ideal ROC curve would pass the point (0,1) (upper left corner of the unit square) which corresponds to a scenario where all observations are correctly classified an ROC curve that passes through the diagonal of the unit square corresponds to a random guess (Akobeng, 2007). The area under the curve (AUC) of ROC curves is a scalar that can be used to compare the two ROC curves (Akobeng, 2007): the higher the AUC, the better the performance of the algorithm. ROC curve essentially represents how well undamaged and damaged test statistics are separated from one another: the better the separation, the more the ROC curve is tending toward (0,1) and the higher AUC.

Table 2-1. AUC values for DSF-1 for different clustering approaches and different number of clusters for LANL dataset.

No. Clusters	K-means	K-means	GMM	GMM
	Long Average	Features	Long Average	Features
5	0.63	0.88	0.70	0.88
6	0.80	0.88	0.81	0.86
7	0.84	0.88	0.85	0.93

Results of implementing the framework is shown in Figure 2-8 in the form of ROC-curves for the two DSFs. Table 2-1 and Table 2-2 show the AUC values for different number of clusters used.

The results show significant improvement when using EOCSFs for clustering compared to long average EOCs. In addition, soft assignment (GMM) improves the AUC values when

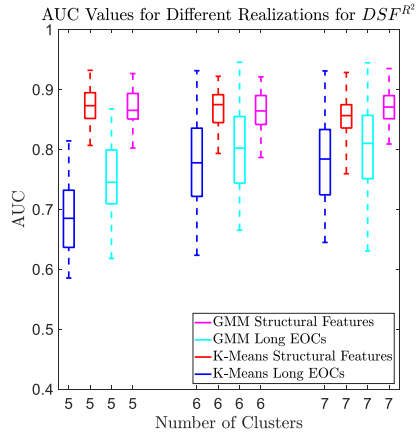
compared to the hard assignment (*K*-means) approach when the same set of EOCs are used for data normalization.

Table 2-2. AUC values for DSF 2 for different clustering approaches and different number of clusters for LANL dataset.

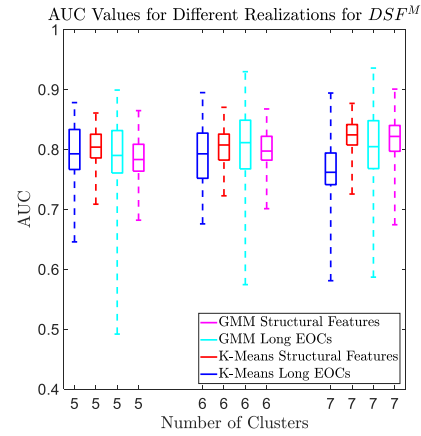
No. Clusters	K-means	K-means	GMM	GMM
	Long Average	Features	Long Average	Features
5	0.72	0.79	0.77	0.84
6	0.81	0.85	0.82	0.85
7	0.81	0.87	0.83	0.88

The ROC curves and AUC values presented previously are the results of one selection of training dataset. To evaluate the effects of using other training subsets, 49 more selections of training datasets were used for the undamaged datasets. For each selection of training datasets, the remainders of the undamaged observations and all the damaged observations were used in the testing stage. Overall, for the total 50 selections, 25 of the selections had 217 undamaged datasets in the training stage, and the other half selections used 200 undamaged datasets for training. The SHM framework was implemented and the AUCs were recorded for each selection. Figure 2-7 shows the bar plots of AUC values for the 50 selections used as a function of number of clusters in the data normalization stage.

As it can be seen, when using hard assignment, structural features yield higher AUC values compared to long average EOCs. Additionally, the variation in AUC values is less for structural features indicating a more reliable data normalization stage. Hence, features prove to be more promising for data normalization compared to the available long averages from meteorological tower.



(a)



(b)

Figure 2-7. Statistics of AUC values for a) DSF-1 and b) DSF-2 for different realizations of the experimental data.

When comparing soft assignment results with those of hard assignment, a general trend is that for long-averaged EOCs, soft assignment results have higher median AUCs than hard assignment. This is true for all DSF-cluster numbers except five clusters for  $DSF^M$ . For both DSFs, the median value for AUCs is higher for soft assignment for seven clusters and the rest have higher values for hard assignment. It is tough to evaluate the significance of the introduced soft assignment methodology on LANL dataset due to its small size. In the next subsection, the analysis would be performed on the larger and more diverse (in terms of variations in EOCs) simulation dataset.

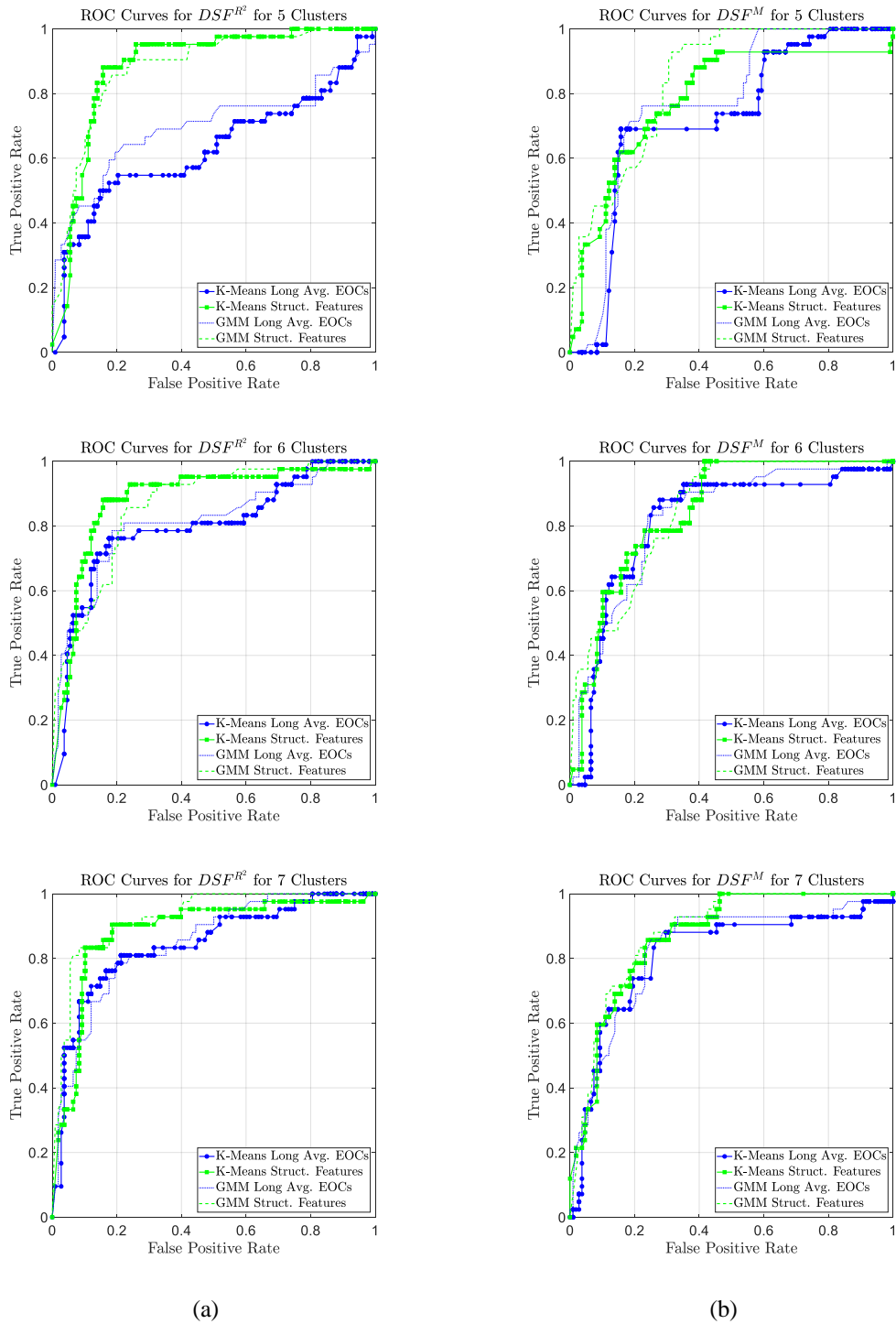


Figure 2-8. ROC curves for experimental dataset for a) DSF-1 and b) DSF-2 for 5, 6 and 7 EOC clusters.

#### 2.4.2.2 *Simulation Dataset*

Recall that the simulations were carried out for 200-seconds whose first 20 seconds was ignored to remove the influence of initial conditions imposed on simulations and the remaining 180 seconds was divided into six individual observations each with a duration of 30 seconds. For the FAST simulation datasets, three sets of features were used for data normalization of EOCs: (1) the averaged yaw angle and rotor angular velocity over the 30-second interval of simulation termed actual average EOCs, (2) the EOCSFs introduced previously, which were again named structural response features, (3) averaged yaw angle and rotor angular velocity over the entire 200 seconds of simulations named as long average EOCs for its corresponding 30-second observations. The third one was used to imitate the situation in LANL whose EOCs can be only averaged over longer time intervals instead of using environmental conditions (those corresponding to wind statistics) and the averages of operational conditions. As it was aforementioned, wind statistics and operational conditions were highly correlated for simulations. Additionally, the introduced set of operational conditions is more influential on structural response compared to environmental conditions. Hence, the long-averaged EOCs introduced for simulation datasets are more likely to outperform those introduced for LANL datasets. The wind statistics generated by TurbSim were stationary, hence, the difference between the long average and actual average EOCs was not enormous.

Three different levels of structural damage introduced previously were used to evaluate how the extent of damage influences the performance of the features for EOCs data normalization. Since the dataset was larger than the LANL dataset, higher number of clusters were required in the data normalization stage. Thus, 3-10, 13-16 and 19-21 clusters were used in the data-normalization stage as show in Tables 2-3 to 2-8.



Tables 3 to 8 show the AUC values for the two DSFs used for different numbers of clusters used and the three different damage scenarios. Figure 2-9 to 2-12 show the ROC-curves for 8, 13, 16 and 21 clusters for different damage levels. Only four ROC-curves per DSF are shown for each damage level due to the similarity of the ROC-curves to one another. Inside each plot, a 0.4 by 0.4 square of the original plot is magnified to make the comparison of ROC curves easier in that region.

When hard assignment is used for data normalization, actual features yield higher AUCs than structural response features and structural response features yield higher AUCs than long averages. This is consistent with what was expected and shown in the previous subsection.

For all three sets of features used for EOCs data normalization, the soft assignment approach outperforms hard assignment. This is true for all number of clusters used for  $DSF^M$  and for most number of clusters for  $DSF^{R^2}$  with the exception of few clusters for structural response features. For some number of clusters, especially for  $DSF^{R^2}$ , using soft assignment with long averages increases AUCs as much as using hard assignment with actual average EOCs highlighting the value of soft assignment approach.

Table 2-3. AUC values for DSF-1 for different clustering approaches and different number of clusters for simulation dataset with weak damage.

No. Clusters	K-means Actual Averages	K-means Features	K-means Long Averages	GMM Actual Average	GMM Features	GMM Long Averages
3	0.71	0.69	0.66	0.71	0.67	0.66
4	0.68	0.67	0.65	0.68	0.70	0.70
5	0.71	0.69	0.68	0.72	0.70	0.70
6	0.70	0.67	0.70	0.72	0.72	0.69
7	0.73	0.68	0.69	0.75	0.71	0.70
8	0.72	0.70	0.68	0.72	0.69	0.72
9	0.72	0.68	0.70	0.73	0.70	0.73
10	0.73	0.68	0.69	0.76	0.71	0.71
13	0.73	0.70	0.71	0.74	0.70	0.70
14	0.73	0.70	0.69	0.75	0.70	0.70
15	0.73	0.70	0.70	0.73	0.70	0.70
16	0.73	0.69	0.70	0.73	0.71	0.72
19	0.73	0.70	0.70	0.74	0.70	0.73
20	0.74	0.71	0.70	0.74	0.70	0.73
21	0.73	0.72	0.71	0.74	0.72	0.72

Table 2-4. AUC values for DSF-1 for different clustering approaches and different number of clusters for simulation dataset with moderate damage.

No. Clusters	K-means Actual Averages	K-means Features	K-means Long Averages	GMM Actual Average	GMM Features	GMM Long Averages
3	0.83	0.79	0.78	0.85	0.81	0.79
4	0.80	0.78	0.80	0.80	0.82	0.81
5	0.83	0.81	0.79	0.82	0.84	0.82
6	0.81	0.82	0.81	0.84	0.83	0.82
7	0.84	0.81	0.80	0.86	0.83	0.82
8	0.83	0.83	0.80	0.85	0.84	0.84
9	0.83	0.83	0.81	0.85	0.83	0.85
10	0.84	0.80	0.80	0.86	0.84	0.84
13	0.85	0.81	0.81	0.87	0.82	0.83
14	0.84	0.82	0.81	0.85	0.81	0.82
15	0.84	0.82	0.81	0.85	0.82	0.82
16	0.84	0.81	0.81	0.84	0.82	0.83
19	0.85	0.82	0.82	0.86	0.80	0.85
20	0.85	0.83	0.82	0.85	0.82	0.86
21	0.85	0.83	0.83	0.86	0.81	0.86

Table 2-5. AUC values for DSF-1 for different clustering approaches and different number of clusters for simulation dataset with strong damage.

No. Clusters	K-means Actual Averages	K-means Features	K-means Long Averages	GMM Actual Average	GMM Features	GMM Long Averages
3	0.87	0.84	0.83	0.89	0.85	0.83
4	0.86	0.84	0.86	0.83	0.86	0.85
5	0.88	0.86	0.84	0.88	0.87	0.87
6	0.86	0.87	0.86	0.88	0.88	0.85
7	0.89	0.87	0.84	0.90	0.88	0.87
8	0.88	0.88	0.86	0.88	0.88	0.87
9	0.88	0.88	0.86	0.89	0.87	0.88
10	0.89	0.86	0.85	0.90	0.88	0.87
13	0.90	0.87	0.87	0.90	0.87	0.87
14	0.88	0.87	0.85	0.89	0.86	0.86
15	0.88	0.87	0.86	0.90	0.86	0.87
16	0.88	0.87	0.86	0.90	0.87	0.87
19	0.89	0.87	0.87	0.91	0.87	0.89
20	0.89	0.87	0.87	0.90	0.87	0.87
21	0.89	0.88	0.87	0.90	0.87	0.89

Table 2-6. AUC values for DSF-2 for different clustering approaches and different number of clusters for simulation dataset with weak damage.

No. Clusters	K-means Long Averages	K-means Features	K-means Actual Averages	GMM Long Average	GMM Features	GMM Actual Averages
3	0.93	0.92	0.86	0.99	0.99	0.92
4	0.94	0.88	0.87	0.98	0.96	0.91
5	0.94	0.88	0.88	0.98	0.97	0.94
6	0.95	0.94	0.90	0.98	0.97	0.94
7	0.96	0.94	0.93	0.96	0.97	0.94
8	0.97	0.93	0.93	0.96	0.96	0.94
9	0.97	0.94	0.92	0.98	0.96	0.95
10	0.97	0.96	0.93	0.99	0.98	0.94
13	0.98	0.96	0.94	0.97	0.97	0.95
14	0.97	0.96	0.92	0.96	0.97	0.95
15	0.97	0.95	0.92	0.97	0.97	0.94
16	0.98	0.92	0.93	0.98	0.96	0.96
19	0.97	0.96	0.93	0.99	0.98	0.95
20	0.98	0.96	0.94	0.99	0.98	0.96
21	0.99	0.98	0.94	0.97	0.98	0.96

Table 2-7. AUC values for DSF-2 for different clustering approaches and different number of clusters for simulation dataset with moderate damage.

No. Clusters	K-means Long Averages	K-means Features	K-means Actual Averages	GMM Long Average	GMM Features	GMM Actual Averages
3	0.96	0.95	0.91	1.00	1.00	0.97
4	0.97	0.93	0.92	0.99	0.98	0.96
5	0.97	0.96	0.94	0.99	0.99	0.98
6	0.98	0.95	0.95	0.99	0.99	0.97
7	0.98	0.96	0.96	0.98	0.99	0.97
8	0.98	0.97	0.95	0.98	0.99	0.97
9	0.99	0.96	0.96	0.99	0.99	0.97
10	0.98	0.98	0.97	1.00	0.99	0.97
13	0.98	0.98	0.97	0.98	0.99	0.98
14	0.98	0.98	0.96	0.98	0.98	0.98
15	0.99	0.98	0.96	0.99	0.99	0.98
16	0.99	0.96	0.97	0.98	0.98	0.98
19	0.98	0.98	0.98	1.00	0.99	0.99
20	0.99	0.98	0.97	0.99	0.99	0.99
21	0.99	0.98	0.98	0.99	0.99	0.99

Table 2-8. Table 8. AUC values for DSF-2 for different clustering approaches and different number of clusters for simulation dataset with strong damage.

No. Clusters	K-means	K-means Features	K-means	GMM	GMM Features	GMM
	Long Averages		Actual Averages	Long Average		Actual Averages
3	0.97	0.96	0.93	1.00	1.00	0.98
4	0.98	0.95	0.93	0.99	0.99	0.98
5	0.98	0.97	0.95	0.99	0.99	0.99
6	0.98	0.97	0.96	0.99	0.99	0.98
7	0.99	0.97	0.97	0.99	1.00	0.99
8	0.99	0.98	0.96	0.98	1.00	0.98
9	0.99	0.97	0.97	0.99	0.99	0.98
10	0.99	0.98	0.97	1.00	1.00	0.98
13	0.99	0.98	0.98	0.99	0.99	0.99
14	0.99	0.98	0.97	0.99	0.99	0.99
15	0.99	0.98	0.97	1.00	1.00	0.98
16	0.99	0.97	0.98	1.00	0.99	0.99
19	0.99	0.99	0.98	1.00	1.00	1.00
20	1.00	0.99	0.98	0.99	1.00	0.99
21	1.00	0.99	0.98	1.00	0.99	0.99

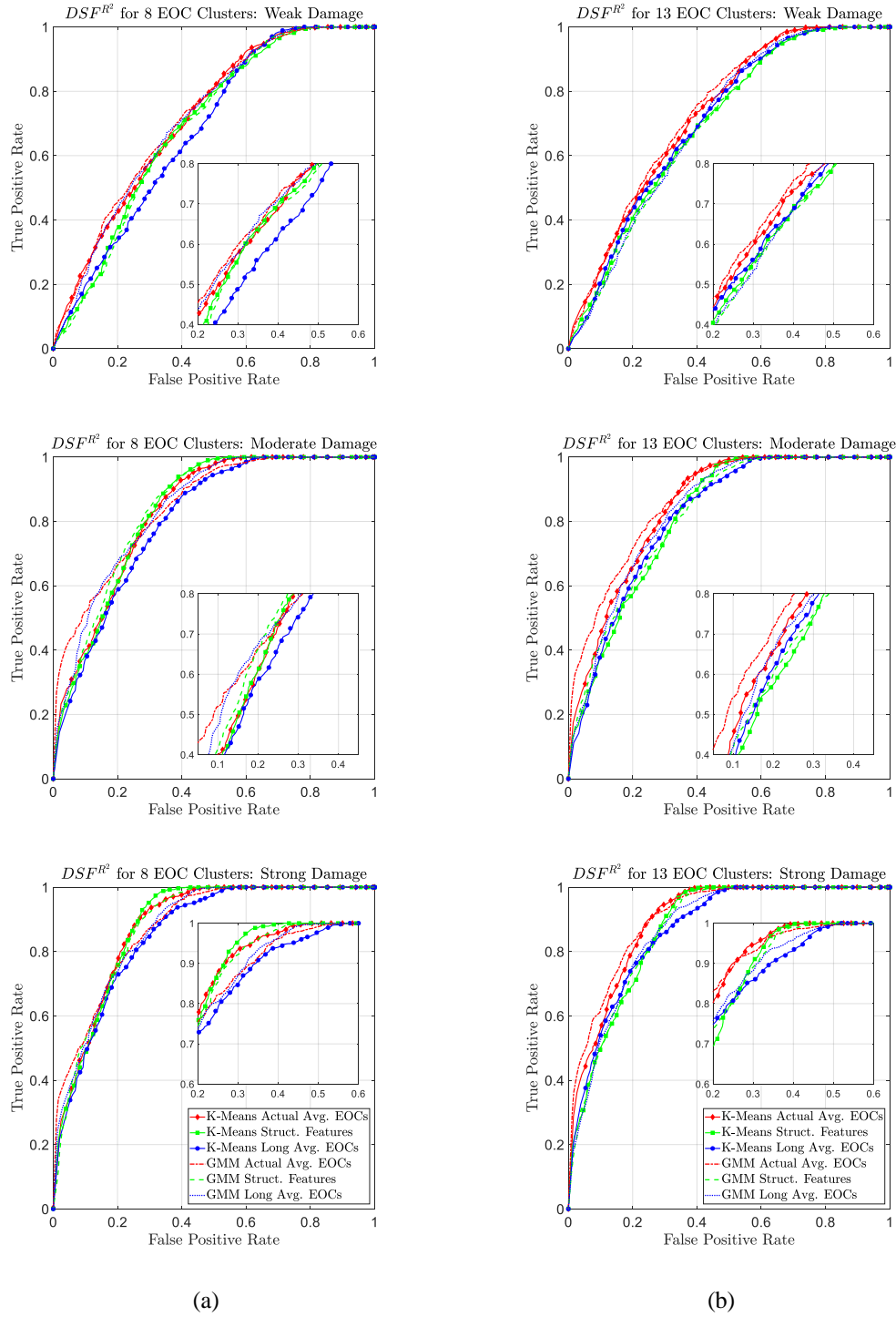


Figure 2-9. ROC curves for simulation dataset for DSF-1 for a) 8 and b) 13 EOC clusters for weak, moderate and strong levels of structural damage.



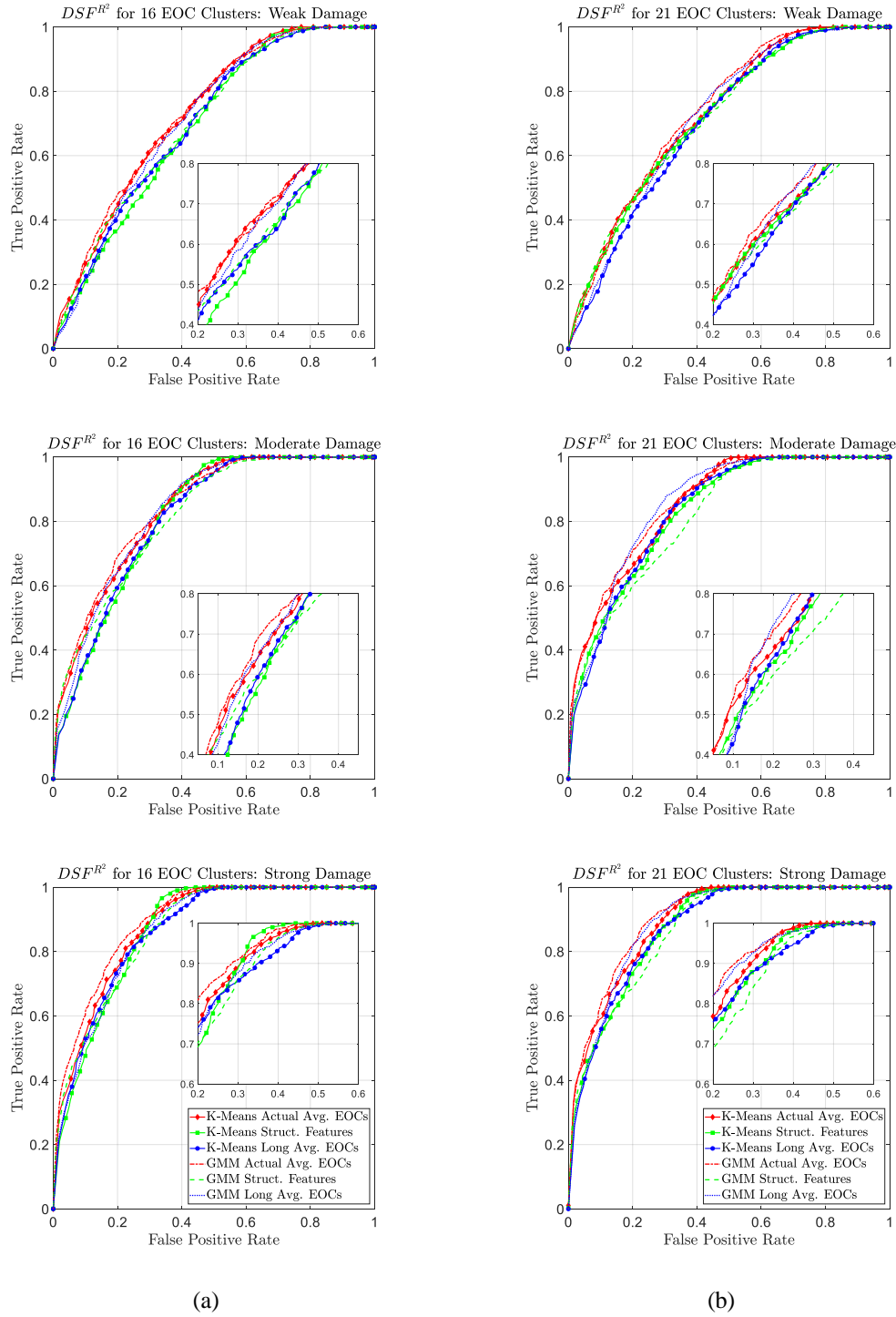


Figure 2-10. ROC curves for simulation dataset for DSF-1 for a) 16 and b) 21 EOC clusters for weak, moderate and strong levels of structural damage.

