CIAMTIS
U.S. DOT Region 3 University Transportation Center

Fatigue Life Estimation of Bridges with Smart Mobile Sensing

June 1, 2021

Prepared by:
S. Pakzad and M. Takac
Lehigh University

r3utc.psu.edu

PennState College of Engineering | LARSON TRANSPORTATION INSTITUTE

Technical Report Documentation Page

1. Report No. CIAM-UTC-REG7	**2. Government Accession No.**	**3. Recipient's Catalog No.**

4. Title and Subtitle Fatigue Life Estimation of Bridges with Smart Mobile Sensing	**5. Report Date** June 1, 2021
	6. Performing Organization Code
7. Author(s) Shamim N. Pakzad, Martin Takac	**8. Performing Organization Report No.**
9. Performing Organization Name and Address Lehigh University 27 Memorial Drive West Bethlehem, PA 18015 USA	**10. Work Unit No. (TRAIS)**
	11. Contract or Grant No. 69A3551847103
12. Sponsoring Agency Name and Address U.S. Department of Transportation Research and Innovative Technology Administration 3rd Fl, East Bldg E33-461 1200 New Jersey Ave, SE Washington, DC 20590	**13. Type of Report and Period Covered** Final Report 03/01/2019 – 12/31/2021
	14. Sponsoring Agency Code

15. Supplementary Notes

Work funded through The Pennsylvania State University through the University Transportation Center Grant Agreement, Grant No. 69A3551847103.

16. Abstract

Bridge structures experience significant vibrations and repeated stress variations during their lifecycle. These conditions are the bases for fatigue analysis to identify fatigue cracking, which can be used to accurately establish the remaining fatigue life of the structures (i.e., the number of stress cycles before fatigue failure). This is typically achieved through a full-field strain assessment of the fatigue-critical locations of the structure over a typical loading period. Traditional inspection methods collect strain measurements by using strain gauges for fatigue life assessment. Large-scale deployment of wired strain gauges, however, poses a fundamental limitation: they are expensive and laboriously impractical as more spatial information is desired. Addressing these limitations begs for an innovative sensing strategy where information can be integrated from inexpensive data sources. Acceleration data, on the other hand, can be collected relatively inexpensively by means of mobile sensing, which is increasingly an area of interest in many fields of engineering. Mobile sensing eliminates the spatially restrictive nature of fixed sensor networks; the spatial frequency of a mobile sensor network is a direct function of the speed and its sensors' sampling frequencies, as well as the number of mobile devices that can simultaneously collect measurements from the same structure. In this project we have developed a deep learning-based methodology that facilitates the use of inexpensively generated structural response data such as accelerations from mobile sensing, to first estimate strain and then subsequently find the remaining fatigue life of a structure and other higher level information that aids in prognosis and decision-making.

17. Key Words Mobile sensing, fatigue life estimation, remaining useful life, bridge structure, lifecycle	**18. Distribution Statement** No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 24	**22. Price**

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Bridge structures are subjected to significant vibrations and repeated stress variations during their operational life. This leads to a potential for fatigue cracking and subsequent fatigue failure in bridges. Hence, fatigue analysis is necessary to accurately estimate the remaining fatigue life of a structure. This is typically estimated in terms of the number of stress cycles to which a structure can be subjected prior to failure. In this regard, the rain-flow counting algorithm is widely used in the analysis of fatigue data in order to reduce a spectrum of varying stress into a set of simple stress reversals and assess the fatigue life of a structure subjected to complex loading (Downing and Socie 1982). This procedure requires a full-field strain assessment of fatigue-critical locations of a structure of interest spanning over a typical loading period. Traditionally this entails collecting strain measurements by deploying strain gauges for fatigue life assessment. However, large-scale deployment of wired strain gauges is limited by cost and impractical deployment effort necessary with increase in desired spatial resolution of the information being sought (Cerda et al. 2010). An innovative sensing strategy is necessary to address these limitations wherein information can be extracted from inexpensive data sources.

Unlike strain, acceleration can be measured relatively inexpensively by means of mobile sensing, an area of significant interest across many fields of engineering. A mobile sensing paradigm eliminates the spatially restrictive feature of a fixed sensing approach. The spatial frequency obtained from mobile sensing is a function of speed, individual sensor sampling frequency, and the number of mobile devices that can simultaneously collect measurements from the same structure (Yang et al. 2004). This approach paves the way for crowdsourcing the data from individual members of the public, ensuring they take a much more active role in data collection. This will aid in providing up-to-date information on a structure's health efficiently (Matarazzo and Pakzad 2013).