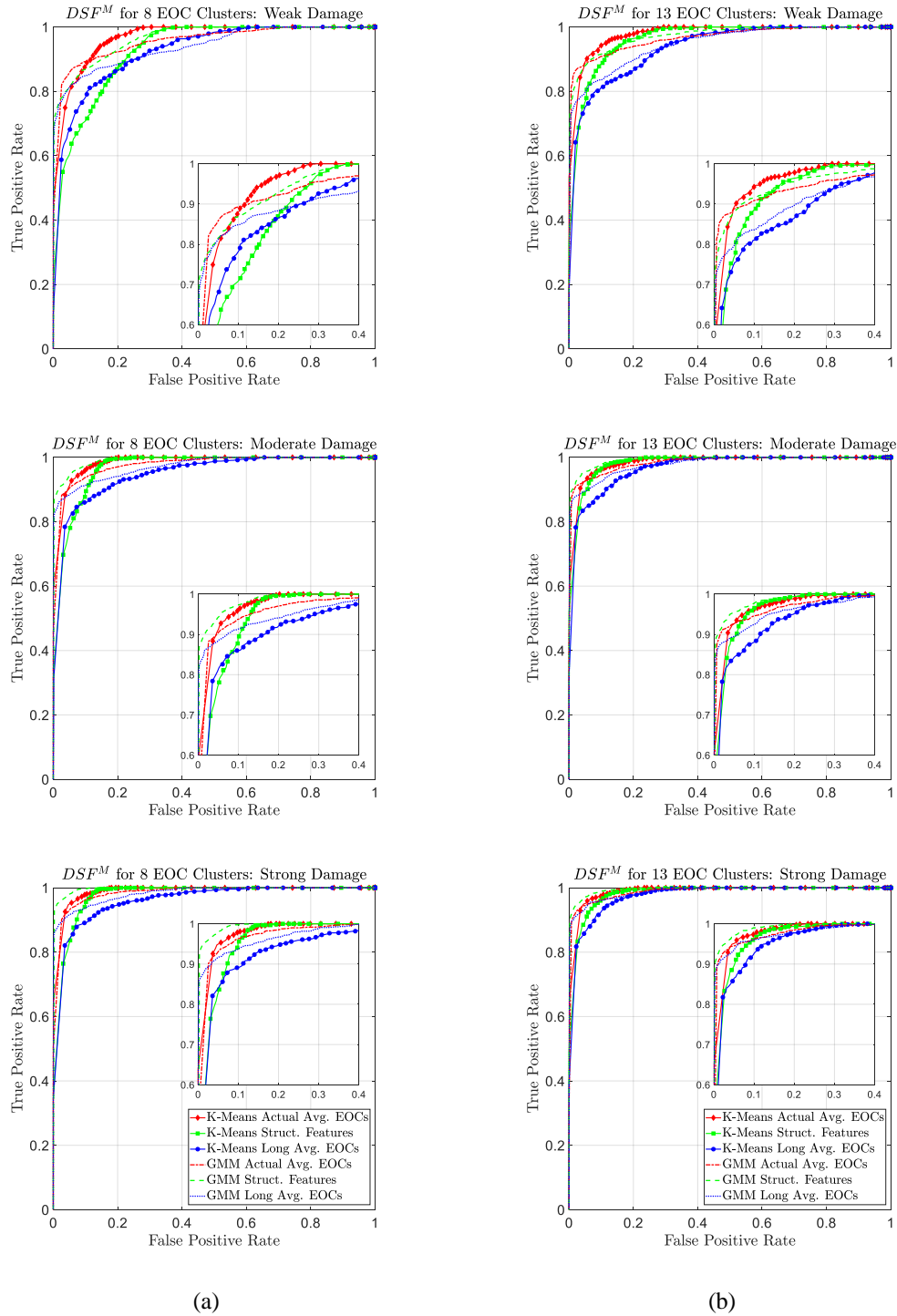
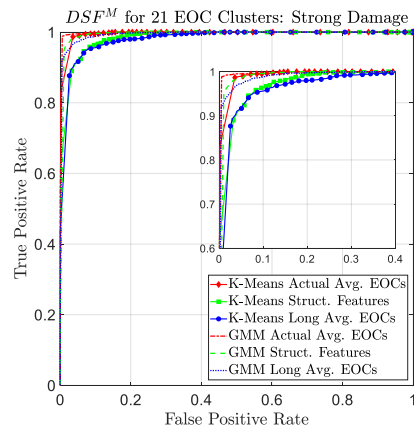
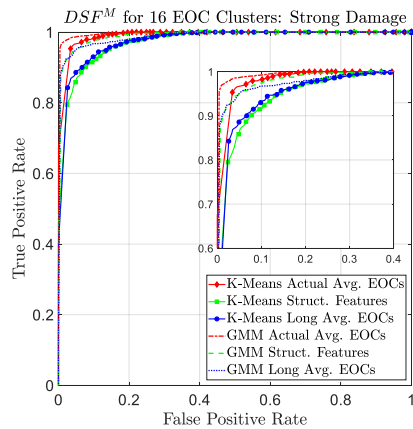
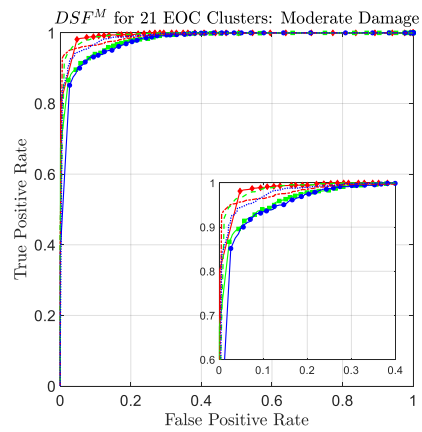
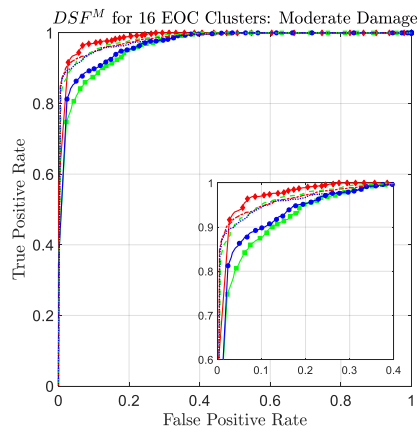
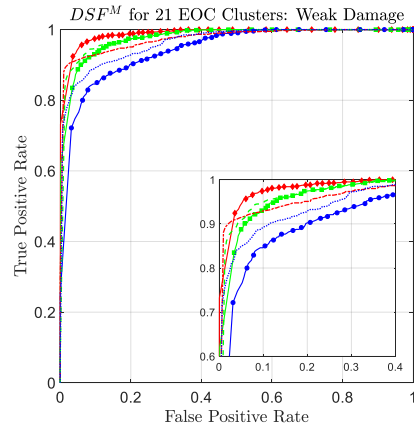
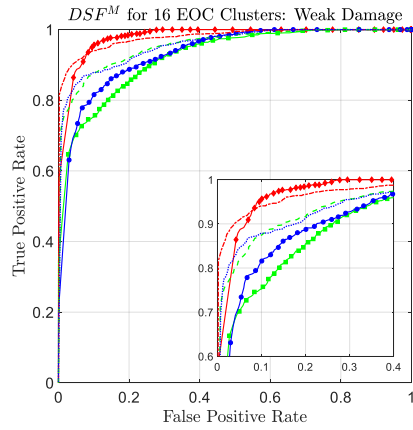


Figure 2-11. ROC curves for simulation dataset for DSF-2 for a) 8 and b) 13 EOC clusters for weak, moderate and strong levels of structural damage.



(a)

(b)

Figure 2-12. ROC curves for simulation dataset for DSF-2 for a) 16 and b) 21 EOC clusters for weak, moderate and strong levels of structural damage.

## 2.5 Conclusion

In this chapter, two approaches for increasing the efficiency of EOCs data normalization for SHM were introduced. First, for situations where accurate measures of EOCs is not available, a methodology to extract EOCSFs from structural response data was defined and organized. Following that methodology two EOCSFs, namely average rotor angular velocity and natural log of signal energy ratio were shown to correlate with two key influential EOCs, rotor angular velocity and nacelle yaw angle for the tower of a small scale wind turbine as the test structure. The insensitivity of EOCSFs to structural damage in the component of consideration was evaluated and the robustness of EOCSFs to structural damage was proved. Next, the features were used for EOCs data normalization and showed to increase the overall performance of damage detection for LANL dataset compared to the baseline average wind statistics available onsite.

Second, a new approach for EOCs data normalization based on soft assignment was introduced to deal with the two main issues associated with clustering in EOCs data normalization. The methodology was proposed based on an alternative view of hypothesis testing. Gaussian Mixture Models were used to assign probabilities of dataset belonging to EOCs clusters. This approach was implemented on both the experimental and simulation datasets using different measures of EOCs for clustering and was shown to either over-perform or perform equally as good as the previously used hard assignment approach. The key advantage of the soft assignment approach lies in its capability to include the effects of EOCs variations in the probability of damage of the structure which can later be used for reliability analysis.

## **Chapter 3.**

### **Hidden Markov Models for Sequential Damage Detection of Bridges**

#### **3.1 Introduction**

Data-driven damage detection frameworks have been established as a promising tool for structural health monitoring (SHM). First, advances in sensing technologies have reduced the cost of data collection. Consequently, valuable datasets have been collected over the past few years on different types of structures such as bridges (Jang et al., 2010; M. Kurata et al., 2011), pipelines (Inaudi & Glisic, 2010), wind turbines (Hackell et al., 2016), and retaining walls (Admassu et al., 2019) just to mention a few. Second, the availability of machine learning (ML) algorithms now enable the extraction of meaningful information from large sets of collected structural response data.

Data-driven SHM methods do not rely on baseline physical models of the structure which can be difficult to acquire for complex structural systems. On the contrary, data-driven SHM frameworks only require observed data to be able to build decision boundaries (i.e.. known as training in the field of ML). An important practical capability of unsupervised SHM method is their ability to detect significant changes in the behavior of the structure without prior knowledge of potential damage scenarios.

Hence, only data from the normal (undamaged) system is used during training of the model.

The problem of detecting changes in the observations when only data from the normal (i.e., undamaged) class is available during the training is referred to outlier detection or anomaly detection (AD) (Farrar & Worden, 2012).

A common assumption in the majority of AD algorithms, is that observations are independent and identically distributed (i.i.d.). This assumption has enabled the application of numerous novel unsupervised ML techniques to SHM. However, it is well known that structural states are not independent; rather, sequential observations are dependent due to the one way transition from being undamaged to damaged. In this work we seek to relax the i.i.d. assumption by taking into account the sequential nature of the structural response during unsupervised AD classifier training. In particular, we use the hidden Markov model (HMM) to build a sequential classifier. HMM-based approaches have been proposed previously for SHM (M., Mollineax; R., 2015; Zhou et al., 2007). The proposed methodology in this research differs from those previously mentioned in that the proposed HMM-based framework does not require observations from the damaged state during training.

The proposed method is implemented on an experimental dataset from the widely used Z-24 bridge dataset (Peeters & De Roeck, 2001). The remainder of this chapter is organized as follows. The HMM-based AD method is first introduced. Then the Z-24 bridge dataset is described. In the next section, a detailed description of the implementation of the HMM-based AD method and baseline i.i.d. methods are presented followed by a summary of the results of the methods applied to the Z-24 data. The chapter concludes with a summary of the key findings.

## 3.2 Theory

### 3.2.1 Hidden Markov Models

A brief description of the HMM is first introduced. For a more detailed description, interested readers are referred to (Bishop, 2006). HMM is a well-known method for modelling sequential observations. When using HMM, observations are assumed to be emitted from a certain set of hidden states and the hidden states possess Markov properties (i.e. given the current state, the future state is independent of previous states).

Let  $x_t$  and  $z_t$  denote the observation and the hidden state, respectively, at a certain time instance  $t$  where  $0 \leq t \leq T$ . Then the Markov property can be expressed as follows:

$$p(z_{t+1}|z_t, z_{t-1}, \dots, z_0) = p(z_{t+1}|z_t) \quad \text{Eq. 3-1}$$

where  $p$  is the transition probabilities.

The schematic of an HMM is shown in Figure 3-1. An HMM is described by three properties. First is the state transition probabilities. Suppose there are a total of  $N$  hidden states (for simplicity, denote each unique state with numbers 1 to  $N$ ). At each time instance, the hidden state can be only one of the  $N$  hidden states. The dynamics of the model is governed by a matrix whose terms are transition probabilities:  $\mathbf{A}(i, j) = p(z_{t+1} = j | z_t = i)$  for  $i, j = 1, 2, \dots, N$ . This matrix is referred to as the state transition matrix.

The second important property of the HMM, is the distribution of the observations given the states, also known as emission probabilities. In this work, we assume that the emission probabilities are Gaussian:

$$p(x_t | z_t = i) = \mathcal{N}(x_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad \text{Eq. 3-2}$$

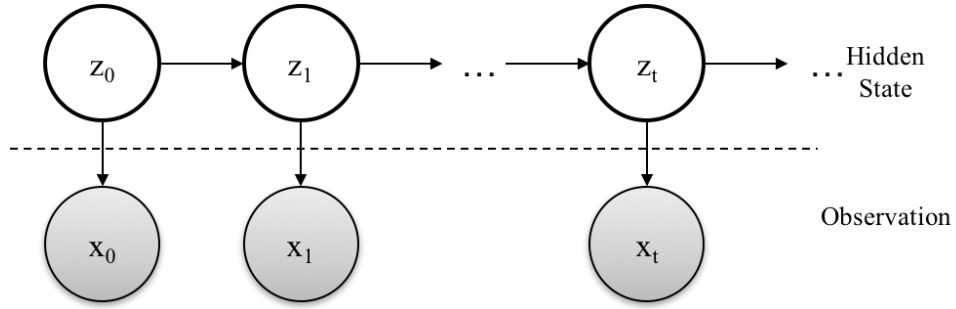


Figure 3-0-1. Graphical representation of the HMM. The observed variable (gray circles) are emitted from states that are not observable (hidden). The hidden states possess Markov property.

where  $\boldsymbol{\mu}_i$  and  $\boldsymbol{\Sigma}_i$  are the mean vectors and covariance matrices of the Gaussian distribution describing the emission probability of the  $i$ -th state respectively. Let  $\boldsymbol{\phi}$  denote the set containing all pairs of means and covariances.

The final property of the HMM is the allocation of the initial state probabilities  $\pi_i = p(z_0 = i)$ . A vector containing all of the initial state probabilities can be denoted as  $\boldsymbol{\pi}$ . The set of parameters  $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi})$  fundamentally defines the HMM.

In order to find the best set of HMM parameters,  $\boldsymbol{\theta}$ , for an observed dataset  $\mathbf{X} = \{x_0, x_1, \dots, x_T\}$  (given a set number of hidden states  $N$ ) the HMM model is trained by maximizing the likelihood of the observed dataset:

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} (p(\mathbf{X}|\boldsymbol{\theta}, N)) \quad \text{Eq. 3-3}$$

This can be done efficiently using the Baum-Welch algorithm also known as the forward-backward-algorithm (Baum et al., 1972). In practice, the logarithm of the probabilities are used when working with HMM. This is due to the fact that as  $T$  grows large, the probability  $p(\mathbf{X})$  becomes small.



### 3.2.2 Damage Detection Using HMM

Once the optimal model parameters  $\theta_{ML}$  have been determined, for any sequence of observations, the forward algorithm can be used to compute the probability of observing that sequence.

Let  $\mathbf{X}_l = \{x_{S+1}, \dots, x_{S+l}\}$  be a sequence of observations of length  $l$  for some  $S \geq T$ . We use  $p(\mathbf{X}_l|\theta_{ML})$  to make a decision on the state of the structure. If this probability is significantly lower than that observed during training, then it is implied that this sequence is coming from a damaged (i.e., not normal) structure.

In this manner, a sequence of observations of size  $l$  are used to compare the consistency of this sequence with those of the previous observations made during training. No constraint is set on the values of the state transition matrix. This method would outperform other AD frameworks that rely on an i.i.d. assumption when there are observations that may appear individually normal, yet when considered as a sequence, are inconsistent with the previously observed sequences.

In the approach proposed here, one can think of the hidden states as different environmental and operational conditions (EOCs) that govern the response of the structure. As a structure responds under varying EOCs, this HMM framework considers the EOCs to decide if structural damage is present or not. Consequently, within this framework, it may occur that a sequence of length  $l$  is labeled as undamaged when another formerly observed sequence of length  $l$  was labeled as damaged. Hence, this framework cannot structurally reject the reversal of damage. To do so, when sufficient damaged sequences are observed, one can use the sequence of  $p(\mathbf{X}_l|\theta_{ML})$  along with the Viterbi algorithm (Viterbi, 1967) to find the most likely point of transition of the system going from an undamaged state to a damaged state.

In contrast, it would be tempting to set the hidden states to be the levels of structural damage. Under such a configuration, one can enforce  $A(i, j) = 0$  for  $i > j$  to ensure that a transition to damage would only progress once in time and not be reversed.

However, training such a model would require observations from the damaged state to be present during HMM training. Such a strategy would be applicable only if multiple structures that are identical are available and one (or some) of the structures has completed its life span with all possible damage cases observed.

Rather than simply using  $p(\mathbf{X}_l | \boldsymbol{\theta}_{ML})$  as proposed herein, there exists more sophisticated methods for sequential AD. An example of such methods can be found in (Görnitz et al., 2015).

### **3.3 Description of the Dataset**

The dataset used in this study is from the Z-24 bridge. This dataset is well known to SHM researchers and has been widely studied in the past (Figueiredo et al., 2014; Peeters & De Roeck, 2001). The dataset consists of hourly measurements of both EOCs and structural responses over a duration of one year starting November 11th, 1997 and lasting until September 10th 1998. During the last month of the measurements, six different damage scenarios were inflicted on the bridge (i.e., settlement of pier, tilt of foundation, spalling of concrete, failure of concrete hinges, failure of anchor heads and failure of tendon wires). The bridge was demolished after the tests. The Z-24 bridge is shown in Figure 3-2.

The environmental measurement system (EMS) measured 53 different environmental conditions. The majority of these measurements consist of temperature measurements at numerous locations on the bridge as well as the ambient temperature; hence the measurements are heavily

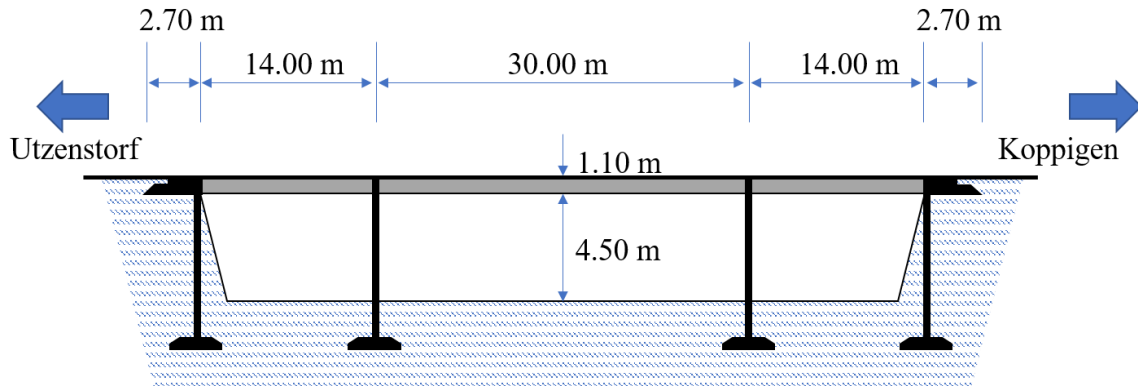


Figure 3-2. The side view of the Z-24 bridge.

correlated. In addition to temperature, other environmental variables such as relative humidity, wind (speed and direction) and precipitation were measured.

The monitoring system also measured structural responses from eight accelerometers located on the bridge. These measurements were collected at a sampling frequency of 100 Hz but passed through a low-pass filter with a cut-off frequency of 30 Hz. The acceleration measurements were sampled every hour and each measurement persisted for roughly 11 minutes and contained roughly 650,000 data points.

Modal frequencies of the structural response are the dominant structural response feature used for analyzing this dataset in the literature. In this work, modal frequencies are extracted using the N4SID algorithm (Van Overschee & De Moor, 2005) implemented in MATLAB.

This algorithm can be implemented using MATLAB's `n4sid` command. The first modal frequency is selected as the damage sensitive feature (DSF) and used to detect structural damage in the bridge.

The dataset prepared for this study contained 5,652 pairs of hourly environmental conditions and structural response measurements. For each observation, an average of the bridge

structural temperatures (measured on the beam webs) and the relative humidity were used to represent the EOCs.

From the 5,652 pairs, the first 3,800 were used during the training and the remaining 1,852 for model testing. The first 1,145 observations in the test dataset were undamaged and the last 707 were from the damaged structure. For the purpose of this work, it is necessary to consider the observations in the temporal order that they were acquired.

### **3.4 Analysis**

#### **3.4.1 Three Tier Damage Detection Framework**

The performance of the proposed method is compared to two other well-known AD techniques which assume that the observations are i.i.d. First, the three-tier modular framework proposed in (Hackell et al., 2016) is used.

This approach is a successful approach for EOCs data normalization and has been applied to numerous SHM problems (Bahrami et al. 2017; Hackell et al., 2016; Tsiapoki et al., 2018). A more detailed description of this framework is found in Chapter 2 of this dissertation. Here, the details of implementation adjusted for the Z-24 dataset is presented.

First the training data is clustered into a smaller number of groups using k-means clustering carried out on the EOC feature space consisting of average relative humidity and average center web temperature. The distribution of the center web temperature is shown in Figure 3-3Figure.

These two EOCs are standardized to remove the effect of differences in their magnitude scales. Within each cluster, during training DSFs (in this case first modal frequency) are used to build the normal region. Next, during the testing stage, each new observation is assigned to the

cluster with the most similar EOCs (in the case of k-means clustering, the cluster with the closest centroid).

Once assigned to a cluster, the DSF is then compared to the normal region of its corresponding cluster. The observation is labeled as undamaged if its DSF lies inside the normal region of its corresponding cluster; otherwise, it is labeled as damaged. Note that the HMM framework takes care of EOC normalization by learning the hidden states.

The results are evaluated for four different number of EOC clusters: namely 5, 6, 7, and 8 clusters. To make a fair comparison, the HMM method is set up with the same number of hidden states as the number of clusters used in the three-tier damage detection methods.

Going forward, the number of EOCs clusters and the number of hidden states can be used interchangeably.

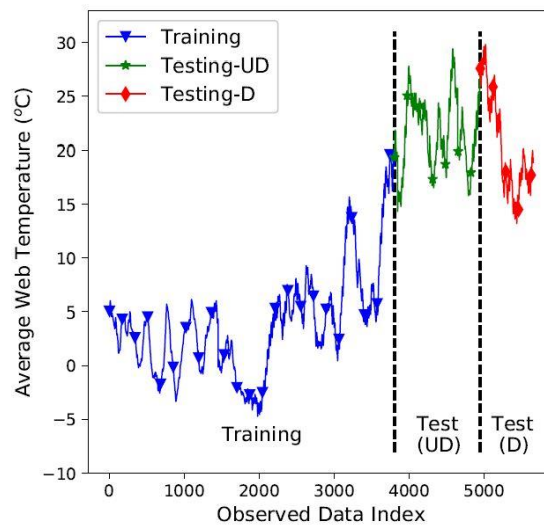


Figure 3-3. Web temperature values for the Z-24 dataset.

To build normal distributions for the DSFs within each EOC cluster, two methods are used. First is the One Class Support Vector Machine (OC-SVM) (Schölkopf et al., 2001). This method attempts to fit a tight boundary around the normal observations in training by taking advantage of

the kernel trick. The required inputs for OC-SVM are the kernel type and its hyper-parameter. Here, a radial basis function kernel, given as  $K(y_i, y_j) = \exp(-\frac{1}{2} \|\frac{y_i - y_j}{2}\|^2)$ , is used. Six different values for the hyper-parameter, namely  $\lambda = \{0.1, 0.5, 1, 2, 5, 10\}$  are explored and the results with the highest performance are reported.

Note that selecting the optimal hyper-parameter for OC-SVM is still an open problem (Wang et al., 2018).

The second method is the Elliptic Envelope (EE) technique which assumes that the data is Gaussian and fits an ellipse to the training data (Rousseeuw & Driessen, 1999). This is in contrast to the OC-SVM which assumes no specific type of distribution for the data. The performance of the EE method clearly deteriorates when the distribution of the DSFs are not Gaussian. However, as the results would show, assuming the modal frequencies in each EOC cluster are Gaussian is reasonable. The models are trained using built-in codes of a Python library named scikit learn (Pedregosa et al., 2011).

For the HMM-based method proposed in this work, sequences of length  $l = \{5, 10, 20, 30\}$  are used in the testing stage. Sequences have no overlap with the exception of the last two sequences for the undamaged (normal) and damaged datasets. This is done to accommodate all of the test datasets. For instance, when  $l = 10$ , the 1,145 undamaged datasets are broken into 115 sequences as follows: the first 114 sequences contain 10 sequential observations without any overlap. So the 114<sup>th</sup> sequence contains undamaged test data points 1,131 to 1,140. The 115<sup>th</sup> sequence contains undamaged test data points 1,136 to 1,145. This is done to ensure that all sequences have the same length and that they also include all the observation in the testing stage.

To compare the performance of the different AD approaches, the receiver-operating characteristic (ROC) curve is used. The ROC curve is a plot of true positive rates versus false

positive rates as the decision boundaries are varied. The ROC curve is a popular tool in evaluating the performance of a classifier.

Ideally, it is desired to achieve a true positive rate of one with a false positive rate of zero which corresponds to correctly labeling all undamaged and damaged datasets. Hence the more the ROC curve tends to the point (0, 1) the better the performance of the classifier. Area under the ROC curve (AUC) is a useful scalar metric to compare multiple ROC curves. An AUC of 1 corresponds to an ideal classifier (correctly classifying all data) and an AUC of 0.5 corresponds to a completely random classifier.

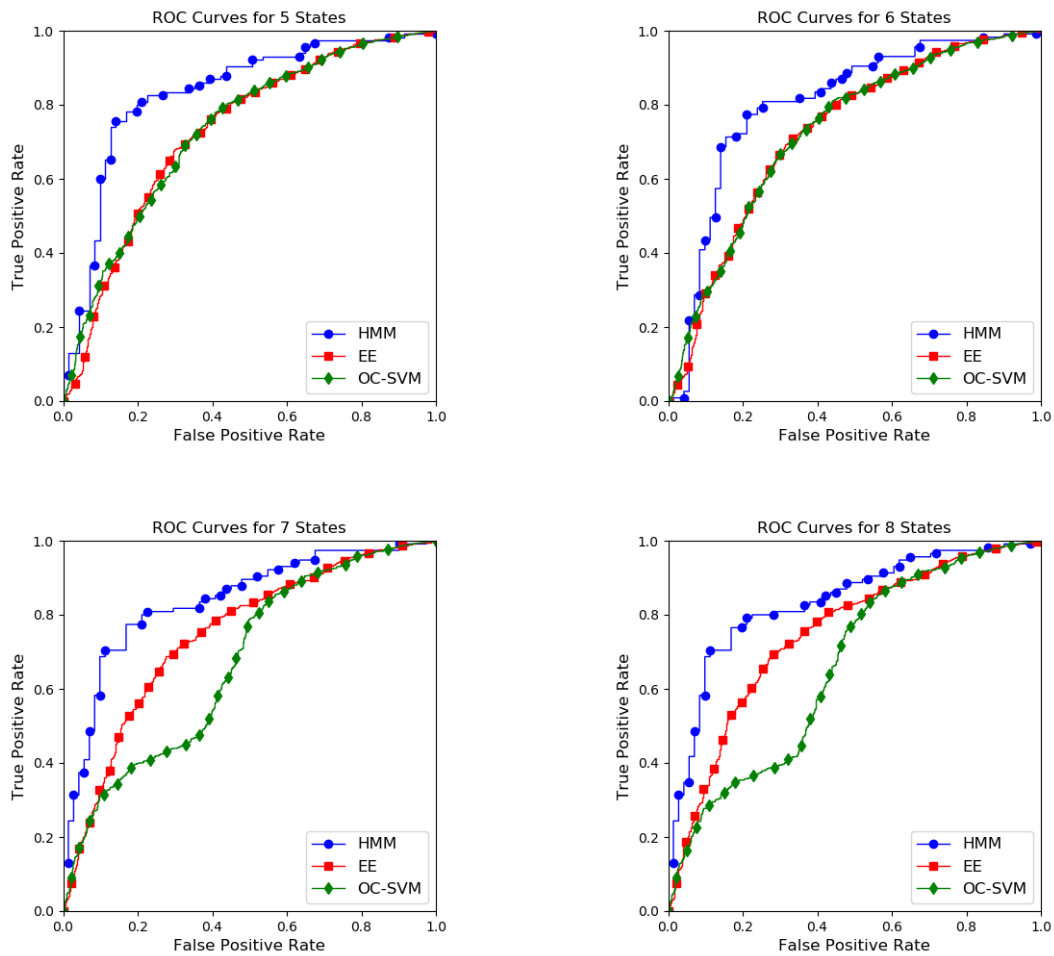


Figure 3-4. ROC curves showing the performance of the three AD methods for four different number of hidden states/clusters.

### 3.4.2 Results and Discussions

Three methods are compared: HMM, the three tier framework using OC-SVM and three tier framework using EE. Figure 3-4 shows the ROCs curves for different number of hidden state for the HMM-based and the two baseline methods (three tier implementation with OC-SVM and EE decision boundaries). For the HMM-based AD results shown in Figure 3-4, the length of the sequences is set to  $l = 10$ . As can be seen, the HMM-based AD outperforms the two baseline methods. Amongst the two baseline methods, it appears that the EE method outperforms OC-SVM. The difference becomes more significant for 7 and 8 EOC clusters (or hidden states in HMM).



Figure 3-5 shows the resulting AUC values for the four different sequence lengths used. The AUC values for the AD methods used can be found in Table 3-1.

In the HMM method, it can be seen that as the length of the sequence increases, the performance of the HMM-based damage detection framework increases as well.

Table 3-1. AUC values for HMM and baseline methods.

No. States /Clusters	HMM (Sequence Length)				EE	OC-SVM
	5	10	20	30		
5	0.77	0.84	0.86	0.88	0.73	0.71
6	0.77	0.80	0.86	0.87	0.73	0.71
7	0.80	0.84	0.86	0.88	0.75	0.66
8	0.80	0.84	0.86	0.87	0.75	0.66

Moreover, when the length of the sequences decreases, the performance of the sequential AD framework converges to that of the two baseline AD frameworks. Such behavior is expected as the hidden structure is explored only within the sequences and the sequences are treated independently. On the contrary, increasing the length of the sequences results in fewer sequences being available during training and testing. For example, when  $l = 10$  for the given dataset, there are 115 undamaged sequences and 71 damaged ones, whereas when  $l = 30$  there are 39 undamaged sequences and 24 damaged ones. Although it has not occurred here, reducing the number of sequences available for inference might have a negative impact on the performance of the framework.

As it can be seen for the majority of the cases, the performance of the AD methods is not significantly influenced by the number of hidden states (with the exception of the OC-SVM for 7

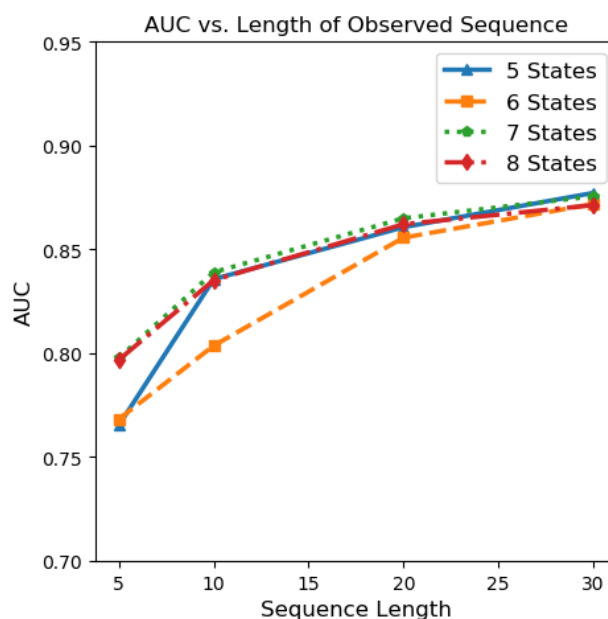


Figure 3-5. HMM AUC values for different sequence lengths.

and 8 clusters). An explanation for this can be found by referring back to Figure 3-3. An important observation is that only a few training observations have similar EOCs to those in the testing stage. Hence, the number of clusters that the training EOCs space is divided into doesn't significantly impact the clusters that the observations in the testing stage are assigned to.

Additionally, rather than selecting the first 3,800 observations, if one were to randomly select 3,800 observations from the 4,945 undamaged observation for training, the EOCs space of that training dataset would have had more similarities with the testing dataset. This would have increased the performance of the AD methods as the training dataset would have been richer.

This highlights a key issue when using data-driven SHM techniques for real-world applications. The testing dataset might have EOCs that are not observed during the training stage. This would be more prevalent at the beginning stages of data collection when the training dataset is still limited. In that case, it is important to keep track of the EOCs and compare them with that of the training dataset available. In that case, using insights from manual inspection of the structure

could be extremely helpful in building a more reliable and capable data-driven model for long-term monitoring of the structure at hand.

Prior to concluding, it is important to restate that the HMM based AD technique introduced in this work used sequences of observations for decision making and treated the sequences independently.

Rather than doing so, one can explore the sequential structure of certain features of the sequences (an example of that feature could be the probabilities  $p(\mathbf{X}_l|\boldsymbol{\theta}_{ML})$ , but other features could be explored as well). Furthermore, the sequences were intentionally forced to have no overlap (with the exception of the compromise made to include all observations in the testing stage). For example, one can choose overlapping sequences.

### **3.5 Conclusion**

In this chapter, a sequential AD technique based on HMM was introduced. The proposed methodology relied solely on undamaged observations in the training stage. A maximum likelihood approach is taken for training the HMM and the set of parameters that result in highest probability for observing the training dataset is selected using scaled forward-backward algorithm. For inference, sequences of fixed length were used and the probability of those sequences was used for decision making. The proposed framework was implemented on the Z-24 dataset.

Two baseline methods treating the observations as i.i.d. were used, namely OC-VM and EE for evaluating the performance of the proposed framework. To deal with variations in EOCs, first the training dataset was clustered via performing k-means clustering on a space consisting of two EOC features, namely standardized temperature of the center web and standardized relative humidity. Models were built for each cluster using the aforementioned OC-SVM and EE methods.

Each test observation was assigned to the cluster with the most similar EOCs and the built model was used for decision making. It was found that the proposed sequential framework outperforms the baseline models.

## Chapter 4.

### Deep Learning Based Bridge Weigh-in-Motion Using Encoders

#### 4.1 Introduction

With an unprecedented number of highway bridges instrumented with sensors for structural monitoring, Bridge WIM (BWIM) has emerged as a viable and cost-effective alternative to the traditional pavement-based WIM stations. Over the past decade, extensive studies have been conducted on BWIM technology (Yu et al., 2016; Zolghadri et al., 2016). With sensors (most commonly strain gauges) installed in bridges, bridge WIM methods rely on measured bridge responses (e.g., strain) to derive weights of passing trucks in real time. Existing bridge WIM methods are mainly based on Moses's algorithm developed in 1979 (Moses, 1979). Assuming the linearity of bridge response, once a unit influence line is determined for the bridge, truck weight properties can be estimated through a least squares solution (Lydon et al., 2016; Moses, 1979). O'Brien et al. (2009) used a regularized solution to the least squares problem which resulted in increased accuracy without the need for additional instrumentation (O'Brien et al., 2009). A key limitation associated with this least squares approach is that the accuracy of such methods is heavily dependent upon the quality of the extracted UIL. Furthermore, for using UIL-based models, knowledge of vehicle speed, axle number and axle spacing is needed (He et al., 2017).

To circumvent the demand for the knowledge of axle numbers of truck load for the BWIM problem, numerous methods have been proposed such as Wavelet domain analysis techniques (Chatterjee et al., 2006; Yu et al., 2017), virtual simply supported beam methods (He et al., 2017)

and optimization techniques such as Lasso regression (He et al., 2019; Pan et al., 2018). Additionally, ML-based techniques have been explored for solving BWIM problem. For instance, Kim et. al. (2009) used ANN for solving the BWIM problem on a dataset with three different types of trucks traversing a bridge (Kim et al., 2009). Similarly, convolutional neural networks (CNN) were used for identifying axle number, vehicle type and vehicle speed for UIL-based bridge WIM with a higher accuracy (Kawakatsu et al., 2018).

In this chapter, a learning-based BWIM method is proposed taking advantage of the recent advances in recurrent neural networks (RNN). Leveraging a large volume of paired bridge response and WIM data, a bidirectional RNN model is trained to predict truck weight characteristics including axle weight, axle spacing, gross weight and travel lane with length-variant bridge strain responses as inputs. The proposed method is evaluated using simulation data obtained from finite element models and proven to be more accurate than the influence line-based bridge WIM method. Furthermore, the robustness of the method against measurement noise in the training labels is explored. The simulations are based on a computer vision-enabled cyber-physical system implemented as a test-bed along a 20-mile I-275 northbound highway corridor in Michigan. The dataset collected from this testbed is the subject of the study in Chapter 5 of this dissertation as well.

## **4.2 Theoretical Background**

In this subsection, the theory used for DL-based BWIM is described. Specifically, encoders utilizing RNNs are trained to learn the predict truck properties from the time series responses corresponding to the same truck. Hence, the input to the encoder is the time series response of the bridge and the output is set truck properties. In this work, the focus is specifically on predicting

axle weights as it is one of the most influential parameters that impact the bridge response to the truck. Obviously, in order to predict the weight of each axle, first, the total number of axles on the truck will be determined. The idea is that the bridge response (i.e. the forced response followed by the free response), carries important information on the truck axle weight distribution, and in presence of sufficient training data, a DL-based model can learn to extract the parameters of interest from the bridge response sequence.

#### 4.2.1 RNN-Based Encoders

The encoder in this research utilizes RNNs to extract important information from the input time series. Let  $X = \{x[1], x[2], \dots, x[T]\}$  represent the input time series. An RNN is a nonlinear transformation that at each time instance  $k$ , uses the input at the same time instance, i.e.  $x[k]$  along with the state from the previous time step,  $\mathbf{s}[k - 1]$  to compute the state at current time step  $\mathbf{s}[k]$  (Goodfellow et al., 2016). In abstract mathematical form, the RNN can be expressed as  $\mathbf{s}[k] = \Phi(\mathbf{s}[k - 1], x[k])$  where  $\Phi$  is used to represent the nonlinear transformation used.

Note that the state of the RNN is a vector with different dimension (usually larger) than the input. In essence, the state of the RNN can be thought of as a unit capturing important information from the time series. Often times, the state is referred to as the hidden state, as it is not observed. To initiate the network, the value of state at time zero is selected in a pre-determined manner. The most common value for the state at time zero is a vector of zeros.

In this research, long short-term memory (LSTM) networks are used as the RNN cell. The equations of LSTM cell are as follows (Paszke et al. 2017, Chung et al. 2014) as shown in Equation 4-1 where  $\mathbf{h}$  and  $\mathbf{c}$  are called the hidden state and cell state, respectively. Furthermore,  $\mathbf{j}$ ,  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{o}$  are referred to as the input, forget, cell and output gates, respectively. It should be noted that, at

each time step, two set of vectors are received from the previous time step, namely the hidden and cell states ( $\mathbf{h}$  and  $\mathbf{c}$ ). Hence, from this point on, when the term hidden state is used for LSTM, it is intended to represent a concatenated vector of the  $\mathbf{h}$  and  $\mathbf{c}$  vectors and for consistency with the RNN literature, this concatenated vector is labeled  $\mathbf{h}$  and the dimension of this vector is referred to as the hidden dimension of the LSTM cell. The schematics of the LSTM cell is shown in Figure 4-1.

In this research, a bi-directional LSTM is used as the encoder. A bi-directional encoder consists of two sets of RNNs with LSTM cells, one passing the information forward in time, labeled as  $\Phi_f$  and referred to as the forward LSTM, and the other passing information backward in time, labeled  $\Phi_b$  and referred to as the backward LSTM. It is common practice to have the same hidden dimensions for the forward and backward LSTMs. Figure 4-2 shows the architecture of the model used in this chapter.

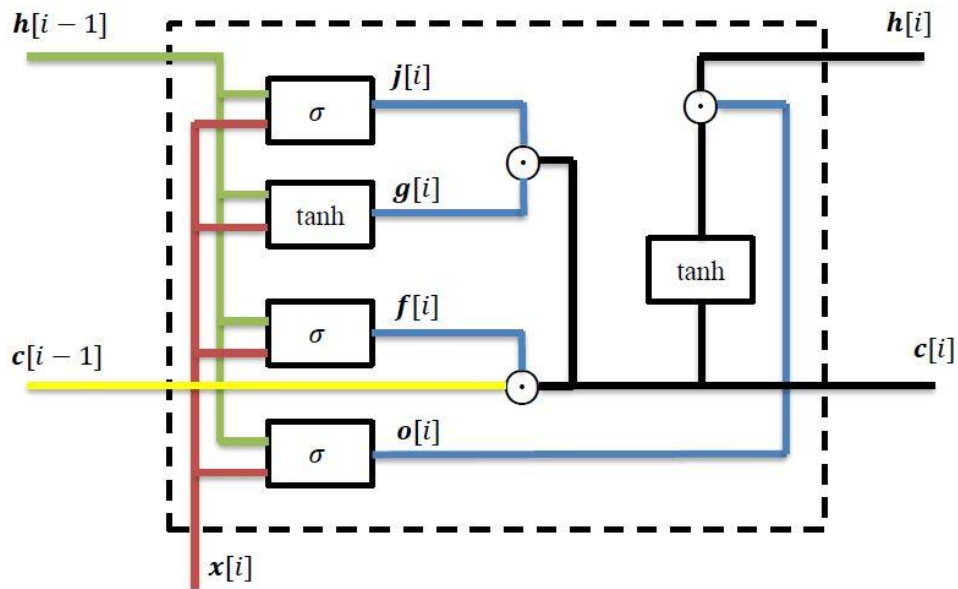


Figure 4-1. Schematics of the LSTM cell.



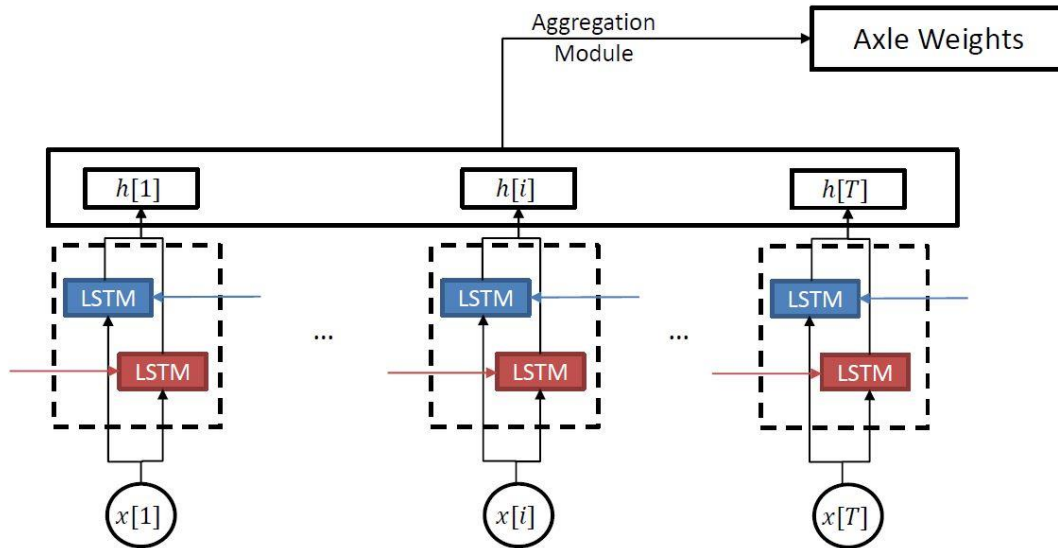


Figure 4-2. Schematic of the Encoder utilizing bi-directional LSTM cells and an aggregation module. The red cells represent the forward LSTMs and the blue cells represent the backward LSTMs.

#### 4.2.2 Hidden State Aggregation

The output of the encoder is truck axle weight distribution. Since the maximum number of axles permissible in the state of Michigan is 11, the output vector of the encoder has a maximum dimension of 11. Let  $d_h$  represent the common hidden dimension of each of the LSTMs. For each bridge response  $X = \{x[1], x[2], \dots, x[T]\}$ , the sequence of hidden states generated will be of dimension  $T \times 2d_h$  (the hidden vector will contain both the forward and backward hidden vectors). Thus, the hidden states are required to be aggregated into an 11dimension space.

In this research, two approaches are taken to hidden dimension aggregation. In the first approach, a fully connected artificial neural network (ANN) is used. The ANN is applied to the last hidden vector. The second approach utilizes a max-pooling layer on the sequence of hidden states generated from the response and then applies a fully connected ANN to the output of the max-pooling layer. Figure 4-3 depicts the schematic application of aggregation modules to the sequence of hidden states.

### 4.3 Description of the Datasets

As stated in Chapter 1, the dataset enabling the implementation of BWIM within this work is collected using an advanced computer vision-enabled monitoring system based on the CPS framework. In this subsection, this monitoring system is described in details to familiarize the reader with this system. Prior to implementing the DL-based method in this chapter, two other simulated dataset are generated for examining the viability of the proposed method on those

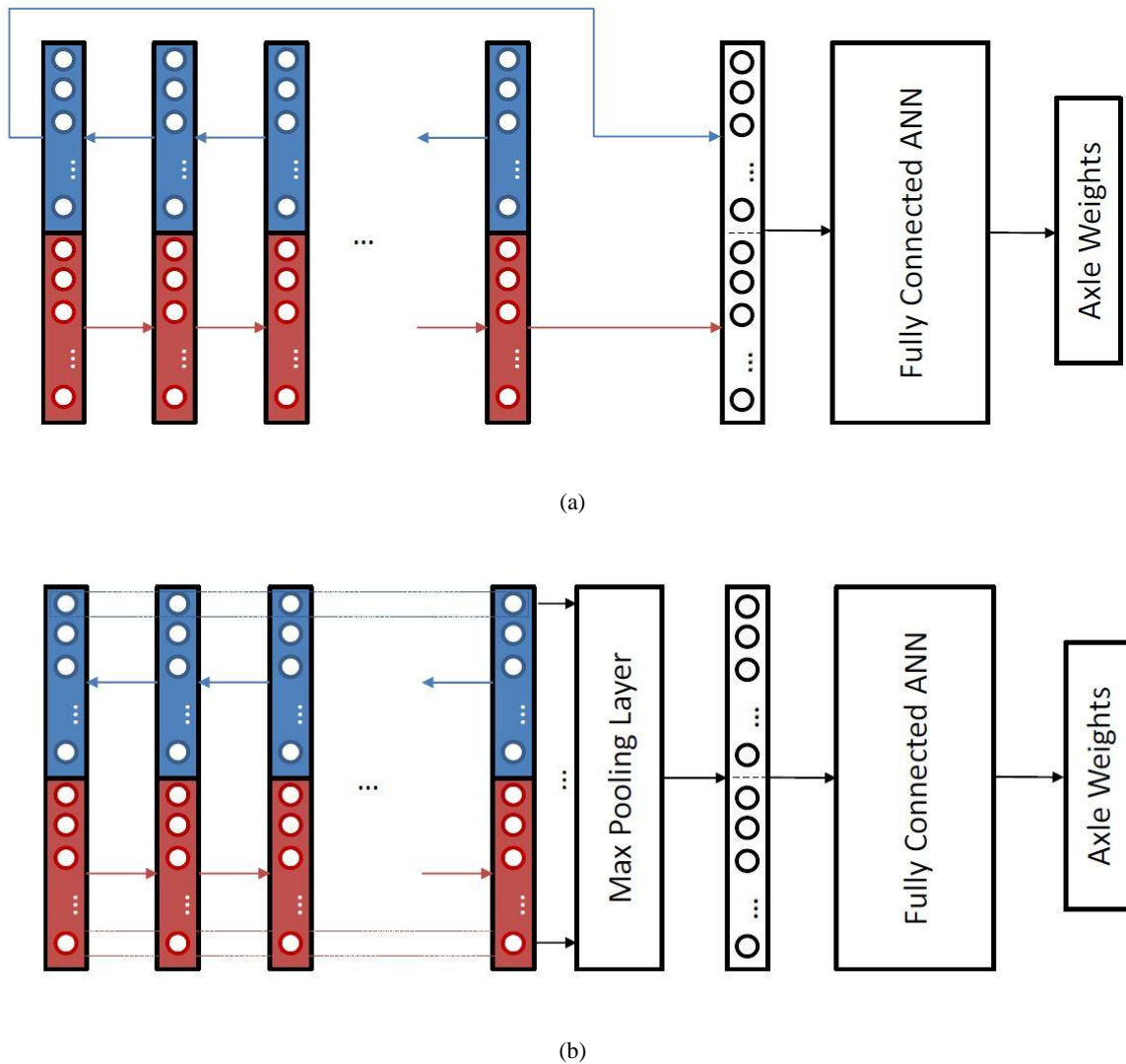


Figure 4-3. Schematics of the aggregation module for a) fully connected ANN b) Max Pooling layer and fully connected aggregation modules.

idealized datasets. Each of the datasets are generated with a specific purpose in mind. These datasets are further described in this subsection.

#### **4.3.1 I-275 Monitoring System**

The experimental testbed is a 20-mile stretch of the I-275 highway corridor located in southeast Michigan. A CPS is installed consisting of four low-cost cameras for highway traffic monitoring, two sets of instrumented bridges (for which the instrumentation would be discussed in details), and a WIM station. The set of cameras and bridge instrumentations are operated by the Laboratory of Intelligent Systems and Technologies (LIST) and the WIM station is operated by Michigan DOT (MDOT). The purpose of this CPS architecture is to enable correlating the bridge responses to the trucks that induce the response through using camera feed as well as measuring truck weightage properties inducing the measured bridge responses using the WIM station.

Two of the bridges are instrumented with wireless sensor nodes to measure the structural response of each bridge to highway traffic. The first bridge is the Telegraph Road Bridge (TRB) and the second bridge is the Newburg Road Bridge (NRB). Both bridges are built in 1973; TRB is instrumented for bridge SHM application since 2011 and NRB since 2016. TRB is a three span bridge with the middle span being connected to other two by a pin and hanger mechanism. NRB is a single-span bridge. The overall length of TRB is 224ft and the length of NRB is 105ft. Both bridges have a concrete-on-steel-plate girder super structure and each have 7 girders supporting three lanes of highway traffic. For detailed description of the bridges, interested readers are referred to (Hou, 2020; Hou et al., 2019).

The bridge monitoring systems consist of three types of sensors, namely, strain gages, accelerometers and thermistors for temperature measurement. Narada WSN is used to establish

communication between the sensors and the base station installed on the bridge. The base station consists of three components: 1- an NVIDIA Jetson TX2 module single board computer, 2- an AT&T Velocity MF861 LTE modem for internet access and 3- a CC2420 IEEE 802.15.4 RF transceiver used for establishing wireless communication between the Narada WSNs and the base station. A base station is installed on each of the two bridges and each base station communicates with the sensor nodes installed on the bridge through wireless connectivity established by Narada WSNs and with other components of the monitoring system as well as the main server located in LIST through internet connectivity. The base station is equipped with a 160W solar panel that charges a 12V 40 A-hr sealed lead acid (SLA) battery which powers the base station. At each WSN, up to four analog sensors can be interfaced with the Narada unit for structural monitoring.

The next important component of the monitoring system is the set of traffic cameras. As mentioned previously, four traffic cameras are placed along the testbed of this study. Each traffic camera consists of the following five items: 1- a Logitech C930 webcam placed in a waterproof enclosure is used to monitor the highway traffic, 2- The operations of each camera is controlled and driven by an NVIDIA Jetson TX2 single board computer, 3- An AT&T Velocity MF861 LTE modem is used to provide internet connectivity, 4- a 160W solar panel is used to power the battery supplying the system with electricity and 5- a 12V 40 A-hr SLA battery is used within each camera system. As obvious, the modem, battery, and solar panel used for the cameras and base stations are the same. Additionally, each single board computer has an embedded integrated NVIDIA Pascal GPU and a hex-core ARMv8 64-bit CPU which enables real-time image processing on the node for vehicle detection. Overall, four traffic cameras are located across the corridor, one at the interchange between I-275 and I-75 (i.e., the entrance of the testbed), one at TRB, another at NRB

and the fourth located underneath the Pennsylvania Road Bridge where the WIM station is located. The components used in the base station and camera system is shown in Figure 4-4.