In this project we propose a deep learning framework to convert acceleration data to strain data utilizing underlying physical principles, and use the resulting strain data for fatigue life estimation. Figure 1 summarizes the goals of the project.

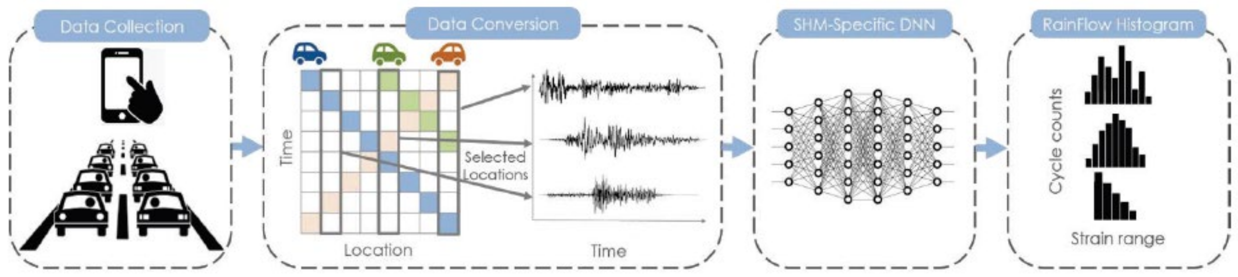


Figure 1. A general framework of the proposed methodology.

BACKGROUND

High-fidelity modeling of complex infrastructure systems is a challenging task owing to limitations of modeling assumptions and significant computational costs necessary for analysis. Instead, data-driven approaches are attractive alternatives in the context of structural health monitoring (SHM) applications. This entails the construction of a surrogate model using acquired data from a system of interest that substitutes the physical model, circumventing the need for high-fidelity modeling (Sen and Nagarajaiah 2018). With recent advancements in sensing technology enabling crowdsourcing, the abundance of sensed data provides an exciting opportunity for inferring the condition of the built environment at an unprecedented rate and resolution through the design life. The availability of "big data," however, leads to challenges associated with storage and analysis that may impact fatigue life estimation due to the inclusion of multiple years' worth of sensed data.

Training deep neural networks (DNNs), also referred to as deep learning, has emerged as the state-of-the-art tool for harnessing big data (Najafbadi et al. 2015). With the increase in size of available data for training DNNs, deep learning algorithms have demonstrated enhanced performance in all required tasks in fields spanning from engineering to natural sciences. This makes DNN an ideal candidate for the problem at hand. DNNs consist of multiple layers of neural network elements to process and extract inherent features of a data set. Each layer holds valuable information related to the features and the underlying structure of the data that helps build a surrogate model that facilitates data-driven decision-making and predictions.

The implementation of DNNs varies greatly depending on the domain of application. For example, convolutional neural networks (CNNs) are preferred to deal with spatial dependencies (e.g., image classification, video recognition) (Lecun and Bengio 1995, Krizhevsky et al. 2012). CNNs are composed of local receptive fields and shared weights that allow the extraction of multiple feature maps to encode spatial correlations in the input data set. On the other hand, recurrent neural networks (RNNs) are extensively used to extract temporal dynamic behavior for a time sequence (e.g., language translation, speech recognition, language modeling) (Gers et al. 1999, Graves et al. 2013, Sutskever et al. 2014). RNNs have two sources of input, the present and the recent past of a temporal sequence, which combine to determine how they respond to new data. In other words, the decision that an RNN makes at time step t-1 affects the decision it will reach at the subsequent time step t.

In civil engineering applications, dense sensor networks produce dynamic and nonlinear spatio-temporal data, which are unique to each structure. The acquired data can then be used for dynamic characterization of the structure, and subsequently for tasks such as parameter estimation and condition assessment that have a physical basis, unlike applications in many other fields. This necessitates the development of customized neural networks that, in addition to harnessing the power of big data, should account for the inherent physical principles that govern the behavior of the system. This physics-inspired, data-driven paradigm will lead to enhanced performance.