The final component of the system is the WIM station located at the end (i.e., northern most point of the testbed) underneath the Pennsylvania Road Bridge. The WIM station is a two-lane type II station with quartz sensors buried under the highway pavement at the slow and middle lane of the highway (i.e., right lane and middle lane). The WIM station communicates the collected data with a remote server operated and maintained by MDOT. With the help of MDOT technicians, this database is queried regularly to acquire WIM station data for the purpose of integration with data collected by the other components of the monitoring system. The measured weight records for the WIM station contains the following 9 attributes: 1) measurement time stamp, 2) Federal

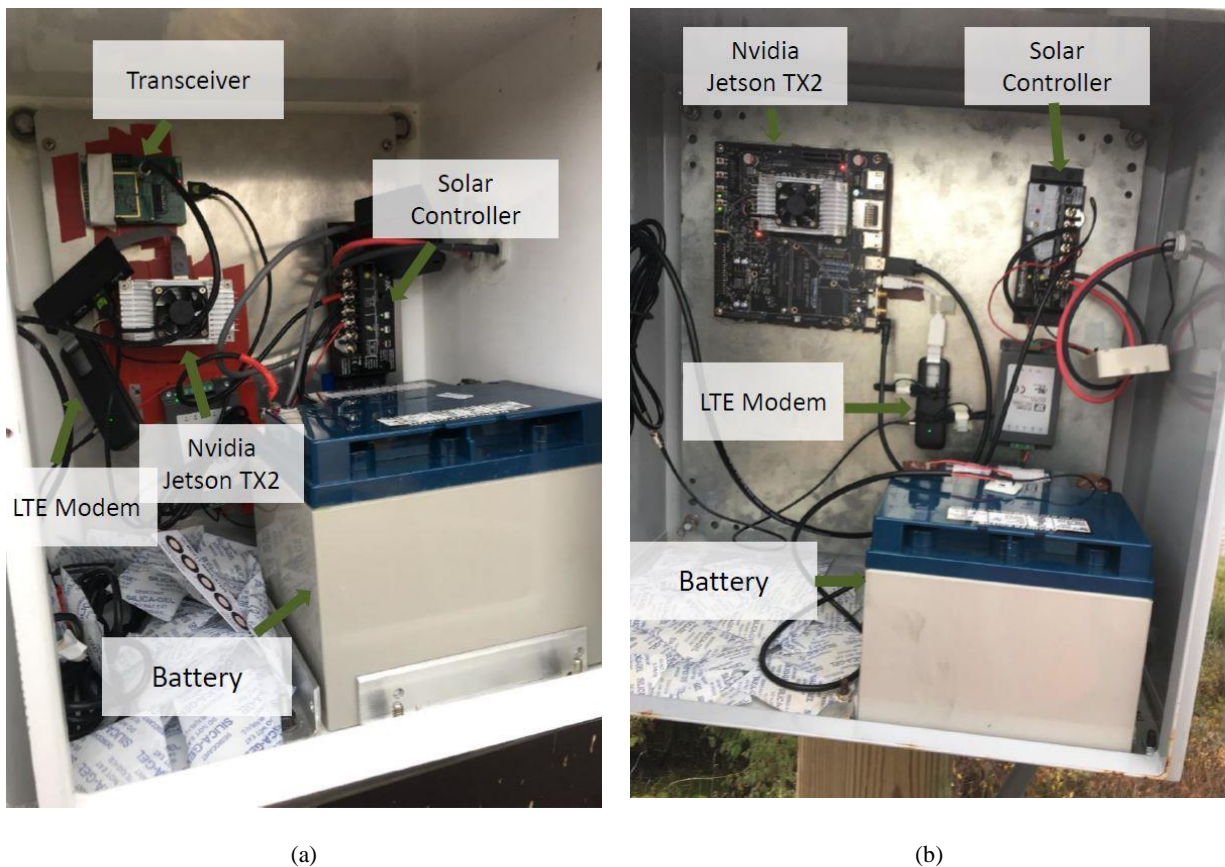


Figure 4-4. Components of the monitoring system in a) base station b) camera monitoring system.

Highway Administration (FHWA) vehicle class, 3) vehicle speed, 4) vehicle gross weight, 5) number of axles, 6) axle weights, 7) axle spacings, 8) vehicle direction and 9) vehicle lane assignment.

#### **4.3.2 Data Acquisition Process**

The data acquisition (DAQ) process consists of two steps; the first step is done in a real-time manner during DAQ and the second step is done in an offline manner after responses have been collected.

The real-time DAQ process consists of numerous messages being exchanged through the internet between the camera and base stations installed across the corridor. An object detection algorithm named You Only Look Once (YOLO) (Redmon et al., 2016) is continuously running on the camera installed at the beginning of the corridor (i.e., at the intersection of I-75 and I-275 highway corridor) for real-time truck detection through the traffic. Once a truck is detected by this camera, a message would be sent across the rest of the network triggering the DAQ processing. Consequently, this camera is also referred to as the trigger camera.

The first message is sent from the trigger camera to TRB camera and TRB base station. Upon receiving the detection signal from the trigger camera, the TRB camera and base station will wait for 70s and then start the DAQ process at TRB. The TRB camera records highway video feed and the TRB base station records the responses measured by sensors installed on bridges. Both systems record the response for a duration of 100s. Next, the TRB camera sends a message to NRB camera and NRB base stations. The message commands the systems to collect data 330s after the trigger camera has detected the truck and both systems conduct DAQ for a durations of 120s (i.e., NRB camera records highway traffic and NRB base station collects bridge responses from WSNs

installed on the bridges for a duration of 120s). Finally, NRB camera sends a message to the WIM camera effectively 810s after the truck detection to record highway traffic at WIM station for a duration of 360s. The time differences between the initiation of DAQ at different stations is calculated based on the physical distance between different station and an assumption that trucks are traveling at a speed of 60mph. The DAQ duration is increased from TRB to NRB and from NRB to WIM station to increase the chance of detecting the truck that is observed at the trigger camera. An alternative to triggering-based DAQ is DAQ at fixed time intervals (e.g., DAQ with in an hourly manner). As shown by Hou (2020) (Hou, 2020), triggering-based DAQ would increase the total number of trucks sampled within the response significantly.

After the observations are collected by each of the stations, the cameras automatically send the video feeds to the LIST server in a daily manner and erase the feeds from local computers to provide storage for DAQ in next days. In contrast, the data is queried in a manual manner with lower frequency from TRB and NRB base stations. The reason behind this is that video feeds consume large volumes of memory storage on the NVIDIA Jetson TX2 computers and can potentially fill up the memory of these computers in a matter of few days. Hence the data is retracted automatically in a daily manner to ensure enough memory exists for DAQ everyday. In contrast, SHM records require very low storage volume and hence can be retracted with a higher frequency.

After the datasets from cameras, base stations and the WIM station are all collected, the offline data processing step begins to identify and re-identify trucks across the corridor as well as extract the corresponding bridge responses and WIM station measurements corresponding to each truck load. The details of this process is outlined at (Hou et al., 2019). Here, a brief description is presented. A CV-based truck re-identification algorithm utilizing YOLO-v3 network and a triplet

network is used to identify and re-identify trucks across the three measurement stations. In other words, this re-identification framework is used to identify trucks and detect the same truck between the three pairs of camera stations, namely the TRB-NRB cameras, TRB-WIM station cameras and NRB-WIM station cameras. Based on the time stamps on the video feeds, the bridge responses to the trucks and the WIM station information corresponding to each truck is isolated as well. Consequently, three valuable sets of data are constructed: pairs of TRB and NRB bridge responses to the same truck load, TRB response to a truck load along with WIM station information and NRB response to a truck load along with WIM station information corresponding to that load. Needless to say that some trucks are re-identified across at least two of the three pairs of cameras resulting in a (obviously smaller sized) data sample consisting of pairs of TRB and NRB responses to the same truck load along with the WIM station properties of that truck. Figure 4-5 depicts a sample truck that is tracked across the monitoring testbed.

This chapter utilizes NRB-WIM station pairs of trucks for the BWIM problem. In chapter 5, time series forecasting models are used to link the pairs of bridge response to the same truck load. Additionally, the signal pre-processing techniques used for the experimental dataset is described in Chapter 5 in greater details.



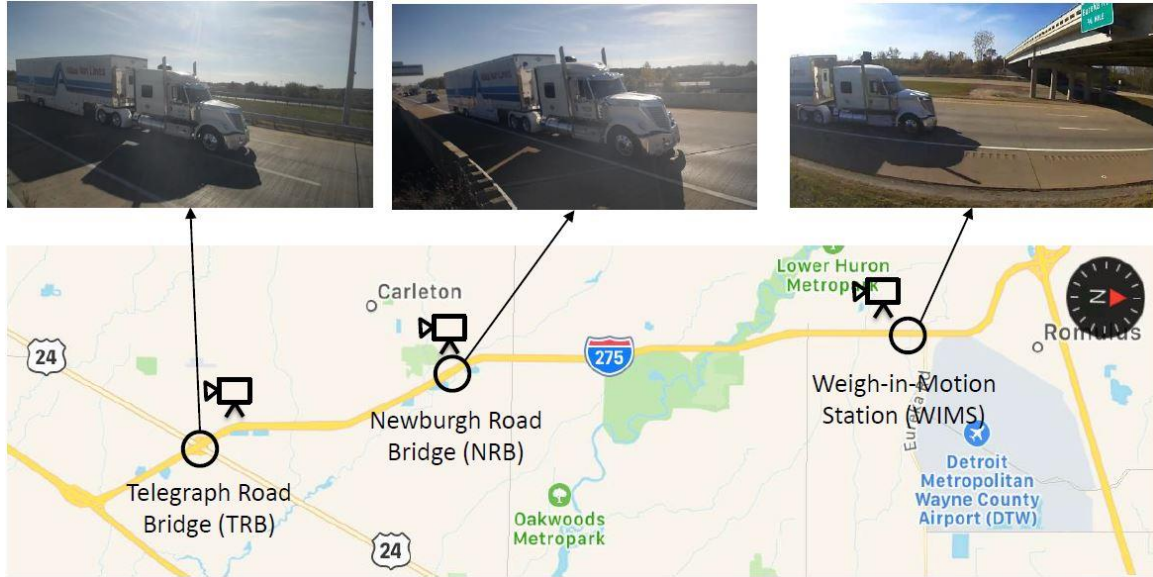


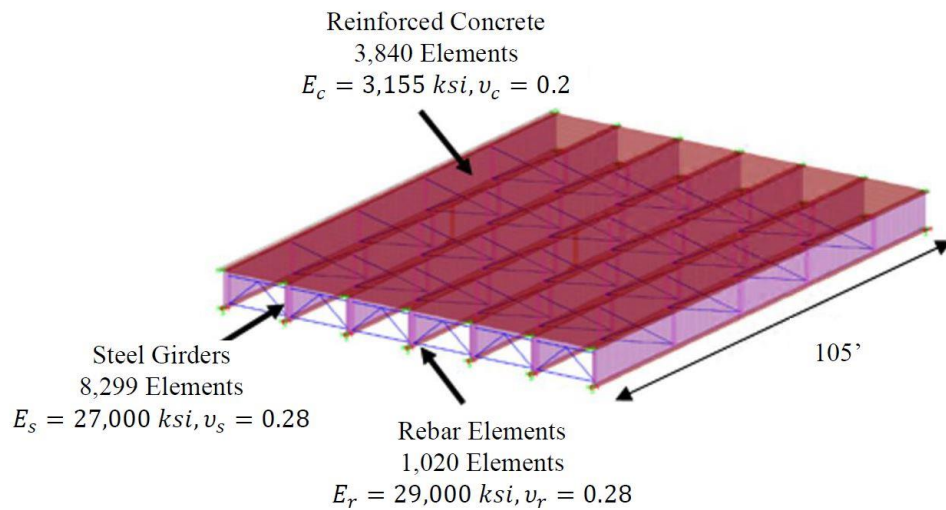
Figure 4-5. Images of a sample truck identified and re-identified across the I-275 monitoring testbed.

### 4.3.3 Description of the Finite Element Model

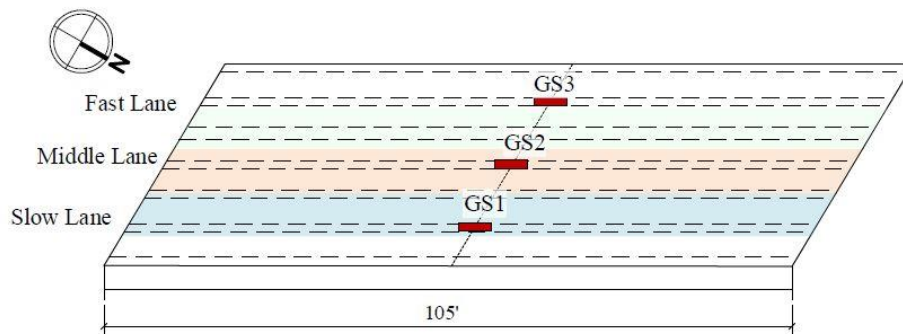
As a first step for both this chapter and the next chapter, a Finite Element (FE) model is developed to evaluate the viability of the proposed methods on a simulated environment prior to implementation on datasets collected from the real-world measurements. Here, the model used for generating this dataset is described in details.

In this study, finite element (FE) models are created for both TRB and NRB in CSiBridge (Computers and Structures, 2011) according to the dimensions and properties of the bridges to evaluate the feasibility of the proposed Encoder model in this chapter and the Seq2Seq model that is used in the next chapter. Though, the results for the Seq2Seq model are not shown in this dissertation, they can be found at (Bahrami et al., 2021). Numerical simulations are carried out for both bridges using the same truck to imitate the scenario of monitoring a given truck passing through the highway corridor loading the TRB and NRB.

For NRB, a total of 12,139 shell elements are used to model the reinforced slab (3,840) and steel girders (8,299), and a total of 1,020 brace elements are used to model the lateral bracings. The concrete has a Young's modulus of 3,155 ksi with a Poisson's ratio of 0.2. The girder steel has a Young's modulus of 27,000 ksi and a Poisson's ratio of 0.28. The Young's modulus for the rebar steel is 29,000 ksi. Unlike the TRB, NRB has two dominant modal frequencies: 4.0Hz and 5.8Hz. The model properties for both bridges have been fine-tuned by model updating, thus, their modal



(a)



(b)

Figure 4-6. a) Schematic drawing for the NRB model, b) sensor locations for the bridge weigh-in-motion.

frequencies closely match those observed in the actual bridges. Figure 4-6 shows the description of the description of the FE model along with sensor locations used to record the bridge response.

Bridge responses were generated by sampling truck properties from the set of WIM station measurements collected by the MDOT. Considering FHWA vehicle classes 1 to 3 represents lightweight vehicles, whose loads on the bridges are not of major interest. In addition, according to the local speed limit (70 mph for cars, and 60 mph for trucks), measurements with abnormal speeds are ignored. Thus, the measured WIM dataset is pruned to only consider trucks with a vehicle class between 4 to 13, and vehicle speeds between 40 to 85 mph.

The simulation has a time step of 0.01 seconds (sampling frequency is 100 Hz). As it is evident, the strain response contains not only the forced response triggered by truck but also the free vibration response after the truck is off the bridge. Thus, each bridge response can be divided into two parts: (1) forced response whose duration and amplitude depend on truck speed and weights, and (2) free vibration response after the truck is off. To capture enough structural information of the bridges, the simulation duration is set to 3 seconds for NRB. This ensures at least four cycles of free vibration is recorded for each bridge. The description of the model used for TRB can be found at (Bahrami et al., 2021).

In this chapter, the FE model for NRB is used to simulate bridge responses to truck traffic and the set of bridge strain responses along with the WIM station properties are used for evaluating the proposed methodology for the BWIM problem. A key point to be kept in mind is that here, truck WIM properties inducing bridge vibrations are assumed to be known precisely. This is an idealization of the real-world scenario where WIM station measurements actually contain measurement noise. For the WIM station used in I-275 corridor which is a Type II WIM station, the measurements may be subject to up to 15% relative error with respect to gross vehicle weight and

30% relative error with respect to axle weight (ASTM, 2017). As such, use of the simulated data is beneficial in that it enables the evaluation of the Encoder model along with different hidden state aggregation modules in an idealized scenario where exact WIM characteristics of the load are known, a scenario which is not achievable for real-world measurements.

#### 4.4 Results and Discussions

The FE model described in section 4.3.3 is used to generate the simulated response dataset. A dataset consisting of 3,800 response sequences is generated by sampling WIM station information collected by MDOT. From these observations, 3,300 are used for training, 200 for validation and 300 for testing. An important point evident within the dataset is that the data is imbalanced in terms of axle numbers for the trucks. Similar pattern is observed in Chapter 5 as well. This is due to the fact that some trucks are more common compared to others. For instance, the most common axle number for the trucks is 5, 2 and 11 axle trucks. As stated in section 4.3.3., since the duration simulation and sampling frequencies are 3s and 100Hz respectively, each generated time history contains 301 data points.

To train the model, A stochastic optimizer known as the Adam optimizer (Kingma & Ba, 2014) is used to train the models. The initial learning rate was selected to be 0.01, and was reduced after each epoch using the following relations:

$$l_e = l_0 \times 0.96^e \quad \text{Eq. 4-2}$$

where  $l_e$  and  $l_0$  represent the learning rate at epoch  $e$  and epoch zero respectively (i.e.,  $l_0 = 0.01$ ). The reduction of learning rate is important in increasing the performance of the training for higher epochs. To ensure better convergence, a gradient clipping step is implemented to clip the gradient

to a maximum norm of 2. The hidden dimension is set to be equal to 512, and the mini-batch size of 20 is used for stochastic optimization.

The loss function used is the L1-smooth loss function defined as:

$$L1_{smooth}(x, y) = \frac{1}{n} \sum_i z_i \quad \text{Eq. 4-3}$$

$$z_i = \begin{cases} \frac{1}{2}(x_i - y_i)^2 & |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5 & o. t. w \end{cases}$$

where  $n$  is the dimension of the two input vectors. The loss function is used on the set of 11-dimension weight vectors, i.e., for an observation with predicted and label (i.e., actual) weight vectors  $w_{pred}$  and  $w_{label}$ , the loss function is  $L1_{smooth}(w_{pred}, w_{label})$ .

The evaluation in this section includes two parts. Firstly, the accuracy of the three types of hidden state aggregation modules is investigated to select the best for real-world use. Secondly, the robustness of the proposed model against noises in training labels is tested for the selected model after the first stage.

The accuracy of the model is evaluated with five metrics, namely average axle weight error, average vehicle gross weight error, 95th percentile of axle weight error, 95th percentile of vehicle gross weight error and axle number accuracy. For a specific kind of weight  $x_w$  (i.e., axle weight or gross weight), the error,  $e$ , is defined as Equation 4-4:

$$e = \frac{|xw_{pred} - xw_{label}|}{xw_{label}} \quad \text{Eq. 4-4}$$

The error is only calculated based on ground-truth labels and zero-value axle weight is not counted. For example, if the model predicts out 6 non-zero axle weights for a 5-axle truck the last axle weight is not calculated since the error will be infinite. The amplitude of the 95th percentile error is determined such that 95% of the calculated errors are smaller than this value. Regarding

the accuracy of axle number, for a specific truck, if the model predict the correct number of axles it counts one towards true positives and the final accuracy is calculated as the ratio of true positives over the total number of trucks. Apart from accuracy, we also compare the inference time to proceed a single sequence with the same length among three different variations.

Three variations of the proposed model with different hidden state modules are trained and tested using the test dataset with the metrics described above. The results are shown in Table 4-1. It can be observed from the table that the max-pooling approach obtained the best performance in terms of accuracy. Both the 95th axle weight error (25.97%) and the 95th gross weight error (5.88%) meet the accuracy requirement of a Type II WIM station. Compared to the ground-truth labels, most of the negative samples with wrongly predicted axle numbers have a difference of 1. Based on the above outcome, the bidirectional LSTM model with the max-pooling aggregation module is selected for further analysis.

Table 4-1. Test results for two types of hidden state aggregation.

	Avg. axle weight error	95 <sup>th</sup> axle weight error	Avg. gross weight error	95 <sup>th</sup> gross weight error	axle no. accuracy
Last step	13.86%	46.33%	3.35%	9.56%	65.67%
Max pooling	7.56%	25.97%	1.66%	5.88%	89.00%

Real-world WIM station measurements suffer a certain level of noises following a normal distribution (Prozzi & Hong, 2007). To test the robustness of the model against noisy training labels, Gaussian noises are randomly sampled to corrupt the training labels (i.e., axle weights). The mean of the Gaussian noise is always 0 and the noises are added in terms of relative error according to Equation 4-4. Five stages of noisiness levels are tested where the standard deviation of the Gaussian error distribution,  $\sigma$ , is increased following the set {0, 0.0125, 0.025, 0.05, 0.1,

0.15} resulting in the 95<sup>th</sup> percentile error in axle weights being 0, 2.5%, 5%, 10%, 20% and 30%, respectively. It should be noted that even though the model is trained using noisy labels it is still tested using uncorrected test labels. The results of the experiment is shown in Table 4-2.

Table 4-2. Testing results for the proposed Encoder model with training labels being corrupted with Gaussian noise.

Noise Level (95 <sup>th</sup> percentile)	0%	2.5%	5%	10%	20%	30%
Avg. axle weight error	7.56%	8.27%	9.02%	9.48%	9.43%	10.09%
95 <sup>th</sup> axle weight error	25.97%	27.93%	30.41%	31.27%	29.76%	31.94%
Avg. gross weight error	1.66%	1.63%	2.56%	2.57%	2.79%	2.95%
95 <sup>th</sup> gross weight error	5.88%	5.11%	8.62%	8.16%	9.68%	8.69%
# axle accuracy	89.00%	87.67%	86.67%	86.67%	88.67%	88.67%

It can be observed that with an increased level of noisiness, the accuracy of the model does not degrade much. With a 30% noisiness level, the accuracy of the model can still nearly satisfy the requirements of a Type II WIM station. As a result, it can be concluded that it is possible to train a model based on corrupted training data generated from a Type II WIM station to meet the same criterion for the same type of WIM station. Besides, it is very likely that better tuned parameters and more training data can help further improve model performance.

#### 4.5 Conclusions

Real-time truck weight measurements such as gross weight and axle weight play an important role in the enforcement of highway truck size and weight regulations. The long-term statistics also provide a rational basis for freight transportation planning and highway asset management. Recently, the emergence of bridge WIM technology presents a more cost-effective alternative for

weighing trucks. With sensors installed in bridges, bridge WIM relies on measured bridge responses to derive weights of passing trucks in real time. In this study, a learning-based bridge WIM method is proposed. Utilizing a large simulated dataset consisting of paired bridge response and WIM data, a bidirectional LSTM network is trained to predict truck weight characteristics including axle weights and gross weight with bridge strain responses as inputs. The proposed method is evaluated using simulation data obtained from finite element models and proven to be sufficiently accurate for a type II WIM station. The robustness of the proposed method against sensing noises in training data is also investigated.

For future work, the proposed methodology will be implemented on the experimental dataset from both the TRB and NRB. Additionally, the effects of more factors, such as ambient temperature, size of training dataset, sensor locations, sensing sampling rate, etc. on the accuracy of the bridge WIM system will be investigated.



## **Chapter 5.**

### **A Sequence-to-Sequence Model for Joint Bridge Response Forecasting**

#### **5.1 Introduction**

Highway bridges are an integral part of the infrastructure system. As of now, the major source of funding for the highway infrastructure comes from the fuel tax revenues collected by the federal and state governments. However, this model is proven unsustainable for two reasons. First, the fuel efficiency of vehicles has constantly increased over the past years resulting in less fuel consumed and hence less revenue generated per vehicle-miles traveled. Second, the increase rate of infrastructure deterioration has resulted in higher demand for funding to maintain civil infrastructure systems. Consequently, there has been an increased attention on alternative financing mechanisms and models for the civil infrastructure systems.

Rather than charging a flat rate from the fuel consumed, a recent focus has been on quantifying the cost of consumed infrastructure life by each vehicle and attempting to charge vehicles based on that cost. A recent implementation of such financing paradigm can be found in dynamic pricing of highway consumed life by over-weight vehicles. Numerous researchers have attempted to quantify the consumed life by over-weight/over-sized trucks and change the flat fee structure to a dynamically priced one (Gungor et al., 2018, 2019).

In addition to determining the cost of consumed life for financing, bridge response forecasting can be used to better understand bridge response given to corridor loading patterns.

Such understanding can enhance highway asset management planning, an issue that is extremely important given the increased strain on infrastructure systems and limited funding for infrastructure asset management. Furthermore, accurate response forecasting can enable better understanding of vehicle-bridge dynamic interactions (VBI). VBI has received significant attention over the past few years and the body of literature allocated to VBI has been ever increasing since the widespread adoption of numerical models (Ding et al., 2010; Elhattab et al., 2016; González et al., 2008; Zhang et al., 2018; Zhao & Uddin, 2014). The goal of such studies have been focused on better understanding of VBI and bridge dynamic response amplification can enhance the understanding of bridge response to truck traffic and can be used for a less conservative design of highway bridges. While numerical models provide reasonable approximation of bridge response, all such models have flaws when attempting to estimate bridge response to the large and extremely diverse set of truck population that a bridge would serve throughout its entire life span. Hence an accurate time series model can be developed to better study VBI and quantify the dynamic amplification response of the bridge under varying traffic load based on our knowledge of other bridge responses to the same truck load.

Finally, in addition to the previously mentioned applications, accurate forecasting of bridge responses to a given truck load can be used to control connected and automated trucks when traveling over a bridge so as to regulate bridge response to truck traffic with the goal of maximizing the life span of the bridge by reducing the incremental damage implemented on the bridge from truck traffic. This can be done by allocating lane and speed trajectories to the trucks communicating with the infrastructure through vehicle-to-infrastructure (V2I) and vehicle-to-everything (V2X) communication technologies.

An enabler of such ambitious goal is the recent advances in sensing and analytics tools available to civil engineers. Recent advances in sensing and wireless communication technologies have enabled civil engineers to monitor bridge responses and construct massive datasets in a cost-effective manner. Furthermore, the field of artificial intelligence (AI) has undergone a recent boom and significant attention has been focused to this field lately. As such, numerous civil engineers have adopted AI and machine learning (ML) for solving pressing problems in this field. An important recent implementation of computer vision in civil engineering was to identify and re-identify truck traffic in a corridor developed by Hou et al . By synchronizing video feed collected by cameras monitoring with bridge response monitoring systems on the bridges, Hou. et. al. managed to construct a valuable dataset containing the response of two highway bridges to the same truck along with the truck information measured by a weigh-in-motion station (WIMS) in the I-275 highway corridor in Michigan.

In this work, we employ deep learning-based time series forecasting model to predict the response of one bridge to a given truck given the response of another bridge to the same truck. To do so, a certain deep learning architecture named sequence-to-sequence (Seq2Seq) model is used. Such model was initially developed for the purpose of neural machine translation (Sutskever et al. 2014). Later on, the model was adopted for time series applications and proved to be efficient (Wilms et al. 2018, Salinas et al. 2019). Perhaps the most important characteristic of the Seq2Seq model is the capability to develop a complex mapping between an input and an output sequence. To do so, the input of the model is fed to an encoder to generate a context vector. In essence, the encoder attempts to condense all the information in the input into the context vector. Next, a decoder is used to unravel the output sequence using the context vector. Thus, sometimes this architecture is also referred to as the encoder-decoder architecture.

The rest of the chapter is organized as follows. In the next section, we will give a thorough description of the datasets used within this work along with techniques used for signal conditioning and processing. Next, we will provide a brief theoretical background on the encoder-decoder model. In the results and discussions subsection, we will show the results of deploying the proposed framework on the a real-world bridge response dataset. Finally, in the last section, we will conclude with the conclusions along with future directions of research.

## **5.2 Description of the Dataset**

The dataset used within this chapter is collected by the same system as described in Chapter 4. As such, the reader is referred to Chapter 4 for a detailed description of the monitoring system implemented in the I-275 highway corridor. However, the dataset used is the portion which consists of identified and re-identified trucks on TRB and NRB rather than between bridges and WIM stations (as it will be described later in the results sections, a portion of the dataset is also re-identified on the WIM station and a subset of two bridge responses to the same load along with the corresponding WIM station information is gathered). In the remainder of this subsection, signal pre-processing techniques employed for selecting bridge responses to the trucks is described. The overall steps taken for signal pre-processing is shown in Figure 5-1. In the following subsections, each component of the pre-processing step will be described in further details. There are three lanes on each of the bridges, namely slow, middle and fast lanes. The total number of trucks identified on TRB and re-identified on NRB are shown in Table 5–1. As evident, the most frequent lane combination is the set of trucks that traverse through the middle lane of TRB and slow lane of NRB (the reason for this fact is that the slow lane of TRB is an entry ramp from the Telegraph road which is underneath the TRB to the I-275 corridor). Hence, in this research, only this combination of trucks is considered for the propose of joint bridge response forecasting. The

Table 5-1. Frequency of re-identified vehicles for different truck lanes combinations on the I-275 highway.

	TRB Slow	TRB Middle	TRB Fast
NRB Slow	733	7664	434
NRB Middle	80	2103	153
NRB Fast	0	6	8

response of the bridge in TRB is measured by a sensor installed on a girder underneath the middle lane and the response of NRB is measured by a sensor installed on a girder underneath the slow lane. The location of sensors is intentionally selected so as to select the measure bridge responses exactly at the lane that the vehicles of interest are travelling on. The schematic location of sensors on the bridge is shown in Figure 5-2.

There are three lanes on each of the bridges, namely slow, middle and fast lanes. The total number of trucks identified on TRB and re-identified on NRB are shown in Table 5–1. As evident, the most frequent lane combination is the set of trucks that traverse through the middle lane of

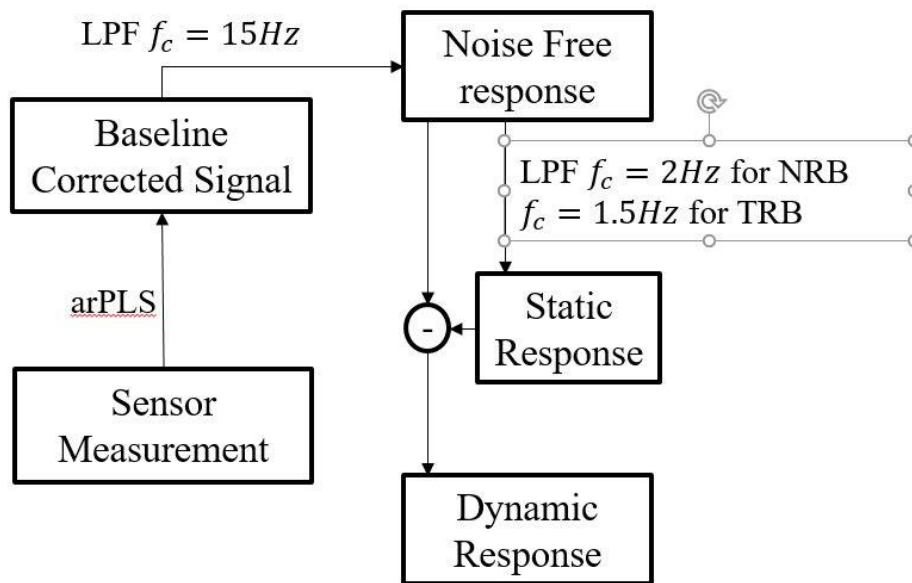


Figure 5-1. Signal pre-processing steps for joint bridge response forecasting.

TRB and slow lane of NRB (the reason for this fact is that the slow lane of TRB is an entry ramp from the Telegraph road which is underneath the TRB to the I-275 corridor). Hence, in this research, only this combination of trucks is considered for the propose of joint bridge response forecasting. The response of the bridge in TRB is measured by a sensor installed on a girder underneath the middle lane and the response of NRB is measured by a sensor installed on a girder underneath the slow lane. The location of sensors is intentionally selected so as to select the measure bridge responses exactly at the lane that the vehicles of interest are travelling on. The schematic location of sensors on the bridge is shown in Figure 5-2.

### **5.2.1 Baseline Removal**

As stated in the description of the monitoring system, the WSNs used for measuring bridge responses operate in a triggered manner. As such, when the system is triggered, a time-varying offset is introduced in the measurements which is the result of the transient response of capacitive elements in the measurement circuitry. To remove this time-varying offset, a baseline removal algorithm using asymmetrically reweighted penalized least squares (named arPLS for short), developed by Baek et. al. (Baek et al., 2015) is used . Initially introduced for removing varying background and noise for spectroscopy applications, arPLS has proven to be a capable and efficient tool for noise removal in other time series data as well. Additionally, this algorithm has been successfully implemented on the I-275 bridge response dataset as well. Here arPLS is used for baseline removal within the raw time series strain responses collected for TRB and NRB. There exists two key hyperparameters for arPLS. First is the smoothness parameter  $\lambda$  and the second parameter is the termination ratio when the changes in the selected weights for baseline removal are low enough for stopping the iterative algorithm. Within this work,  $\lambda = 10^{13}$  was used for both

TRB and NRB datasets. However, due to the differences in vibration characteristics of the two bridges, the termination ratios were selected to be  $2 \times 10^{-2}$  and  $10^{-2}$  for NRB and TRB respectively.

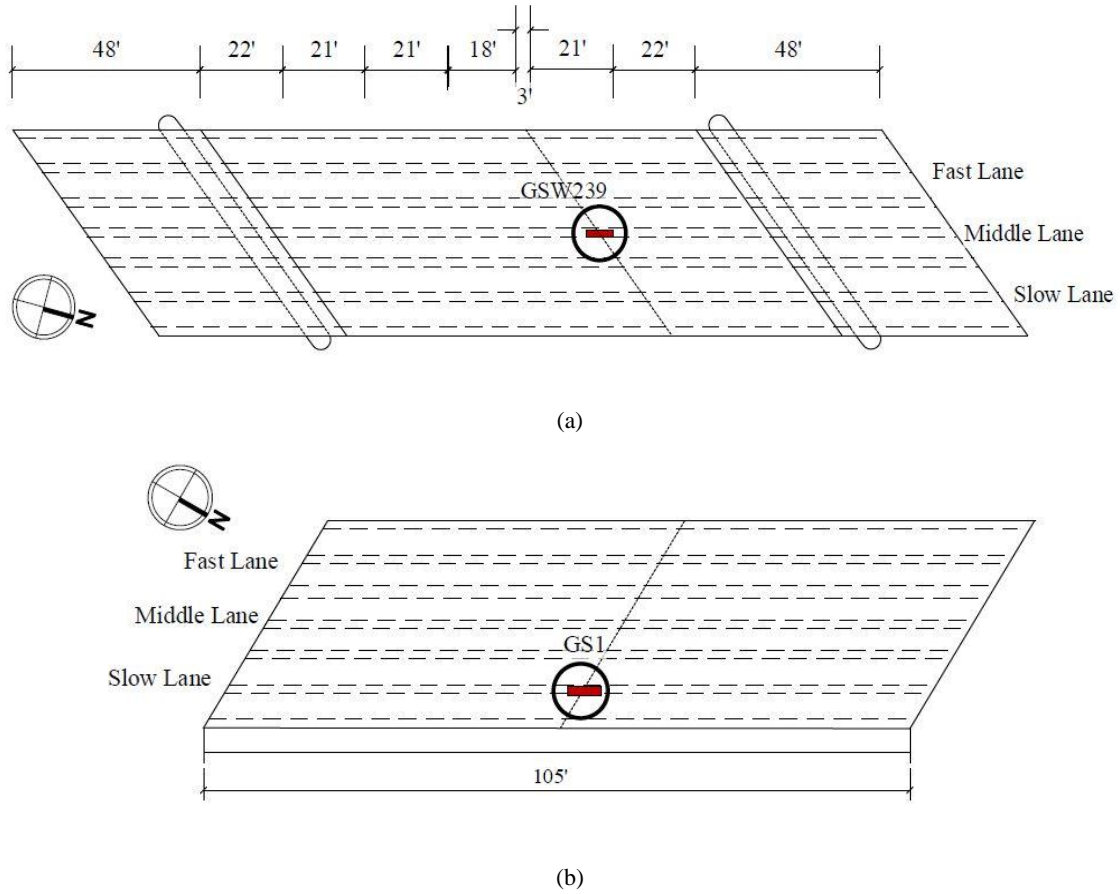


Figure 5-2. Strain measurement locations on a) TRB and b) NRB.

For the sake of conciseness, the details of the arPLS are not described here. However, in order to better understand the algorithm and the significance of the corresponding hyperparameters, the interested reader is referred to (Baek et al., 2015).

### 5.2.2 Measurement Noise Removal

In addition to the time varying background noise discussed in the previous subsection, there exists measurement noise in the form of high frequency, zero-mean random variations within the sensor

readings due to imperfections within measurement instruments. To remove this undesirable noise, a low-pass filter is used. Based on the engineering judgement of the team, signal components with frequencies above 15Hz were considered to be due to measurement noise based on the understandings from the behavior of the bridges. Consequently, an elliptic filter was designed and implemented in MATLAB. The filter passband and stopband frequencies were set to 15Hz and 20Hz respectively and passband and stopband attenuations were selected to be 1dB and 50dB respectively. The same filter was used for both TRB and NRB responses as the random measurement noise was expected to be identical for both bridges.

### **5.2.3 Signal Decomposition**

When dealing with bridge response data, it is of high interest to decompose the bridge response time series into static and dynamic components. By definition, the static component of bridge response is caused only by the load of the truck while the dynamic component is a consequence of the dynamic interactions between the truck suspension system and the bridge. Consequently, the static response of the truck is only affected by the total truck weight and weight distribution on the truck. In contrast, the dynamic component of the response not only depends on the previous two factors, but also on other parameters such as truck's speed, suspension system characteristics, tire conditions and surface conditions of the bridge.

Ideally, the static response is achieved when the truck is passed through the bridge with near zero speed (also known as crawl speed). The dynamic truck response is the response of the bridge to a truck traveling at a given speed subtracted by the static response of the bridge to the same truck.



Thus, in order to decompose a bridge measurement into static and dynamic components, it is sufficient to know the bridge static response to the truck. Since for a given truck it is not possible to determine the actual static response defined previously, engineers attempt to estimate the static response by low-pass filtering the signal with a very low cutoff frequency. The cutoff frequency is selected such that it is well below the first modal frequency of the bridge. The rationale behind such approach is that by selecting a cutoff frequency well below the first mode of structural response, all dynamic components of the signal will be removed and hence the resulting output of the signal represents the component without any dynamics. While it is obvious that the estimated static response using such approach could deviate from the actual static response, the approximation errors are considered reasonable.

For selecting the static component of the signal, a Butterworth filter is designed in MATLAB. For both TRB and NRB, a 10-th order filter was selected. The cutoff frequencies were selected to be 2Hz and 1.5Hz for NRB and TRB respectively. Note that the first modal frequencies for NRB and TRB were 4.0Hz and 2.17Hz respectively and hence the cutoff frequencies were selected to be below these modal frequencies. The dynamic component of the signal was extracted by subtracting the estimated static response from the actual measured bridge response (also known as mixed signal). A sample signal is decomposed into the corresponding static and dynamic components in Figure 5-3.

### **5.3 Theory**

In this subsection, the theoretical background of the time series forecasting models used as well as the steps taken for training these models are discussed.

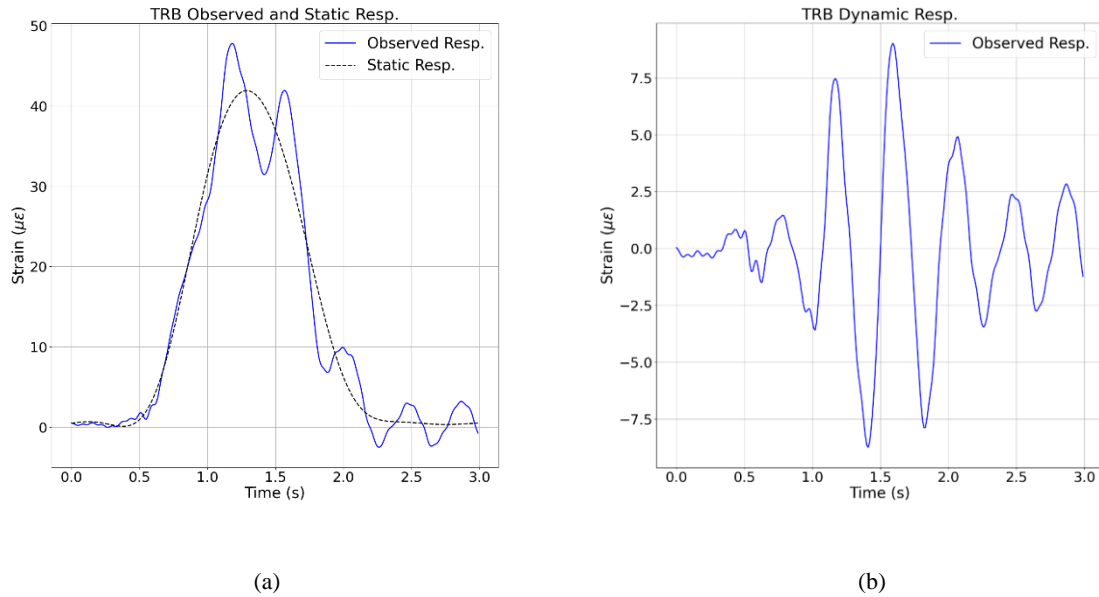


Figure 5-3. Sample signal decomposition. a) represents the observed TRB response (after noise removal) along with the static response signal b) the dynamic response signal.

### 5.3.1 Seq2Seq Model

The backbone of a Seq2Seq model is a recurrent neural network (RNN). An RNN can be viewed as a unit cell that is rolled over itself for a set number of times. At each time step, the cell takes an input  $x[i]$ , and the hidden state from the previous step  $h[i - 1]$  and returns the current hidden state  $h[i]$ . The network aims at capturing sequential information within the data presented to it through the use of hidden states from the previous time steps. The main computation of the RNN is the unit cell  $\Phi$ , where  $\Phi_E$  denotes the cell for the encoder and  $\Phi_D$  for the decoder. The RNN architecture is used to model an encoder whose design is optimized to output a low dimension context vector using input time series  $x[i]$ ; the decoder uses the context vector to feed another RNN that outputs another time series  $y[i]$ . In this study,  $x$  and  $y$  correspond to responses of the two bridges to the same truck. The overall architecture of the Seq2Seq model used within this chapter is shown in Figure 5-4.

Two different cell types are used in this study: namely gated recurrent unit (GRU) and LSTM cells. The brief descriptions of these two cell types are presented in this section. For more description of these cell types, interested readers are referred to (Chung et al. 2014) and (Malhotra et al. 2016) for details. The RNN cells apply a combination of linear matrix multiplications and non-linear transformations. The equations describing the GRU are given as (Paszke et al. 2017, Chung et al. 2014):

$$\begin{aligned}
 \mathbf{r}[i] &= \sigma(\mathbf{W}_{xr}\mathbf{x}[i] + \mathbf{b}_{xr} + \mathbf{W}_{hr}\mathbf{h}[i - 1] + \mathbf{b}_{hr}) \\
 \mathbf{z}[i] &= \sigma(\mathbf{W}_{xz}\mathbf{x}[i] + \mathbf{b}_{xz} + \mathbf{W}_{hz}\mathbf{h}[i - 1] + \mathbf{b}_{hz}) \\
 \mathbf{n}[i] &= \tanh(\mathbf{W}_{xn}\mathbf{x}[i] + \mathbf{b}_{xn} + \mathbf{r}[i] \odot \mathbf{W}_{hn}\mathbf{h}[i - 1] + \mathbf{b}_{hn}) \\
 \mathbf{h}[i] &= (1 - \mathbf{z}[i]) \odot \mathbf{n}[i] + \mathbf{z}[i] \odot \mathbf{h}[i - 1]
 \end{aligned}
 \tag{Eq. 5-1}$$

where  $\tanh$  is the hyperbolic tangent function,  $\odot$  represents the entrywise product, and  $\sigma$  represents the sigmoid function defined as  $\sigma(\cdot) = 1/(1 + \exp(\cdot))$ . Additionally, the index  $i$  represents the time index to show how the current hidden state is related to both the current input and previous hidden state.  $\mathbf{x}[i]$  is the input at time  $i$ . The terms  $\mathbf{r}[i]$ ,  $\mathbf{z}[i]$ , and  $\mathbf{n}[i]$  are termed the reset, update, and new gates, respectively. Finally, the  $\mathbf{W}$ 's and  $\mathbf{B}$ 's are the weights and biases of the linear transformations. The schematic of the GRU cell is shown in Figure 5-5. The description of the LSTM cell can be found in section 4.2 and Equation 4-1 expresses the equations pertaining to that cell.

In summary, the functionality of this cell unit in the encoder RNN can be expressed by:

$$\mathbf{h}[i] = \Phi(\mathbf{x}[i], \mathbf{h}[i - 1]) \quad \text{Eq. 5-2}$$

where  $\mathbf{x}[i]$ ,  $\mathbf{h}[i]$ , and  $\mathbf{h}[i - 1]$  are the input, hidden state at time  $i$ , and the hidden state at time  $i - 1$ , respectively. For the GRU cell, this hidden state is basically the  $\mathbf{h}$  vector in Equation 5-1.

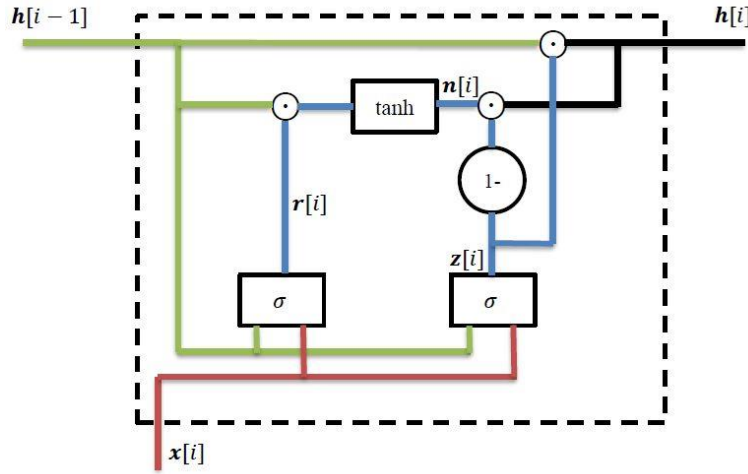


Figure 5-5. Schematic representation of the GRU cell.

While for the LSTM cell, this hidden state is a concatenated vector of  $\mathbf{h}$  and  $\mathbf{c}$  vectors in Equation 4-1.

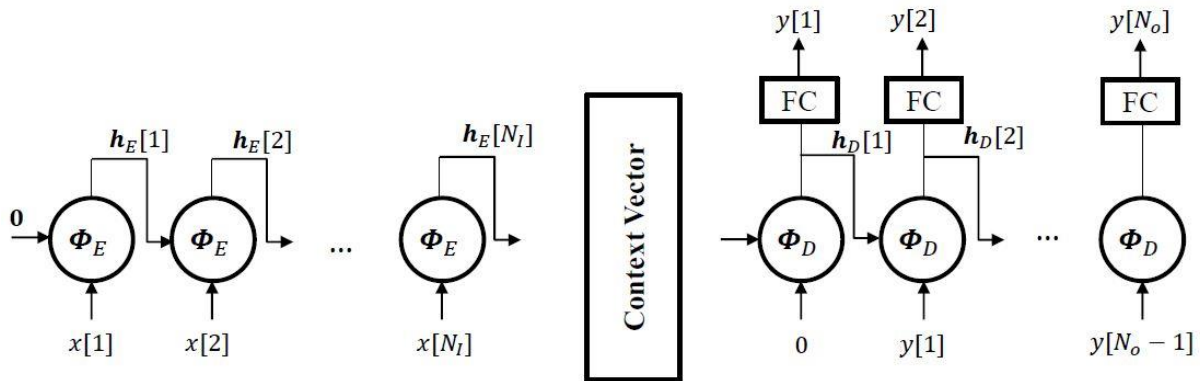


Figure 5-4. Seq2Seq model architecture. The Encoder condenses the information from the input signal into the context vector and the Decoder uses this vector to forecast the output response.

The Seq2Seq model used here consists of two different RNNs: one is the encoder  $\Phi_E$  and the second is the decoder  $\Phi_D$  (this structure is also referred to as an encoder-decoder architecture). The encoder takes the input time series and condenses the information into a fixed-length vector which is referred to as the context vector. The context vector is the last hidden state of the encoder but serves as the input on the decoder's side. On the decoder side, the encoder's last hidden state (context vector) is fed in with the decoder then predicting the output using that context vector. For each time step, the decoder recurrent cell's output is used as the input to the cell at the next time step. This is in contrast to the encoder where the actual time series values are used at each time step.

The dimensions of the hidden states of the encoder and the decoder have to be the same. Meanwhile, the dimensions of the hidden state of the decoder could be different than that of the output. Hence, a fully connected layer is used on the decoder side of the model. This layer acts as follows: let  $\mathbf{h}_d[i]$  be the decoder's hidden state at time step  $i$ ; then, this hidden state is multiplied by a weight matrix  $\mathbf{W}_{da}$  such the product is a scalar.

### 5.3.2 Training the Seq2Seq Model

The Seq2Seq model is implemented in PyTorch (Paszke et al. 2017). This subsection contains technical details of the implementation and training the model using that software tool.

A key feature within the Seq2Seq model is the dimension of the context vector in the last hidden state of the encoder. A set of three different hidden dimensions, namely  $\hat{H} = \{32, 64, 128\}$  for both cell types are iterated on and the one that results in highest performance is selected.

To train the model, a stochastic gradient-based optimizer named Adam (Kingma & Ba 2014) is. Similar to other stochastic gradient-based approaches, at each iteration, Adam takes a

subset of the training dataset (called a mini-batch) and approximates the gradient of the MSE with respect to the weights. The optimizer then takes a step using the gradient at that iteration. Consequently, the size of the mini-batches, referred to as the batch-size, is another important training feature (so called hyperparameter). Two different batch sizes, namely  $\hat{B} = \{20, 50\}$  are used and the one with best performance is selected.

Epoch is another hyperparameter to be set. It refers to the number of times the entire set (consisting of multiple batches) is used to update the neural network. Here the dataset in training is 1500 observations divided into batches (with network weights updated after each batch). During an epoch the training will repeat the learning process on 15 batches when batch size is 100, 30 times when batch size is 50, and so on. The number of epochs used during training will be described shortly.

Another important hyperparameter set prior to training is the learning rate for the Adam optimizer. Adjusting the learning rate throughout training is critically important in achieving an optimal set of network weights.. Initially, the learning rate must be high to allow large enough steps towards the optimal weights but these steps must be gradually reduced to prevent overshooting a local minimum of the cost function. To adjust the learning rate, a built-in feature of PyTorch named ReduceLROnPlateau is used. This function keeps track of the MSE on the validation dataset used after each epoch to test the Seq2Seq model. If this value doesn't decrease for 10 epochs, it reduces the learning rate to a half of its previous value. It should be noted that it is up to the user to decide on the learning rate and the number of epochs to wait before reducing the rate. In this chapter, the initial learning rate is selected as 0.02.

Overall, a maximum of 2,000 epochs are used for training the Seq2Seq model. However, it was realized that beyond a certain point, the model overfitted the training data where the error

on the validation dataset would not decrease (and may increase). Thus, to speed up the process of search for optimal hyperparameters, training was terminated if the validation error did not decrease for 70 epochs. For each value of hidden dimension and batch size, the set of weights that resulted in the minimum MSE on the validation dataset were stored. Additionally, a section of each plot has been magnified to depict the divergence of the training and validation errors.

### 5.3.3 ARX Model

To validate the capacity of the proposed encoder-decoder architecture, a linear ARX model is employed as a baseline model. The ARX model can be expressed as follows:

$$y'[\eta] = \alpha_1\psi[\eta - 1] + \alpha_2\psi[\eta - 2] + \dots + \alpha_p\psi[\eta - p] + \beta_0x[\eta + k] + \beta_1x[\eta + k - 1] + \dots + \beta_qx[\eta + k - q] \quad \text{Eq. 5-3}$$

where  $p$  is the number of lags on the output (poles),  $q$  is the number of lags on the input (zeros) and  $k$  is the offset between the input and output. The set of parameters  $\{\alpha_1, \dots, \alpha_p, \beta_0, \beta_1, \dots, \beta_q\}$  are the weights in the ARX model. Here,  $\psi[\eta]$  is the target (NRB's) response and  $y'[\eta]$  is the ARX model forecasting NRB response at the  $\eta^{\text{th}}$  time step, respectively.

This model can be expressed in the form of multiplication of a matrix with the vector of weights as follows. For each time step  $\eta$ , the right-hand side of Equation (5) can be expressed as the inner product of a row vector  $\Theta_\eta = [\psi[\eta - 1], \psi[\eta - 2], \dots, \psi[\eta - p], x[\eta + k], \dots, x[\eta + k - q]]$  and a column vector being  $\mathbf{w}' = [\alpha_1; \dots; \alpha_p; \beta_0; \dots; \beta_q]$ . Let  $\Theta$  be a matrix whose rows are  $\Theta_\eta$  vectors and let  $\mathbf{y}'$  be column vector whose elements are  $y'[\eta]$ . Then the ARX model can be expressed as a matrix multiplication in the following form:

$$\Theta \mathbf{w}' = \mathbf{y}' \quad \text{Eq. 5-4}$$

To find the set of optimal weights,  $\mathbf{w}'_{opt}$ , for a selected set of model orders and offset, a regularized least-squares approach is taken (Golub et al. 1999). Let  $\boldsymbol{\psi}$  be a column vector containing all target variables constructed similar to  $\mathbf{y}'$ . Then:

$$\mathbf{w}'_{opt} = \operatorname{argmin}_{\mathbf{w}'} \left( \frac{1}{2} \|\boldsymbol{\Theta}\mathbf{w}' - \boldsymbol{\psi}\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}'\|_2^2 \right) \quad \text{Eq. 5-5}$$

where  $\lambda$  is the regularization parameter and  $\|\cdot\|_2$  represents the  $l$ -2 norm. The solution to Equation 5-5 is given by:

$$\mathbf{w}'_{opt} = (\boldsymbol{\Theta}^T \boldsymbol{\Theta} + \lambda \mathbf{I})^{-1} \boldsymbol{\Theta}^T \boldsymbol{\psi} \quad \text{Eq. 5-6}$$

where  $\boldsymbol{\Theta}^T$  representing the transpose of matrix  $\boldsymbol{\Theta}$ , and  $\mathbf{I}$  being the identity matrix with the appropriate size. Equation 5-6 is solved using Python's Numpy package (Van Der Walt et al. 2011).