OBJECTIVES

The overarching theme of the project was to employ recent advances in deep learning in conjunction with rigorous physics-based foundations to exploit sensed data from SHM applications. The key objectives of the project were as follows:

1. To develop a deep learning-based framework that accounts for known structural behavior and estimate strains from acceleration data. The proposed DNN architecture will also address SHM-specific issues such as measurement noise, missing data, and robustness of the resulting decision-making framework. Since the proposed DNN architecture involves only linear operations followed by simple nonlinear operations, post training it enables real-time analysis.
2. To develop a mobile phone application that collects acceleration and GPS data (e.g., when the phone is traveling over a bridge) from multiple phones and uploads it to a server. On this server, the data

uploaded from multiple phones will be aligned, categorized, and processed. The acquired data will then be used to train the DNN framework.

3. To validate the performance of the proposed frameworks through experimental studies. The laboratory experiment includes a horizontally curved girder.

DATA AND DATA STRUCTURES

The project evaluates the performance of the proposed DNN framework through laboratory experiments. For objective 1, a laboratory experiment involving a curved girder was performed. The details of the testbed and instrumentation are provided in the subsequent sections of this report. The installed sensors collected both accelerations and strains from the girder. For objective 2, a laboratory experiment involving two mobile phones placed on a shaking table was performed. This was a first step to test the efficacy of the data collection prowess and time synchronization that are crucial for an effective analysis.

The raw data are available upon request from the PI as ASCII text files.

CHAPTER 2

Methodology

PROPOSED DNN FRAMEWORK

DNN for converting acceleration to strain

Overview

In this section we provide a general map of the proposed technique composed of training and testing phases. The training phase aims to establish a relationship between acceleration and strain for selected locations on a structure (i.e., the number of locations, R) under the loading conditions included in the training dataset. The proposed DNN is trained to minimize the error between predicted and actual strain responses at the selected locations to establish the required relationship. The trained model parameters are subsequently used in the testing phase, wherein acceleration data from desired locations are used as inputs to the proposed architecture to predict strain responses. It should be noted that the number of desired locations is greater than R.

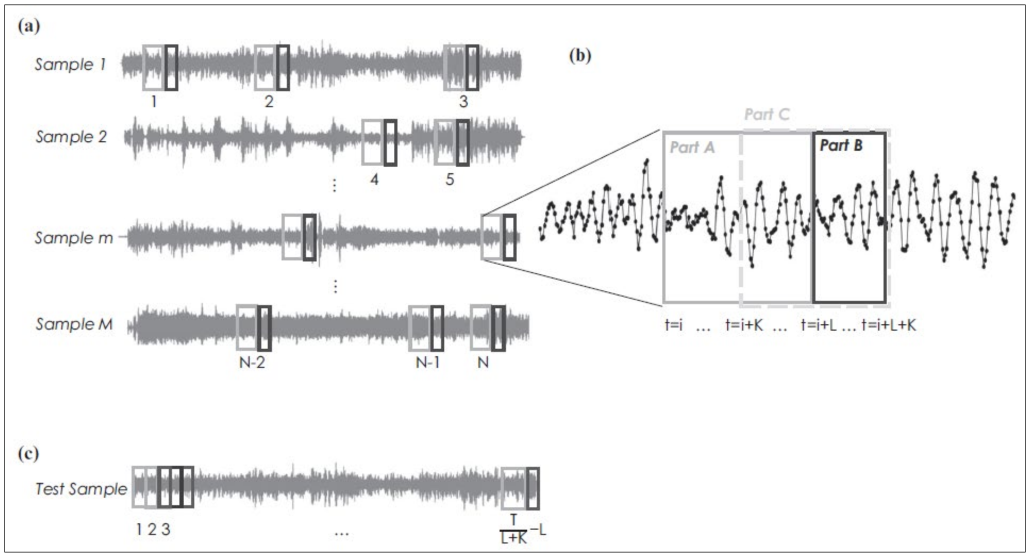
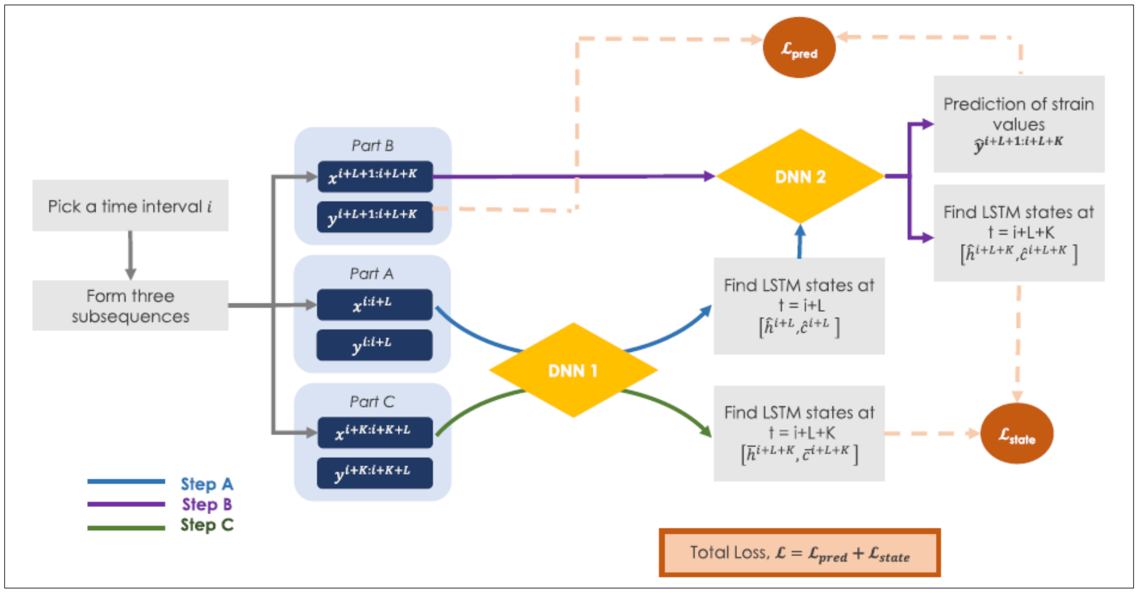