For training, the true values of the target sequence are used in the  $\boldsymbol{\psi}$  vector ( $\boldsymbol{\psi} = \mathbf{y}'$ ). However, in the validation and testing stages, only the first  $p$  values match the targets. For the remaining time steps, the forecast of the ARX model is used. This implementation is similar to that of the Seq2Seq model when the model forecast of the NRB response at previous time steps are used as opposed to the actual values of the target sequence.

To find the optimal set of hyperparameters, a grid search similar to that of the Seq2Seq model was conducted. The values for  $p$  and  $q$  were from the set  $\hat{P} = \{5, 10, \dots, 70\}$ , the value for  $k$  was selected to be zero as the lengths of the two time series were equal, and the  $\lambda$  parameters were from  $\hat{\Lambda} = \{0.01, 0.1, 0.5, 1, 2, 5, 10, 100\}$ . For each combination of hyperparameters, the ARX model was trained by using Equation 5-6 and the trained model was used on the validation dataset. The combination of hyper-parameters resulting in the lowest validation error were used in the model.



### 5.3.4 Regression-based Baseline

Another baseline used within this work is based on linear regression. This method was suggested by Professor Byon during the pre-defense session. The rationale of fitting the model is as follows. First a linear regression model is fitted to each input and target observation to relate the truck response at each stage of the observation to the time. In essence, the coefficients of regression essentially capture important information contained within the response data and the response can be reconstructed using those regression coefficients. Similar to Equation 5-3, let  $x[\eta]$  and  $\psi[\eta]$  represent the response strain response for the input and target time series at time step  $\eta$ . Additionally, let  $t[\eta]$  represent the time corresponding to time step  $\eta$  (in this work, since responses are sampled at 100Hz, simply  $t[\eta] = \eta/100$ ). Then:

$$\hat{x}[\eta] = \vartheta_0 + \vartheta_1 t[\eta] + \dots + \vartheta_{o_x} t[\eta]^{o_x} \quad \text{Eq. 5-7}$$

and:

$$\hat{\psi}[\eta] = \varrho_0 + \varrho_1 t[\eta] + \dots + \varrho_{o_y} t[\eta]^{o_y} \quad \text{Eq. 5-8}$$

where  $\hat{x}[\eta]$  and  $\hat{\psi}[\eta]$  are the predicted input and target responses at time step  $\eta$  through the regression models and  $o_x$  and  $o_y$  represent the order of regression models used for the input and target respectively.

Note that the above models are fitted to one pair of observation corresponding to a single truck load. Let  $\Xi$  and  $Y$  represent matrices containing the set of all regression coefficients corresponding to training inputs and targets, respectively. In other words, for the  $\iota$ -th sample in the training dataset,  $\iota$ -th row of  $\Xi$  is the vector of coefficients  $[\vartheta_0, \vartheta_1, \dots, \vartheta_{o_x}]$  and the  $\iota$ -th row of  $Y$  is the vector of coefficients  $[\varrho_0, \varrho_1, \dots, \varrho_{o_y}]$  fitted to the input and target observations respectively.

Another regression model is used to predict  $Y$  using  $\Xi$  and hence relate the input and target observations to each other. Here, a linear regression model is used:

$$\hat{Y} = \Xi\Gamma + \zeta \quad \text{Eq. 5-9}$$

where  $\Gamma$  is an  $o_x \times o_y$  matrix and  $\zeta$  is the bias vector with a length of  $o_y$ . Note that whilst the above

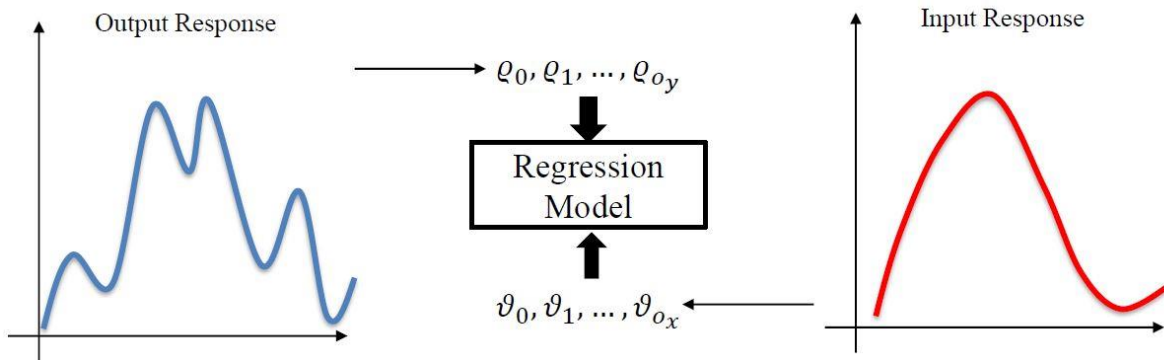


Figure 5-6. Schematic representation of the regression-based baseline model. The information content of each signal is captured in regression coefficients and another regression model is used to related these coefficient vectors.

equation utilizes a linear model to relate the set of coefficients, other non-linear alternatives (such as an ANN) can be used to relate the set of regression coefficients.

To train the model, least squares regression coefficients are extracted from each pair of input and target observation in the training dataset. Then,  $\Xi$  and  $Y$  are constructed and  $\Gamma$  and  $\zeta$  in Equation 5-9 are found again by minimalizing the squared error. At this point, a model relating inputs and targets are constructed. For each validation/test observation, a linear regression model is fitted to the input response using Equation 5-7. Then  $\Gamma$  and  $\zeta$  found in Equation 5-9 are used to estimate the set of regression coefficients for the target response. Afterwards, the target response can be estimated using Equation 5-8. A grid search is conducted to find model orders  $o_x$  and  $o_y$  that minimize the validation error. Figure 5-6 provides the schematic representation of this baseline model for better visualization.

### 5.3.5 A Note on the Baseline Models

As it may have been noticed by now, there are stark differences between the baseline models and the proposed Seq2Seq model. Here, we highlight the differences and caution the reader to note that the superior performance of the Seq2Seq model in this context must not be interpreted as the superiority of Seq2Seq model over the ARX and Regression-based baselines as a whole and on all universal applications.

In summary, the greatest two differences between the ARX and Seq2Seq models are: (1) the first  $p$  terms of the predicted results in the ARX model match exactly that of the target values, whereas the Seq2Seq values are initiated with zeros and only the model forecasts are used from the very beginning, and (2) the encoder observes the entire input sequence and then passes the context vector to the decoder, whereas in the ARX model, at each time step, only a subset of the input observations are used. The regression-based model alleviates this matter through capturing the entire content of input and target responses within the regression coefficients.

The key point to emphasize is that the goal of this chapter is to develop a model enabling the forecast of the response of the downstream bridge to a truck load using the response of the upstream bridge to the same load. Towards this end, the suitability of the Seq2Seq architecture for this problem must be considered a strong point of this model as opposed to being viewed as an unfair advantage over the proposed linear baseline models. None of the time series based models (both the Seq2Seq model and the baselines) are developed by the author. The author does not claim any credit for any of the models nor does the author claim general superiority of one over others. Whilst the author has attempted to utilize the linear baseline models to fullest capacity, there may well be other strategies to increase the performance of baseline models. Furthermore, alternative models can be proposed to relate the input and target time series, models either in time domain or

frequency domain and either linear or nonlinear. Any of these models might outperform the Seq2Seq model. That case would be even a more favorable occurrence, as the goal of the research is to forecast the response on the downstream bridge as accurately as possible. Due to suitable architecture, the Seq2Seq model is considered as the main model here. As it will become evident, the only disadvantage of the Seq2Seq model is the high computational cost of training the model. Once the model is trained, the inference of the target response for new observations will be very rapid.

### 5.3.6 Error Metrics

As stated previously, the loss function used for training the Seq2Seq model is the Mean Squared Error (MSE). The MSE is defined as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y[i] - \hat{y}[i])^2 \quad \text{Eq. 5-10}$$

In this research, the bridge responses have the unit of micro-strains ( $\mu\epsilon$ ). Consequently, the MSE loss function has the unit of ( $\mu\epsilon^2$ ). While the MSE loss function contains important information, when reporting the performance of the model, the square root of the MSE loss value is reported to ensure that the reported metric has a unit of ( $\mu\epsilon$ ) hoping to better enable the reader to judge the physical significance of the reported numbers. The square root of MSE is commonly known as the root MSE (RMSE).

Due to higher stiffness, the amplitude of NRB response is often smaller than that of TRB's. Hence, to better compare the performance of the model when evaluating TRB and NRB observations, we use normalized root mean squared error (nRMSE) to compare model errors:

$$nRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad \text{Eq. 5-11}$$

where  $y_{max}$  and  $y_{min}$  are the maximum and minimum values of the recorded target values respectively. In addition to enabling better comparison between NRB and TRB, using nRMSE allows for better comparison between observations with different amplitudes. This is due to the fact that larger amplitudes of response (which are caused by heavier trucks) would have higher RMSE values. Such higher RMSE values are not solely caused by poor model performance as the scale of the observation also plays a role in the RMSE value. Hence by normalizing the RMSE values by the amplitude of the response, the intent is to remove the scale effects and have an indicator that is more correlated to the performance of the forecasting model. For reporting the performance on a population of observations, the average of nRMSE values for all observations is reported.

Although MSE and average nRMSE values are the two main error metrics and performance indicators used within this work, it is worth mentioning that for time series forecasting applications, other metrics exist as well. One popular metric is mean absolute percentage error (MAPE) defined as:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y[i] - \hat{y}[i]}{y[i]} \right| \quad \text{Eq. 5-12}$$

as it is evident by the definition of MAPE, the forecasting error is normalized by the magnitude of the observation at each time step within that observation. As a result, it tends to penalize forecasting errors on time instances with smaller amplitudes more severely than errors on time instances with larger amplitudes. Consequently, an error on a time instance with a small amplitude response would have a higher impact on this metric compared to the same amount of error on a

time instance with larger amplitude response. However, an error on the peak response is more important than the error on a instance with small amplitude, such as that of bridge's free vibration. This fact makes MAPE not a suitable metric for this research and hence the metrics used within this research are limited only to the MSE and average nRMSE values.

#### **5.4 Results and Discussions**

Here, the results of training different models are presented. A common theme when presenting the results is to report the RMSE values for training the models on the training and validation datasets. Then, the hyper-parameters resulting in the best performance (i.e., lowest RMSE on the validation dataset) is implemented on the test dataset. In addition to RMSE values, average nRMSE values are also reported.

First, NRB's response is taken as input to the model to forecast TRB's response. Three different Seq2Seq models are trained, one for establishing forecasting model between static responses alone, the other for establishing forecasting relations between the dynamic responses alone and the final model is trained to forecast the actual response (i.e., sum of static and dynamic response). For the models trained on static and dynamic responses separately, the responses are combined and compared to the actual response.

The RMSE values for static and dynamic responses are presented in Table 5–2 and Table 5-3 respectively. The RMSE values after combining the results of the separately trained models are shown in Table 5-4. Furthermore, RMSE values for training the combined responses are shown in Table 5-5. It can be observed that the Seq2Seq model performs better when it is used to forecast the combined response compared to when used to forecast static and dynamic responses separately. The best models are achieved with  $b = 20$  and  $H = 64$ , and  $b = 20$  and  $H = 256$  for the GRU and LSTM cell types respectively. Note that for GRU and LSTM cell types, the best performances for training the models separately are achieved at a different batch size and hidden dimensions. For these models, the training and validation errors are plotted in Figure 5-7 as the training progresses.

Table 5-3. Dynamic response forecasting RMSEs for forecasting TRB using NRB.

		GRU Cell			LSTM Cell		
		H=32	H=64	H=128	H=32	H=64	H=128
Training	b=20	3.30	3.04	3.23	3.30	3.29	2.93
Errors	b=50	3.30	3.29	2.98	3.30	3.30	3.29

Table 5-2. Static response forecasting RMSEs for forecasting TRB using NRB.

		GRU Cell			LSTM Cell		
		H=32	H=64	H=128	H=32	H=64	H=128
Training	b=20	13.91	2.61	2.52	1.95	2.72	1.78
Errors	b=50	13.91	13.91	2.38	2.43	2.30	2.18
Validation	b=20	14.36	2.47	2.28	2.24	3.38	2.21
Errors	b=50	14.36	14.36	2.27	2.37	2.37	2.34

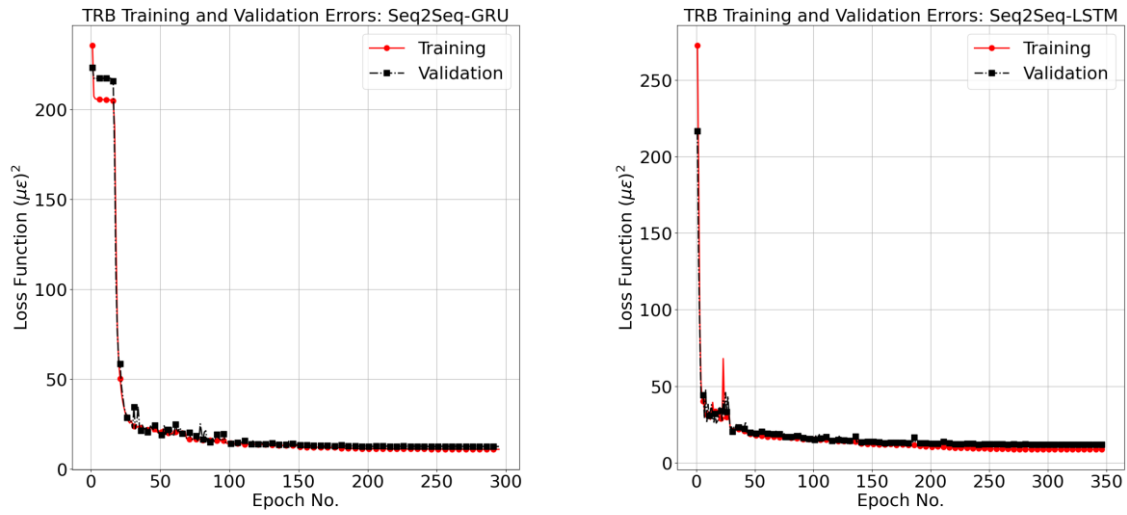


Figure 5-7. TRB forecasting training and validation errors (MSE) for the Seq2Seq model for GRU (left) and LSTM (right) cell types.

Validation	b=20	3.20	2.92	3.10	3.20	3.20	2.95
Errors	b=50	3.20	3.20	2.90	3.20	3.20	3.20

Table 5-4. Combined TRB forecasting RMSEs when static and dynamic components are forecasted separately.

		GRU Cell			LSTM Cell		
		H=32	H=64	H=128	H=32	H=64	H=128
Training	b=20	3.30	3.04	3.23	3.30	3.29	2.93
Errors	b=50	3.30	3.29	2.98	3.30	3.30	3.29
Validation	b=20	3.20	2.92	3.10	3.20	3.20	2.95
Errors	b=50	3.20	3.20	2.90	3.20	3.20	3.20

Table 5-5. Combined TRB forecasting RMSE when the actual signal is used without separating the static and dynamic components.

		GRU Cell	LSTM Cell



		H=32	H=64	H=128	H=32	H=64	H=128
Training	b=20	3.30	3.04	3.23	3.30	3.29	2.93
Errors	b=50	3.30	3.29	2.98	3.30	3.30	3.29
Validation	b=20	3.20	2.92	3.10	3.20	3.20	2.95
Errors	b=50	3.20	3.20	2.90	3.20	3.20	3.20

After observing that the Seq2Seq model achieves the highest performance when used to forecast the total response (rather than forecasting static and dynamic components individually), the baseline ARX model is trained only on the total response data. The best set of hyper-parameters for the ARX model where  $p = q = 10$  along with  $\lambda = 0.5$ . Additionally, for the regression-based baseline model, the best set of hyper-parameters were  $o_{NRB} = 8$  and  $o_{TRB} = 16$ .

Finally, the best models for the Seq2Seq models with GRU and LSTM cell types, along with the best baseline models were implemented on the test dataset. The results are shown in Table 5-6. As it is evident, the Seq2Seq models out-perform their linear counterparts.

Table 5-6. Model performance on the test dataset when forecasting TRB's response.

	Seq2Seq: GRU ( $b = 20, H = 64$ )	Seq2Seq: LSTM ( $b = 20, H = 256$ )	ARX: $p = q = 10,$ $\lambda = 0.5$	Regression-based Model $o_{NRB} = 8,$ $o_{TRB} = 16$
RMSE ( $\mu\epsilon$ )	3.10	3.06	6.40	5.56

To better visualize the performance of the models, the forecasted response is plotted for four sample test datasets, named samples 1 to 4 respectively. Figure 5-9 represents the plots



(a)



(b)



(c)



(d)

Figure 5-8. Sample trucks used to evaluate the performance of the Seq2Seq model. Figures a) to d) represent samples 1 to 4 respectively.

indicating the performance of the Seq2Seq model in forecasting TRB responses. The sample trucks were intentionally selected so as to represent a wide range of vehicle weights. Table 5-7 provides a description of the characteristics of the sample trucks used. Moreover, the sample trucks are shown in Figure 5-8.

Table 5-7. Description of properties for sample trucks used.

	Sample 1	Sample 2	Sample 3	Sample 4
Weight (kips)	25	47	75	152
FHWA Class	5	9	9	13
No. Axles	2	5	5	11

Next, the direction forecasting is flipped and TRB responses are used to forecast NRB responses. Similar to the previous problem setting, three different Seq2Seq models are trained to

predict the static, dynamic and total responses respectively. Similar to the previous case of forecasting TRB, the models trained to forecast the combined response outperform those aimed at forecasting response individually. The results for forecasting the response are shown in Table 5-8. Similarly, the evolution of the loss function throughout training for the best models is shown in Figure 5-10.

It can be observed that for the GRU cell type, the model with  $b=20$  and  $H=256$  results in the highest performance on the validation dataset. For the LSTM cell type, the model with  $b=50$  and  $H=128$  results in the highest performance on the validation dataset. Additionally, the ARX model is implemented to forecast NRB's response in a similar manner.

The highest performance was achieved with  $p=q=65$  and  $\lambda=5$ . Additionally, for the regression-based baseline model, the highest performance was achieved with  $\sigma_{TRB} = 4$  and  $\sigma_{NRB} = 8$ . The highest performing models were implemented on the test dataset and the results are indicated in Table 5-9. As it can be seen, the Seq2Seq model with LSTM cell type achieves the highest performance on the test dataset.

Table 5-8. Combined forecasting results for using the response of TRB to forecast the response of NRB.

		GRU Cell			LSTM Cell		
		H=32	H=64	H=128	H=32	H=64	H=128
Training	b=20	3.35	2.53	3.09	3.02	3.21	3.11
Errors	b=50	9.72	3.17	3.08	3.37	2.57	3.27
Validation	b=20	2.89	2.63	2.59	2.78	2.77	2.89
Errors	b=50	9.85	2.61	2.61	3.09	2.43	2.91

Table 5-9. Model performance on the test dataset when forecasting NRB's response.

	Seq2Seq: GRU $(b = 20, H = 256)$	Seq2Seq: LSTM $(b = 20, H = 128)$	ARX: $p = q = 65,$ $\lambda = 5$	Regression-based Model $\sigma_{TRB} = 4,$ $\sigma_{NRB} = 8$
RMSE ( $\mu\epsilon$ )	2.59	2.51	3.31	5.77

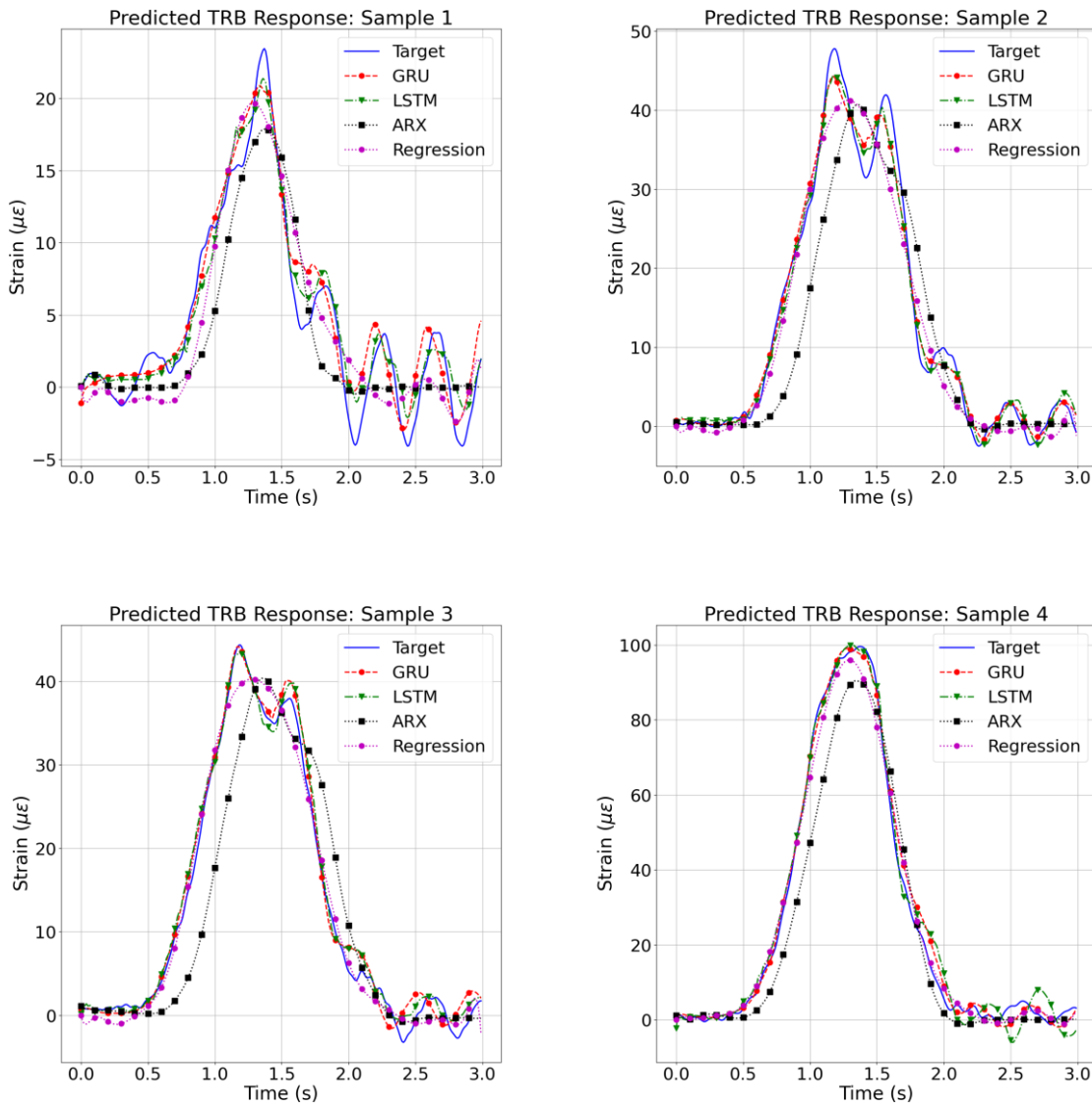


Figure 5-9. TRB forecasting results for 4 sample observations.

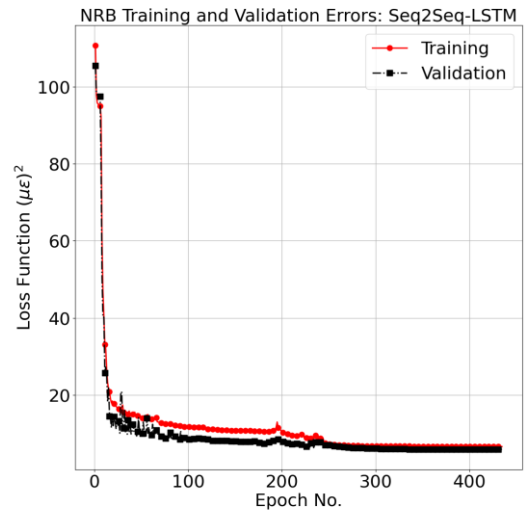
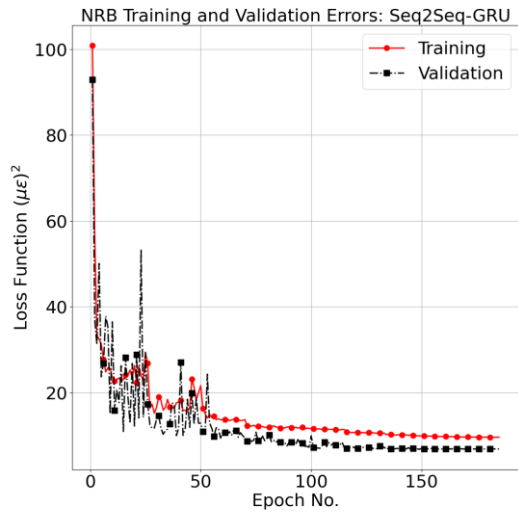


Figure 5-10. NRB forecasting training and validation errors (MSE) for the Seq2Seq model for GRU (left) and LSTM (right) cell types.

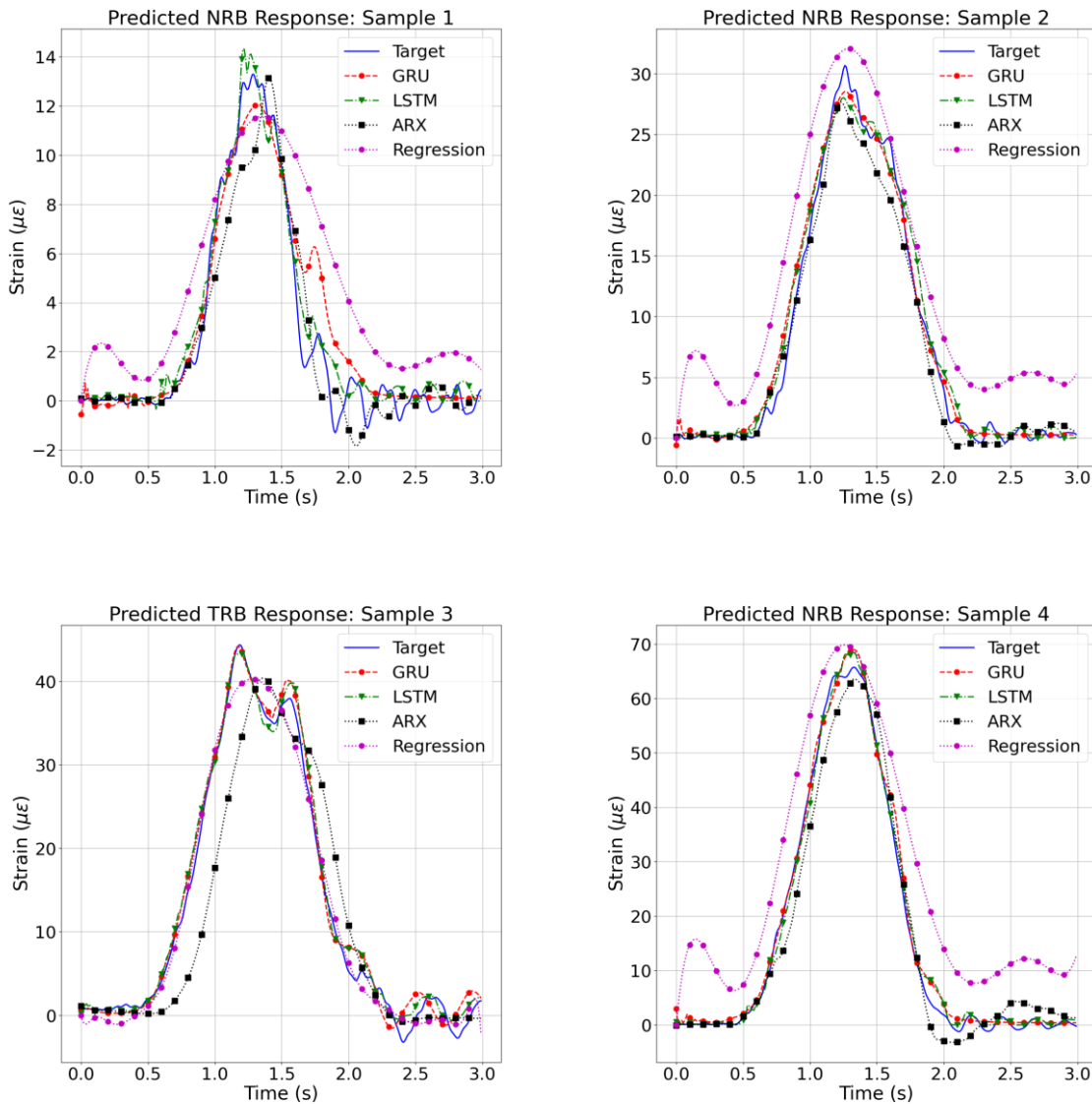


Figure 5-11. NRB forecasting results for 4 sample observations.

Thus far, the forecasting problem was solved without considering the effects of the types of trucks that are traversing the bridges. After this point, the focus of the research is to examine the effect of vehicle types on the performance of the forecasting models. An interesting point to keep in mind, is that the ratio of trucks traversing the bridges (and hence sampled within this

dataset), is not uniform across all truck type. In fact, certain types of trucks are more common and certain types are rarer. Figure 5-12 shows the frequency of trucks used within this dataset for different vehicle classes and axle numbers. By vehicle class, it is referred to the standard truck classifications introduced by the Federal Highway Administration (FHWA). For the purpose of self-sufficiency and for the better visualization of vehicle classification, the original image describing vehicle classification by FHWA is provided in Figure 5-13. As it can be seen, five axle trucks are the most common class of trucks observed. Additionally, trucks from FHWA class 9 are

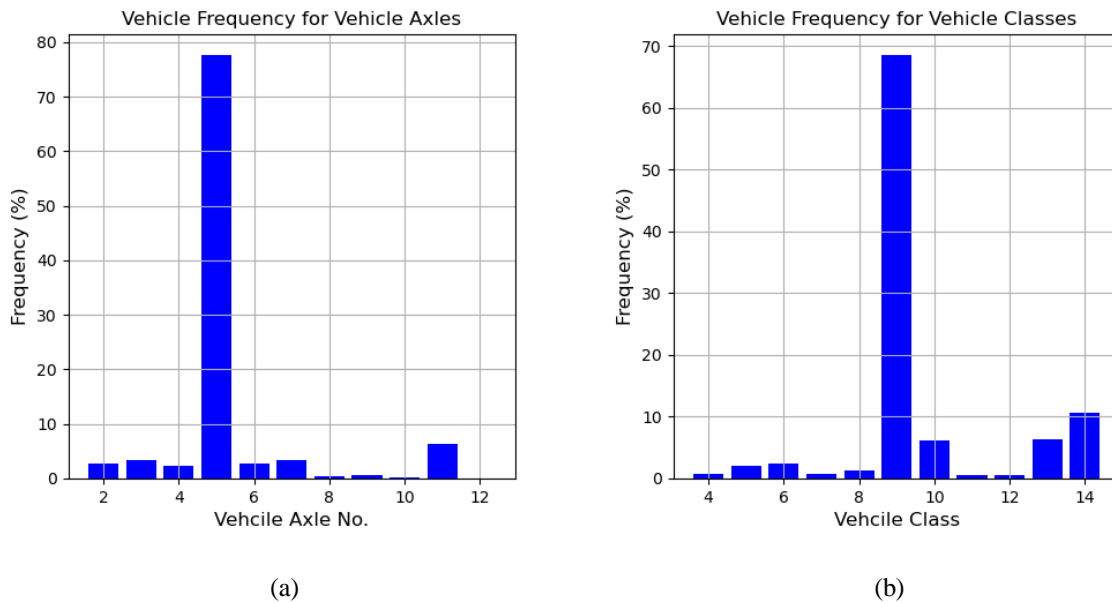


Figure 5-12. Truck frequency in the dataset for a) different axle numbers and b) different FHWA vehicle classes.

the most common class of trucks observed. FHWA class 9 trucks correspond to the standard five axle single trailer trucks.

The main EOCs affecting the performance of the Seq2Seq model is the type of load that is traversing through the bridge. Using FHWA vehicle class to group observations together is a more effective manner of classification as trucks from one FHWA class tend to have more similarities compared to trucks with the same axle number. Hence, to explore the relationship between the Seq2Seq model performance and load type, the effects of vehicle class type on model performance







































<b>Class 1</b> Motorcycles		<b>Class 7</b> Four or more axle, single unit	
<b>Class 2</b> Passenger cars		<b>Class 8</b> Four or less axle, single trailer	
			
			
			
<b>Class 3</b> Four tire, single unit		<b>Class 9</b> 5-Axle tractor semitrailer	
			
			
<b>Class 4</b> Buses		<b>Class 10</b> Six or more axle, single trailer	
			
			
<b>Class 5</b> Two axle, six tire, single unit		<b>Class 11</b> Five or less axle, multi trailer	
			
			
<b>Class 6</b> Three axle, single unit		<b>Class 12</b> Six axle, multi-trailer	
			
			
			

Figure 5-13. Description of highway vehicle classification by FHWA (image source: FHWA).

is evaluated. The distribution of model error against different types of vehicle classes is indicated in Figure 5-14.

Since the distribution of truck properties is heavily skewed towards one axle count and one FHWA vehicle class, to explore the effects of the type of vehicle load on the model performance, the focus is given to trucks from class 9 alone.

Trucks from class 9 are separated to generate a new datasets with trucks from class 9 alone. The dataset consisted of a total of 1333 truck observations. From this dataset, 900 were used for training, 100 for validation and 333 for testing.

For this case, focus is given to forecasting TRB's response. Similar to the previous case of trucks from all classes, Seq2Seq models are trained with GRU and LSTM cell types. Since training

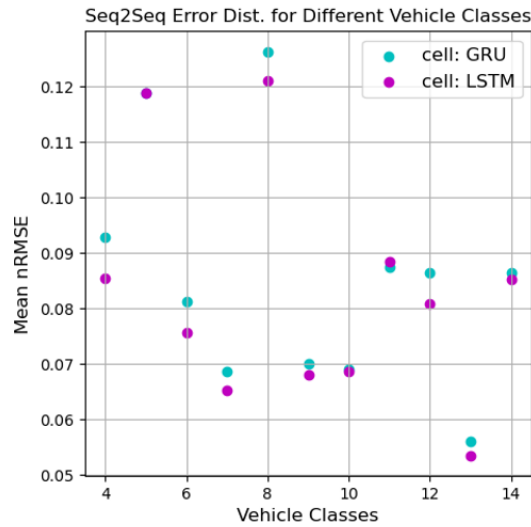


Figure 5-14. Seq2Seq model performance for different truck classes.

for the total response proved to be the most effective version, the models are trained on the total response. The average nRMSE for class 9 is compared between the two cases in Table 5-10.

Table 5-10. Average nRMSE values for class 9 trucks for models trained with the entire data and data from class 9 only.

Type of Data	GRU Cell		LSTM Cell	
	Entire Data	Class 9	Entire Data	Class 9
Average nRMSE	7.0%	6.3%	6.9%	6.2%

As it can be observed, training models for one class of vehicle alone increases the performance of the Seq2Seq model. This increase is observed for both the GRU and LSTM cell types. This fact suggests that FHWA class impacts the performance of the Seq2Seq model and by focusing on solely one truck class, the performance of the Seq2Seq model can be increased.

Note that due to scarcity of the data from other vehicle classes, it is difficult to train the Seq2Seq model for those classes separately. The model is expected to overfit the limited training data resulting in low training error but high validation and test errors. Consequently, to make

statements for other types of truck classes, and also on the optimal manner to normalize the data with respect to EOCs, more data from different truck classes and a wide range of EOCs is needed.

## **5.5 Conclusion**

Two different time series forecasting frameworks were developed for joint bridge response forecasting in the I-275 highway corridor. The first framework used a Seq2Seq architecture and the second framework was developed using the ARX model. Both frameworks were implemented on an experimental dataset with 2558 pairs of trucks responses. For the Seq2Seq framework, two different RNN cell types, namely GRU and LSTM, were used. Two different directions of forecasting was explore, namely using NRB's response to forecast TRB's and vice versa.

In both directions of forecasting, the Seq2Seq models outperformed the ARX model with a significant margin. Additionally, it was observed that LSTM cell type would result in a higher forecasting performance compared to the GRU cell type for both directions of forecasting, although the increased performance was much smaller compared to the gap with the ARX model indicating the superiority of the Seq2Seq model.

Finally, the Seq2Seq model was trained on the most common type of truck in the dataset, which were trucks from class 9 and it was observed that the performance of the Seq2Seq model increased when focusing on one class of vehicles suggesting that proper EOCs data normalization would increase the performance of the forecasting framework.

## **Chapter 6.**

### **Conclusion and Future Work**

#### **6.1 Summary**

The overarching goal of this dissertation was to advance data-driven techniques for structural monitoring, performance assessment, and damage detection by explicitly considering the environmental and operational conditions (EOCs) that influence structural behavior. The specific focus on EOCs was motivated by the large role these external factors play in infrastructure system (IS) assets. For example, measurement of wind turbine structural responses is driven by the complex interactions that exist between the wind that drive the energy generation function of the mechanical turbine, and between the turbine in the nacelle and the turbine tower. Similar, for bridges monitored with sensors, their responses are largely governed by large traffic loads that dynamically couple with the dynamics of the bridge. In all of these cases, the EOCs are often not directly measured (e.g., traffic loads on bridges) or measured in indirect ways (e.g., use of a meteorological tower in relative proximity to the turbine). Nonetheless, understanding the influence of EOC on structural responses measured by structural monitoring is essential to ensuring data-driven algorithms can accomplish their mission with accuracy and precision. Specifically, what constitutes normal (healthy) behavior as a baseline must be considered in the context of the EOC which will influence these baselines.

The thesis has three primary objectives. First, a framework for extracting EOC sensitive features, or EOCSFs, from structural response data was developed and validated on a wind turbine

dataset. In structures like wind turbines, it provides a basis for indirectly measuring EOCs without the need for EOC-specific sensors. Using EOCSFs, a novel data normalization approach was proposed for data-driven structural health monitoring based on unsupervised clustering of observation in the EOC space during training. A soft assignment approach is taken during testing to assign observations to more than one cluster with cluster assignment weighted by the probability of the observation belonging to the cluster. The result is a more efficient data normalization strategy that enhances the overall performance of damage detection methods based on EOCSFs combined with damage sensitive features (DSFs). The work is further enhanced by considering the sequential relationship between observations over time using Hidden Markov Models (HMM) that do not rely on assuming independent or identically distributed observations. The second objective was to consider the use of structural response data to estimate the loads of the EOC space for highway bridges using machine learning (ML) methods. Using encoder models, a data driven approach to training a model between bridge responses to trucks of known weight parameters (e.g., gross vehicular weight, axle weights) is explored to develop a bridge weigh in motion (BWIM) system capable to estimating these weight parameters using its response. The third objective builds on the BWIM work to develop sequence-to-sequence (Seq2Seq) time series model to use one bridge response to predict that of another. Again, an EOC data normalization strategy is taken to enhance the predictive capabilities of these models. A Seq2Seq model has potential use in predicting bridge responses to control connected vehicles (e.g., by prescribing speed or lane to minimize dynamic amplifications of bridges) and to track long-term deterioration at bridges not monitored using the response of another bridge in the same corridor that is monitored.

### **6.1.1 Key Intellectual Contributions**

In Chapter 2 of this dissertation, two major contributions to the body of knowledge of the SHM field were made. First and foremost, an approach for extracting EOCSFs was introduced. The process proposed consists of four key steps: first, developing a mechanical model for the structure of interest to model the response of the structure under a diverse set of EOCs (without requiring experimental setups to do so); second, use mechanical intuition to propose potential EOCSFs; third, model structural responses under varying EOCs to regress EOC-response relationships that will establish potential EOCSFs; forth, explore the robustness of EOCSF under structural damage scenarios to assess their insensitivity to damage states. The rationale behind the last step is to ensure that when using EOCSFs for data normalization, the EOCs normalization phase is not affected by the presence of structural damage. The final set of EOCSFs, because they are derived from structural response data, will often be collected at data rates identical to the response data used to extract DSFs. This gives them even greater utility during data normalization. To validate the ability to extract EOCSFs from response data, a small-scale wind turbine system located at LANL was used as a testbed study. For this specific structure, two key EOCs were considered to have the most influence on the behavior of the turbine structure: wind speed (which correlates to rotor angular velocities) and wind direction (which correlate to nacelle yaw angle). Two EOCSFs were hypothesized and validated to be sensitive to the EOC parameter sought but insensitive to damage. Specifically, the frequencies of transient peaks manifest in the frequency domain of horizontal acceleration time histories were shown to correlate almost linearly with rotor angular velocity while the ratio of acceleration energy between two orthogonal horizontal directions was linearly related to the nacelle yaw angle. The effectiveness of these EOCSFs for data normalization were also proved using both simulated and experimental field data. Whilst these methods are

implemented and tested on a small scale wind turbine, the concept of EOCSFs are applicable to many other structures where global responses are strongly correlated to EOC parameters.

Also in Chapter 2, a modular, three tier damage detection framework trained in an unsupervised fashion was proposed. The framework specifically relies on clustering observations of a structural system to normalized the observations by EOC. Previously, such frameworks relied on clustering EOCs in their EOC state space to define clusters to which observations of the structure in an unknown state would be compared. To normalize, EOCs are used to make a hard, single-cluster assignment. This approach has two drawbacks: first, it assumes perfect certainty in the cluster assignment and second, the unknown state is assessed using a small subset of the training data available. This dissertation introduces a novel approach to assignment by performing soft assignment where an observation is assigned to more than one cluster with the influence of the assigned cluster weighted by the probability of assignment. In addition to acknowledging less than perfect certainty in a cluster assignment, it also allows a much larger set of training data to be used as a baseline for comparative purposes. The proposed soft assignment framework utilized GMMs to assign probabilities of belonging to each of the EOCs clusters with a comparison of each testing observation made to the entire training dataset. Ultimately, using the law of total probability, the probability of observing a certain observation (with a given structural response under a specific set of EOCs) was computed with decisions on the state of the structure (i.e., health or damaged) made by this probability. The performance of the soft assignment data normalization methodology was compared to the traditional hard assignment approach; health assessments where soft assignment was used outperformed the hard assignment approach both on the simulated and experimental data proving the effectiveness of this approach.

In Chapter 3, the i.i.d. assumption of existing ML-based technique was relaxed by considering sequences of structural response data under varying EOCs for damage detection. Specifically, HMMs were used to assign probabilities to each sequence and sequences with a significantly lower likelihood compared to the sequences of data observed in training stage (i.e., sequences from the undamaged structure) are labeled as damaged. This proposed framework is tested on the Z-24 dataset, one of the fewest (if not the only) real-world bridge response datasets with actual physical damage introduced in a controlled manner in the structure. The proposed method where i.i.d. assumptions were not made were compared to two baseline damage detection methods which assume that incoming observations are i.i.d.; the proposed approach consistently outperformed the baseline methods. This provided strong evidence that there exists valuable information within sequences of structural observations and sequential damage detection frameworks can be extremely useful for real-world applications. Furthermore, analysis was conducted on the impact of the lengths of the sequences considered. It was observed that the longer the length of the sequence, the higher the performance of the HMM-based damage detection framework (although this outcome must not be generalized given the fact that this framework is tested only on one dataset and example structure).

The innovation of soft assignment during EOC data normalization and considering sequences of system observation without assuming i.i.d., the thesis makes a major contribution to the field of data-driven SHM by relaxing two of the most pressing limiting assumptions commonly used. Beyond relaxing these assumptions, the key point that is made by developing these methods in Chapters 2 and 3, is the fact that when utilizing a general purpose ML-based framework for structural damage detection, the end-user must understand the underlying assumptions of the methods used and be aware of the potential limitations those assumptions impose on the damage



detection framework. This work has attempted to set an example for doing so by clearly articulating these limiting assumptions and the steps taken to overcome these assumptions.

Chapters 4 and 5 of this dissertation begin the transition from damage detection to structural assessment for management and decision making. In chapter 4, an encoder with bi-directional LSTM cell is used to estimate truck weightage properties from highway bridge response. The importance of solving this problem was fully discussed in the introduction chapter as well as in chapter 4. Effectively, truck axle weight distribution constitutes a large portion of the EOCs dominating the force response of a bridge structure. Truck weight distribution is the main operational condition influencing the response of the bridge and hence in effect, this problem is extremely similar to extracting EOCSFs from structural response data. In contrast to the process utilized in chapter 2, the DL-based enables the extraction of this information from the structural response with minimal effort required for feature engineering. This chapter highlights the capability of non-linear DL-based models to learn complex patterns from the data with minimal interventions needed. Similar to the process in chapter 2, a mechanical model is also developed to simulate bridge response to highway traffic under the entire EOC space, though the use of the encoder for BWIM problem on experimental data would be less dependent on this model compared to the case of chapter 2 where the model was fully utilized in the feature selection process.

Finally, following successful utilization of an Encoder in chapter 4, an Encoder-Decoder model with Seq2Seq architecture was used to forecast the response of a downstream bridge to a vehicle load given the response of an upstream bridge to the same load. Two cell types, namely LSTM and GRU, were explored as the RNN cells used within the Seq2Seq model. Furthermore, two baseline models based on traditional time series forecasting tools, i.e., an ARX model and a regression-based model, were developed for the purpose of bridge response prediction. It was

observed that the Seq2Seq model can accurately forecast bridge responses. The baseline models underperformed the Seq2Seq model, but still managed to provide helpful forecasting results. As it was pointed out, due to the unique nature of the problem, there were stark conceptual differences between the baseline models and the Seq2Seq model which was ideally suitable for this problem setting. The suitable structure of the Seq2Seq model for this problem must be interpreted as a strength of the Seq2Seq model.

A key component of this chapter was the set of signal pre-processing steps taken to enable the successful implementation of the time series forecasting models. The backbone of all these steps was the intuition and the domain-specific knowledge of the mechanics of the bridge response to the traversing truck load. Specifically, three main steps were taken: 1- the length of each response was selected so as to contain both the forced response of the bridge to the load, as well as enough free vibration cycles following the trucks passage of the bridge. The bridge response sequences were manually processed to isolate and select truck responses that are far enough from other loads that ensure the response sequence to the current truck is not impacted by previous and next vehicles. 2- the free response of the TRB bridge was removed to ensure the previous loads do not adversely impact the performance of the time series forecasting models. 3- The response of the bridges was decomposed into static and dynamic components to evaluate the performance of the Seq2Seq model when forecasting these components. Although it was ultimately discovered that the performance of the model was higher when the bridge response was not separated, this separation provided valuable insights into the performance of time series models for future applications. Specifically, it was observed that using the response of the bridge with more complicated dynamics (i.e., NRB's response, that had two major modal frequencies) to forecast the response of the simpler bridge (i.e., TRB's response, that had one major modal frequency)

proved to yield more accurate forecasting results. As expected, the performance of the Seq2Seq model when forecasting the static component of the response was independent of the relative complexity of the dynamics of the input and output bridges. Since the static portion of the response dominated the overall response of the bridge, the performance of forecasting in either directions, i.e., from NRB to TRB (or the bridge with complex dynamics to the bridge with simpler dynamics) or from TRB to NRB (or the bridge with simpler dynamics to the bridge with more complicated dynamics) was similar, highlighting the robustness of the overall forecasting framework to the relative dynamic complexity of the two bridges.

Lastly, it was highlighted that the EOC space of the problem is not uniform. A separate Seq2Seq model was trained for the most common class of truck loads (i.e., FHWA class 9) and it was observed that this would increase the performance of the Seq2Seq model. This observation effectively highlights the importance of EOCs data normalization in the application of data-driven models for structural assessment applications.

The final point that is worth mentioning, is the evolution of data-driven techniques in this dissertation. Initially, chapter 2 began with shallow ML-based models and the models became deeper as the dissertation proceeded. The reason behind this evolution was the adoption of ML and DL-based methods by the field during the timespan of the development of this dissertation. As researchers and practitioners recognized the benefits of the AI-based methods and embraced it for structural engineering applications, this dissertation attempted to be at the forefront of this application hoping to enable the implementation and further advancement of AI-based methods for IS assessment under complex set of loads that impact these systems during the span of their normal operation.

## 6.2 Future Work

### 6.2.1 Research Questions

The major question to be answered following the research presented, is to quantify the financial gains from implementing improved ML-based methods for the proposed application. The norm in the ML and DL literature, is to compare the performance of the ML-based techniques to other models based on comparing certain performance metrics. Similarly, in this dissertation for example AUC was used in chapters 2 and 3 and the RMSE and average nRMSE were used in chapter 5 for comparing the performance of the proposed models. The models showed improvements in the order of 5-15% in terms of the metrics. Needless to say, that as obvious from the ML-literature a consistent 5% over-performance is extremely valuable. Especially given the fact that whence the computational intensive alternatives are developed and trained, implementation of the models is actually quite inexpensive. Another key item to keep in mind is that use of the proposed models is to only extract insights and patterns that are already present in the data. A large difference between baseline results and the proposed method could indicate a wrong or sub-optimal implementation of the baseline methods. The key point to keep in mind, is that from the perspective of stakeholders, the key question to answer is not how much a system outperforms others in terms of metrics used in the ML community, but rather, how much gain would the model have for the society. As such, it is extremely needed to utilize a systematic framework for quantifying financial gains from implementing better models compared to existing IS management and maintenance practices. This might not be as critical for ML-based models since the implementation of models is not significantly costly, but it could be extremely important for scenarios where two different hardware instrumentations are needed and one could result in higher performance on the quantitative metrics by a known threshold.

### **6.2.2 Broader Impact**

By solving pressing challenges in the application of ML-based methods for structural assessment applications, this dissertation fills an important gap in the path that the field of SHM and IS maintenance is expanding in. As such, this dissertation sets the stage for transformative changes to IS management practices and enables IS stakeholders to ensure the existing systems can meet the increasing demand on these systems and they continue to age.

First, is using these methods for IS finance. The methods can set a great foundation for transitioning from the existing IS financing models to the more sustainable usage-based models that aim to charge each IS user based on the amount that user consumes the infrastructure. A low-hanging fruit is using these models for developing usage-based fee structures for over-weight/over-sized trucks traveling in the highway network. Another approach is to use for PPP contracts and sharing costs amongst different stakeholders. Finally, IS maintenance can be connected to IS finance by incorporating the already existing data-driven models for credit ratings that affect the interest rates that is set on bonds issued by the IS owners and hence encourage for better maintenance.

By adopting innovative and more sustainable financing models, the data generated by the IS can be priced and hence traded. Consequently, research must be conducted to determine the value of information that the data generated by the IS has. For example, as observed in chapter 5, class 11 trucks are less common compared to class 9 trucks and hence recording the response from those trucks could be more informative and hence more valuable. Thus, it is worth the effort to quantify the value of information that these data contain for undertaking best strategies in IS instrumentation and data collection. Nonetheless, the possibility of selling IS data is itself a major

incentive for IS owners for instrumenting these systems which enables better monitoring and hence better maintenance of the IS assets.

## Bibliography

- A. P., Dempster; N. M., L. D. B. R. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1), 1–38. <https://doi.org/10.1115/1.3424485>
- A Comprehensive Assessment of America's Infrastructure*. (2021).
- Abdel-Ghaffar, A., & Scalan, R. (1985). Ambient vibration studies of Golden Gate bridgeL I. Suspended Structure. *Journal of Engineering Mechanics*, 111(4), 463–482.
- Admassu, K., Lynch, J., Athanasopoulos-Zekkos, A., & Zekkos, D. (2019). Long-term wireless monitoring solution for the risk management of highway retaining walls. *Proc. SPIE 10971, Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure and Transportation*, 1097103(April 2019), 1. <https://doi.org/10.1117/12.2516081>
- Agdas, D., Rice, J. A., Martinez, J. R., & Lasa, I. R. (2016). Comparison of Visual Inspection and Structural-Health Monitoring As Bridge Condition Assessment Methods. *Journal of Performance of Constructed Facilities*, 30(3). [https://doi.org/10.1061/\(asce\)cf.1943-5509.0000802](https://doi.org/10.1061/(asce)cf.1943-5509.0000802)
- Agrawal, A. K., Washer, G., Alampalli, S., Gong, X., & Cao, R. (2021). Evaluation of the Consistency of Bridge Inspection Ratings in New York State. *Journal of Infrastructure Systems*, 27(3), 04021016. [https://doi.org/10.1061/\(asce\)is.1943-555x.0000622](https://doi.org/10.1061/(asce)is.1943-555x.0000622)
- Akobeng, A. K. (2007). Understanding diagnostic tests 3: Receiver operating characteristic curves. *Acta Paediatrica, International Journal of Paediatrics*, 96(5), 644–647. <https://doi.org/10.1111/j.1651-2227.2006.00178.x>
- Ashuri, B., Kashani, H., Molenaar, K. R., Lee, S., & Lu, J. (2012). Risk-Neutral Pricing Approach for Evaluating BOT Highway Projects with Government Minimum Revenue Guarantee Options. *Journal of Construction Engineering and Management*, 138(4), 545–557. [https://doi.org/10.1061/\(asce\)co.1943-7862.0000447](https://doi.org/10.1061/(asce)co.1943-7862.0000447)
- ASTM. (2017). *E1318-09. standard specification for highway weigh-in-motion (wim) systems with user requirements and test methods*.
- Baek, S.-J., Park, A., Ahn, Y.-J., & Choo, J. (2015). Baseline correction using asymmetrically reweighted penalized least squares smoothing. *The Analyst*, 140(1), 250–257. <https://doi.org/10.1039/c4an01061b>
- Bahrami, O.; Tsiapoki, S.; Kane, M. B.; Lynch, J. P.; Rolfes, R. (2017). Extraction of Environmental and Operational Conditions of Wind Turbines using Tower Response Data for Structural Health Monitoring. *Proceedings of Structural Health Monitoring*, 1689–1699.
- Bahrami, O., Hou, R., Wang, W., & Lynch, J. P. (2021). Time series forecasting to jointly model bridge responses. *Bridge Maintenance, Safety, Management, Life-Cycle Sustainability and Innovations*, 299–307. <https://doi.org/10.1201/9780429279119-37>
- Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1972). A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains Author ( s ): Leonard E . Baum , Ted Petrie , George Soules , Norman Weiss Source : The Annals of Mathematical

- Statistics , Vol . 41 , No . 1 ( Feb. *Institute of Mathematical Statistics Stable*, 41(1), 164–171.
- Bedon, C., Bergamo, E., Izzi, M., & Noè, S. (2018). Prototyping and validation of MEMS accelerometers for structural health monitoring—the case study of the Pietratagliata cable-stayed bridge. *Journal of Sensor and Actuator Networks*, 7(3). <https://doi.org/10.3390/jsan7030030>
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. In *Springer*.
- Box, A. G. E. P., & Box, B. Y. G. E. P. (2011). Biometrika Trust A General Distribution Theory for a Class of Likelihood Criteria. *Biometrika*, 36(3), 317–346.
- Byon, E., & Ding, Y. (2010). Season-dependent condition-based maintenance for a wind turbine using a partially observed markov decision process. *IEEE Transactions on Power Systems*, 25(4), 1823–1834. <https://doi.org/10.1109/TPWRS.2010.2043269>
- Chatterjee, P., O'Brien, E., Li, Y., & González, A. (2006). Wavelet domain analysis for identification of vehicle axles from bridge measurements. *Computers and Structures*, 84(28), 1792–1801.
- Chaudhari, T. D., & Maiti, S. K. (1999). Crack detection in geometrically segmented beams. *Damage Assessment of Structures, Proceedings of the International Conference on Damage Assessment of Structures (DAMAS 99), Dublin, Ireland*, 343–353. <https://doi.org/10.4028/www.scientific.net/kem.167-168.343>
- Chipka, J. B.; Lisicki, A. R.; Nguyen, C. T; Taylor, S. G.; Ammerman, C. N; Farrar, C. R. . (2013). Experimental Characterization and Predictive Modeling of a Residential-Scale Wind Turbine. *31 St Conference Proceedings of the Society for Experimental Mechanics Series*, 39(4). <https://doi.org/10.1007/978-1-4614-6555-3>
- Ciang, C. C., Lee, J. R., & Bang, H. J. (2008). Structural health monitoring for a wind turbine system: A review of damage detection methods. *Measurement Science and Technology*, 19(12). <https://doi.org/10.1088/0957-0233/19/12/122001>
- Copeland, K. A. F. (1997). Applied Linear Statistical Models. In *Journal of Quality Technology* (Vol. 29, Issue 2). <https://doi.org/10.1080/00224065.1997.11979760>
- CSMIP. (2021). *California Geological Survey - About CSMIP (California Strong Motion Instrumentation Program)*. Department of Conservation, State of California. <https://www.conservation.ca.gov/cgs/s mip>
- Ding, Y., Kong, S., Huang, J., & Xie, X. (2010). FEM analysis of vehicle-bridge vibration considering bridge deck's vibration. *2010 International Conference on Mechanic Automation and Control Engineering, MACE2010*, 1, 788–791. <https://doi.org/10.1109/MACE.2010.5535836>
- Eggimann, S., Mutzner, L., Wani, O., Schneider, M. Y., Spuhler, D., Moy De Vitry, M., Beutler, P., & Maurer, M. (2017). The Potential of Knowing More: A Review of Data-Driven Urban Water Management. *Environmental Science and Technology*, 51(5), 2538–2553. <https://doi.org/10.1021/acs.est.6b04267>
- Elhatab, A., Uddin, N., & O'Brien, E. (2016). Drive-by bridge damage monitoring using Bridge Displacement Profile Difference. *Journal of Civil Structural Health Monitoring*, 6(5), 839–850. <https://doi.org/10.1007/s13349-016-0203-6>
- Espinoza, D., Rojo, J., Cifuentes, A., & Morris, J. (2020). DNPV: a valuation methodology for infrastructure and Capital investments consistent with prospect theory. *Construction Management and Economics*, 38(3), 259–274. <https://doi.org/10.1080/01446193.2019.1648842>
- Evans, J. R., Allen, R. M., Chung, A. I., Cochran, E. S., Guy, R., Hellweg, M., & Lawrence, J. F.



- (2014). Performance of several low-cost accelerometers. *Seismological Research Letters*, 85(1), 147–158. <https://doi.org/10.1785/0220130091>
- Farrar, C. R., & Worden, K. (2012). Structural Health Monitoring: A Machine Learning Perspective. In *Structural Health Monitoring: A Machine Learning Perspective*. <https://doi.org/10.1002/9781118443118>
- FHWA. (2001). *Reliability of Visual Inspection for Highway Bridges*. <https://www.fhwa.dot.gov/publications/research/nde/01020.cfm>
- Figueiredo, Eloi; Radu, Lucian; Worden, Keith; Farrar, C. R. (2014). A Bayesian approach based on a Markov-chain Monte Carlo method for damage detection under unknown sources of variability. *Engineering Structures*, 80, 1–10.
- Fugro. (2021). *Fugro Offshore Structural Monitoring*.
- García Márquez, F. P., Tobias, A. M., Pinar Pérez, J. M., & Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46, 169–178. <https://doi.org/10.1016/j.renene.2012.03.003>
- González, A., Rattigan, P., OBrien, E. J., & Caprani, C. (2008). Determination of bridge lifetime dynamic amplification factor using finite element analysis of critical loading scenarios. *Engineering Structures*, 30(9), 2330–2337. <https://doi.org/10.1016/j.engstruct.2008.01.017>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>
- Görnitz, N., Braun, M., & Kloft, M. (2015). Hidden Markov anomaly detection. *32nd International Conference on Machine Learning, ICML 2015*, 3, 1833–1842.
- Gunes, V., Peter, S., Givargis, T., & Vahid, F. (2014). A survey on concepts, applications, and challenges in cyber-physical systems. *KSII Transactions on Internet and Information Systems*, 8(12), 4242–4268. <https://doi.org/10.3837/tiis.2014.12.001>
- Gungor, O. E., Al-Qadi, I. L., & Mann, J. (2018). Detect and charge: Machine learning based fully data-driven framework for computing overweight vehicle fee for bridges. *Automation in Construction*, 96(September), 200–210. <https://doi.org/10.1016/j.autcon.2018.09.007>
- Gungor, O. E., Petit, A. M. A., Qiu, J., Zhao, J., Meidani, H., Wang, H., Ouyang, Y., Al-Qadi, I. L., & Mann, J. (2019). Development of an overweight vehicle permit fee structure for Illinois. *Transport Policy*, 82(June), 26–35. <https://doi.org/10.1016/j.tranpol.2019.08.002>
- Häckell, Moritz W.; Rolfes, R. (2013). Monitoring a 5MW offshore wind energy converter- Condition parameters and triangulation based extraction of modal parameters. *Mechanical Systems and Signal Processing*, 40, 322–343.
- Häckell, M. W. (2015). *A Holistic Evaluation Concept for Long-Term Structural Health Monitoring*. Leibniz Universität Hannover.
- Hackell, M. W., Rolfes, R., Kane, M. B., & Lynch, J. P. (2016). Three-Tier Modular Structural Health Monitoring Framework Using Environmental and Operational Condition Clustering for Data Normalization: Validation on an Operational Wind Turbine System. *Proceedings of the IEEE*, 104(8), 1632–1646. <https://doi.org/10.1109/JPROC.2016.2566602>
- He, W., Deng, L., Shi, H., Cai, C., & Yu, Y. (2017). Novel virtual simply supported beam method for detecting the speed and axles of moving vehicles on bridges. *Journal of Bridge Engineering*, 22(4).
- He, W., Ling, T., OBrien, E. J., & Deng, L. (2019). Virtual axle method for bridge weigh-in-motion systems requiring no axle detector. *Journal of Bridge Engineering*, 24(9).
- Hou, R. (2020). *Bridge Structural Health Monitoring Using a Cyber-Physical System Framework*. University of Michigan.

- Hou, R., Jeong, S., Law, K. H., & Lynch, J. P. (2019). Reidentification of trucks in highway corridors using convolutional neural networks to link truck weights to bridge responses. *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, March 2019*, 27. <https://doi.org/10.1117/12.2515617>
- Inaudi, D., & Glisic, B. (2010). Long-range pipeline monitoring by distributed fiber optic sensing. *Journal of Pressure Vessel Technology, Transactions of the ASME*, 132(1), 0117011–0117019. <https://doi.org/10.1115/1.3062942>
- Jang, S., Jo, H., Cho, S., Mechitov, K., Rice, J. A., Sim, S. H., Jung, H. J., Yun, C. B., Spencer, B. F., & Agha, G. (2010). Structural health monitoring of a cable-stayed bridge using smart sensor technology: Deployment and evaluation. *Smart Structures and Systems*, 6(5–6), 439–459. [https://doi.org/10.12989/sss.2010.6.5\\_6.439](https://doi.org/10.12989/sss.2010.6.5_6.439)
- Johnson, N. R., Lynch, J. P., & Collette, M. D. (2018). Response and fatigue assessment of high speed aluminium hulls using short-term wireless hull monitoring. *Structure and Infrastructure Engineering*, 14(5), 634–651. <https://doi.org/10.1080/15732479.2017.1380676>
- Jonkman, J. M.; Buhl Jr., M. L. (2005). FAST User's Guide. In *National Renewable Energy Lab.* <http://www.ncbi.nlm.nih.gov/pubmed/21564034>
- Jonkman, B. J., & Kilcher, L. (2012). TurbSim User's Guide. In *National Renewable Energy Laboratory* (Issue September). <papers2://publication/uuid/0CDF5717-D4F8-4B8F-ABE8-A0C02844D1DE>
- Kane, M. B. (2014). *Wirelessly Enabled Control of Cyber-Physical Infrastructure with Applications to Hydronic Systems*. University of Michigan.
- Kashima, S., Yanaka, Y., Suzuki, S., & Mori, K. (2001). Monitoring the Akashi Kaikyo bridge: first experiences. *Structural Engineering International*, 11(2), 120–123.
- Kawakatsu, T., Aihara, K., Takasu, A., & Adachi, J. (2018). Deep sensing approach to single-sensor vehicle weighing system on bridges. *IEEE Sensors Journal*, 19(1), 243–256.
- Kerkez, B., Gruden, C., Lewis, M., Montestruque, L., Quigley, M., Wong, B., Bedig, A., Kertesz, R., Braun, T., Cadwalader, O., Poresky, A., & Pak, C. (2016). Smarter stormwater systems. *Environmental Science and Technology*, 50(14), 7267–7273. <https://doi.org/10.1021/acs.est.5b05870>
- Khaitan, S. K., McCalley, J. D., Liu, C. C., & Eds. (2015). *Cyber physical systems approach to smart electric power grid*. Springer.
- Kim, S., Lee, J., Park, M. S., & Jo, B. W. (2009). Vehicle signal analysis using artificial neural networks for a bridge weigh-in-motion system. *Sensors*, 9(10), 7943–7956.
- Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv: 1412.6980*.
- Krogh, B., Lee, E., Lee, I., Mok, A., Rajkumar, R., Sha, L. R., Vincentelli, A. S., Shin, K., Stankovic, J., & Sztipanovits, J. (2008). *Cyber-Physical Systems, Executive Summary*. [http://iccps.acm.org/2011/\\_doc/CPS-Executive-Summary.pdf](http://iccps.acm.org/2011/_doc/CPS-Executive-Summary.pdf)
- Kurata, M., Kim, J., Zhang, Y., Lynch, J. P., van der Linden, G. W., Jacob, V., Thometz, E., Hipley, P., & Sheng, L.-H. (2011). Long-term assessment of an autonomous wireless structural health monitoring system at the new Carquinez Suspension Bridge. *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2011*, 7983(April 2011), 798312. <https://doi.org/10.1117/12.880145>
- Kurata, N., Spencer, B. F., & Ruiz-Sandoval, M. (2005). Risk monitoring of buildings with wireless sensor networks. *Structural Control and Health Monitoring*, 12(3–4), 315–327.

- <https://doi.org/10.1002/stc.73>
- Lee, E. A., & Seshia, S. A. (2016). *Introduction to embedded systems: A cyber-physical systems approach*. MIT Press.
- Leutenegger, T., Schlums, D. H., & Dual, J. (1999). Structural Testing of Fatigued Structures. *Smart Structures and Integrated Systems, Proceedings of SPIE*, 3668, 987–997. <https://doi.org/10.1117/12.350775>
- Liu, C. (2012). *Foundations of MEMS*. Pearson Education India.
- Lydon, M., Taylor, S. E., Robinson, D., Mufti, A., & Brien, E. (2016). Recent developments in bridge weigh in motion (b-wim). *Journal of Civil Structural Health Monitoring*, 6(1), 69–81.
- Lynch, J. P., & Loh, K. J. (2006). A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring. *The Shock and Vibration Digest*, 38(2), 91–128. <https://doi.org/10.1177/0583102406061499>
- Lynch, Jerome Peter. (2007). An overview of wireless structural health monitoring for civil structures. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 345–372. <https://doi.org/10.1098/rsta.2006.1932>
- M., Mollineax, R., R. (2015). Structural health monitoring of progressive damage. *Earthquake Engineering and Structural Dynamics*, 44, 583–600. <https://doi.org/10.1002/eqe.2562>
- Maierhofer, C., Reinhardt, H.-W., & Dobmann, G. (2010). *Non-Destructive Evaluation of Reinforced Concrete Structures: Non-Destructive Testing Methods*. Elsevier.
- McKinsey Global Institute. (2020). *Reimagining infrastructure in the United States: How to build better* (Issue July). <https://www.mckinsey.com/business-functions/operations/our-insights/reimagining-infrastructure-in-the-united-states-how-to-build-better>
- Moller, D. P. F., & Vakilzadian, H. (2016). Cyber-physical systems in smart transportation. *IEEE International Conference on Electro Information Technology*, 776–781. <https://doi.org/10.1109/EIT.2016.7535338>
- Moses, F. (1979). Weigh-in-motion system using instrumented bridges. *Journal of Transportation Engineering*, 105(3).
- Ni, Y. Q., & Wong, K. Y. (2012). Integrating Bridge Structural Health Monitoring and Condition-Based Maintenance Management. *4th International Workshop on Civil Structural Health Monitoring*, 6–8.
- O'Brien, E. J., Rowley, C., Gonz'alez, A., & Al., E. (2009). A regularised solution to the bridge weigh in motion equations. *International Journal of Heavy Vehicle Systems*, 16(3), 310–327.
- Pakzad, S. N. (2010). Development and deployment of large scale wireless sensor network on a long-span bridge. *Smart Structures and Systems*, 6(5–6), 525–543. [https://doi.org/10.12989/sss.2010.6.5\\_6.525](https://doi.org/10.12989/sss.2010.6.5_6.525)
- Pakzad, S. N., Fenves, G. L., Kim, S., & Culler, D. E. (2008). Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring. *Journal of Infrastructure Systems*, 14(1), 89–101. [https://doi.org/10.1061/\(asce\)1076-0342\(2008\)14:1\(89\)](https://doi.org/10.1061/(asce)1076-0342(2008)14:1(89))
- Pan, C. D., Yu, L., Liu, H. L., Chen, Z. P., & Luo, W. F. (2018). Moving force identification based on redundant concatenated dictionary and weighted l1-norm regularization. *Mechanical Systems and Signal Processing*, 98, 32–49. <https://doi.org/10.1016/j.ymssp.2017.04.032>
- Pedregosa, Fabian; Varoquaux, Gael; Gramfort, Alexandre; Michel, Vincent; Bertrand, T. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <https://doi.org/10.1289/EHP4713>
- Peeters, B., & De Roeck, G. (2001). One-year monitoring of the Z24-bridge: Environmental effects versus damage events. *Earthquake Engineering and Structural Dynamics*, 30(2), 149–

171. [https://doi.org/10.1002/1096-9845\(200102\)30:2<149::AID-EQE1>3.0.CO;2-Z](https://doi.org/10.1002/1096-9845(200102)30:2<149::AID-EQE1>3.0.CO;2-Z)
- Petersen, K. E. (1982). Silicon as a mechanical material. *Proceedings of the IEEE*, 70(5), 420–457. <https://doi.org/10.1109/9780470545263.sect1>
- Poularikas, A. D., & Ramadan, Z. M. (1998). Discrete-time signal processing. In *Prentice Hall* (2nd ed.). <https://doi.org/10.1201/9781315221946-2>
- Prozzi, J. A., & Hong, F. (2007). Effect of weigh-in-motion system measurement errors on load-pavement impact estimation. *Journal of Transportation Engineering*, 133(1), 1–10. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2007\)133:1\(1\)](https://doi.org/10.1061/(ASCE)0733-947X(2007)133:1(1))
- Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. *Proceedings - Design Automation Conference*, 731–736. <https://doi.org/10.1145/1837274.1837461>
- Rawat, D. B., Bajracharya, C., & Yan, G. (2015). Towards intelligent transportation Cyber-Physical Systems: Real-time computing and communications perspectives. *SOUTHEASTCON*, 1–6. <https://doi.org/10.1109/SECON.2015.7132923>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Rousseeuw, P., & Driessen, K. (1999). A Fast Algorithm for the Minimum Covariance. *Technometrics*, 41(3), 212–223.
- Rytter, A. (1993). *Vibrational Based Inspection of Civil Engineering Structures*. Aalborg University.
- Schmidt, M., & Åhlund, C. (2018). Smart buildings as Cyber-Physical Systems: Data-driven predictive control strategies for energy efficiency. *Renewable and Sustainable Energy Reviews*, 90, 742–756. <https://doi.org/10.1016/j.rser.2018.04.013>
- Schneider, T., & Neumaier, A. (2001). Algorithm 808. *ACM Transactions on Mathematical Software*, 27(1), 58–65. <https://doi.org/10.1145/382043.382316>
- Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 1443–1471. <https://doi.org/10.1162/089976601750264965>
- Seo, J., Hu, J. W., & Lee, J. (2016). Summary Review of Structural Health Monitoring Applications for Highway Bridges. *Journal of Performance of Constructed Facilities*, 30(4). [https://doi.org/10.1061/\(asce\)cf.1943-5509.0000824](https://doi.org/10.1061/(asce)cf.1943-5509.0000824)
- Seshia, S. A., Hu, S., Li, W., & Zhu, Q. (2017). Design Automation of Cyber-Physical Systems: Challenges, Advances, and Opportunities. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9), 1421–1434. <https://doi.org/10.1109/TCAD.2016.2633961>
- Soga, K., & Schooling, J. (2016). Infrastructure sensing. *Interface Focus*, 6(4). <https://doi.org/10.1098/rsfs.2016.0023>
- Sohn, H., Czarnecki, A., & Farrar, C. R. (2000). Structural health monitoring using statistical process control. *Journal of Structural Engineering*, 126(11), 1356–1363.
- Sohn, H., & Farrar, C. R. (2001). Damage diagnosis using time series analysis of vibration signals. *Smart Materials and Structures*, 10(3), 446–451. <https://doi.org/10.1088/0964-1726/10/3/304>
- Sohn, Hoon, Farrar, C. R., Hemez, F., & Czarnecki, J. (2003). *A Review of Structural Health Monitoring Literature 1996 – 2001*.
- Sohn, Hoon, Farrar, C. R., Hunter, N. F., & Worden, K. (2001). Structural health monitoring using

- statistical pattern recognition techniques. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 123(4), 706–711. <https://doi.org/10.1115/1.1410933>
- Straser, E. G., Kiremidjian, A. S., Meng, T. H., & Redlefsen, L. (1998). Modular, wireless network platform for monitoring structures. *Proceedings of the International Modal Analysis Conference - IMAC, 1*, 450–456. [https://doi.org/10.1016/s0920-5489\(99\)91996-7](https://doi.org/10.1016/s0920-5489(99)91996-7)
- Structures, C. and. (2011). *CSI analysis reference manual for SAP2000, ETABS, SAFE, and CSiBridge*. <http://docs.csiamerica.com/manuals/etabs/Analysis Reference.pdf>
- Tang, Y. T., Chan, F. K. S., O'Donnell, E. C., Griffiths, J., Lau, L., Higgitt, D. L., & Thorne, C. R. (2018). Aligning ancient and modern approaches to sustainable urban water management in China: Ningbo as a “Blue-Green City” in the “Sponge City” campaign. *Journal of Flood Risk Management*, 11(4), 1–14. <https://doi.org/10.1111/jfr3.12451>
- Tibaduiza, D. A., Torres-Arredondo, M. A., Mujica, L. E., Rodellar, J., & Fritzen, C. P. (2013). A study of two unsupervised data driven statistical methodologies for detecting and classifying damages in structural health monitoring. *Mechanical Systems and Signal Processing*, 41(1–2), 467–484. <https://doi.org/10.1016/j.ymsp.2013.05.020>
- Tomer, A., Kane, J., & George, C. (2021). *Rebuild with purpose, AN AFFIRMATIVE VISION FOR 21ST CENTURY AMERICAN INFRASTRUCTURE* (Issue April).
- Tsiapoki, S., Häckell, M. W., Griebmann, T., & Rolfes, R. (2018). Damage and ice detection on wind turbine rotor blades using a three-tier modular structural health monitoring framework. *Structural Health Monitoring*, 17(5), 1289–1312. <https://doi.org/10.1177/1475921717732730>
- Van Overschee, P., & De Moor, B. (2005). *Two subspace algorithms for the identification of combined deterministic-stochastic systems*. 311(I), 511–516. <https://doi.org/10.1109/cdc.1992.371682>
- Verweij, S., Teisman, G. R., & Gerrits, L. M. (2017). Implementing Public–Private Partnerships: How Management Responses to Events Produce (Un) Satisfactory Outcomes. *Public Works Management and Policy*, 22(2), 119–139. <https://doi.org/10.1177/1087724X16672949>
- Vincent, G. S. (1958). Golden gate bridge vibration studies. *Journal of Structural Division*, 84(1817), 1–42.
- Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13(2), 260–269. <https://doi.org/10.1109/TIT.1967.1054010>
- Walford, C. a. (2006). Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs. In *Global Energy Concepts, LLC* (Issue March). <http://prod.sandia.gov/techlib/access-control.cgi/2006/061100.pdf>
- Wang, M. L., Lynch, J. P., & Sohn, H. (2014). Sensor Technologies for Civil Infrastructures, Volume 1: Sensing Hardware and Data Collection Methods for Performance Assessment. In *Elsevier*. <https://doi.org/10.1533/9780857099136.1>
- Wang, S., Liu, Q., Zhu, E., Porikli, F., & Yin, J. (2018). Hyperparameter selection of one-class support vector machine by self-adaptive data shifting. *Pattern Recognition*, 74, 198–211. <https://doi.org/10.1016/j.patcog.2017.09.012>
- Williams, E. J., & Messina, A. (1999). Applications of the multiple damage location assurance criterion. *Proceedings of the International Conference on Damage Assessment of Structures (DAMAS 99)*, Dublin, Ireland, 256–264. <https://doi.org/10.4028/www.scientific.net/kem.167-168.256>
- Wong, K.-Y., & Ni, Y.-Q. (2009). Modular Architecture of SHM Systems for Cable-Supported

- Bridges. In *Encyclopedia of Structural Health Monitoring* (pp. 2089–2106).
- Wong, K. Y. (2004). Instrumentation and health monitoring of cable-supported bridges. *Structural Control and Health Monitoring*, 11(2), 91–124. <https://doi.org/10.1002/stc.33>
- Worden, K., Farrar, C. R., Manson, G., & Park, G. (2007). The fundamental axioms of structural health monitoring. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2082), 1639–1664. <https://doi.org/10.1098/rspa.2007.1834>
- Ying, Y., Garrett, J. H., Oppenheim, I. J., Soibelman, L., Harley, J. B., Shi, J., & Jin, Y. (2013). Toward Data-Driven Structural Health Monitoring: Application of Machine Learning and Signal Processing to Damage Detection. *Journal of Computing in Civil Engineering*, 27(6), 667–680. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000258](https://doi.org/10.1061/(asce)cp.1943-5487.0000258)
- Yu, Y., Cai, C., & Deng, L. (2016). State-of-the-art review on bridge weigh-in-motion technology. *Advances in Structural Engineering*, 19(9), 1514–1530.
- Yu, Y., Cai, C., & Deng, L. (2017). Vehicle axle identification using wavelet analysis of bridge global responses. *Journal of Vibration and Control*, 23(17), 2830–2840.
- Zaidi, S. S. H., Aviyente, S., Salman, M., Shin, K. K., & Strangas, E. G. (2011). Prognosis of gear failures in dc starter motors using hidden Markov models. *IEEE Transactions on Industrial Electronics*, 58(5), 1695–1706. <https://doi.org/10.1109/TIE.2010.2052540>
- Zeng, T., Semiari, O., Saad, W., & Bennis, M. (2019). Joint Communication and Control for Wireless Autonomous Vehicular Platoon Systems. *IEEE Transactions on Communications*, 67(11), 7907–7922. <https://doi.org/10.1109/TCOMM.2019.2931583>
- Zhang, Y., Zhao, H., & Lie, S. T. (2018). A nonlinear multi-spring tire model for dynamic analysis of vehicle-bridge interaction system considering separation and road roughness. *Journal of Sound and Vibration*, 436, 112–137. <https://doi.org/10.1016/j.jsv.2018.08.039>
- Zhao, Z., & Uddin, N. (2014). Determination of Dynamic Amplification Factors Using Site-Specific B-WIM Data. *Journal of Bridge Engineering*, 19(1), 72–82. [https://doi.org/10.1061/\(asce\)be.1943-5592.0000491](https://doi.org/10.1061/(asce)be.1943-5592.0000491)
- Zhou, W., Kovvali, N., Papandreou-Suppappola, A., Cochran, D., & Chattopadhyay, A. (2007). Hidden Markov model based classification of structural damage. *Proc. SPIE 6523, Modeling, Signal Processing, and Control for Smart Structures 2007*, 6523(April 2007), 652311. <https://doi.org/10.1117/12.716132>
- Zolghadri, N., Halling, M. W., Johnson, N., & Barr, P. J. (2016). Field verification of simplified bridge weigh-in-motion techniques. *Journal of Bridge Engineering*, 21(10).