Figure 2. (a) Randomized mini-batch for training phase, (b) closer look at the subsequence, (c) sampling for testing phase.

Let us assume that the training data set has a total number of M samples each of length T, that includes sensor readings from only one location ($R = 1$). Each sample ($m: 1, \cdots, M$) has both acceleration $\{x_m^1, x_m^2, \dots, x_m^i, \dots, x_m^T\}$ and strain responses $\{y_m^1, y_m^2, \dots, y_m^i, \dots, y_m^T\}$. The proposed approach randomly selects sequences from the given training dataset and utilizes only a portion of these sequences. Figure 2a shows M acceleration responses from a sensor and randomized mini-batch selection with size N. The sub-sequences form mini-batches such that there might be overlap across sequences, but not across the samples. The random sampling is repeated for each iteration during the training.



$\mathcal{L}_{pred}$
Prediction of strain values $\hat{y}^{i+L+1:i+L+K}$
Part B
$x^{i+L+1:i+L+K}$
$y^{i+L+1:i+L+K}$
DNN 2
Pick a time interval i
Find LSTM states at t = i+L+K $[\hat{h}^{i+L+K}, \hat{c}^{i+L+K}]$
Form three subsequences
Part A
$x^{i:i+L}$
$y^{i:i+L}$
Find LSTM states at t = i+L $[\hat{h}^{i+L}, \hat{c}^{i+L}]$
DNN 1
Part C
$x^{i+K:i+K+L}$
$y^{i+K:i+K+L}$
Find LSTM states at t = i+L+K $[\bar{h}^{i+L+K}, \bar{c}^{i+L+K}]$
$\mathcal{L}_{state}$
Step A
Step B
Step C
Total Loss, $\mathcal{L} = \mathcal{L}_{pred} + \mathcal{L}_{state}$

Figure 3. A general map of the proposed approach.

The proposed methodology performs a three-step training to learn long time sequences. In each step, the randomly selected subsequence forms three parts: Part A, Part B, and Part C (Figure 1b) to be used in Step A, Step B, and Step C, respectively. The procedure of the approach for each iteration of an epoch is presented in Figure 3 (i.e., an epoch is one forward and one backward pass of all training datasets; an iteration is one forward and one backward pass of the batch). In Step A, Part A from time i up to time $i + L$, $\{x_m^{i:i+L}\}$, feeds the first network (DNN 1). This L length sequence is adopted for the high-level estimation of the input and the initial state for Step B. Such a step can also be thought of as the nonlinear mapping of the input to summarize the dynamics of the sequence for more efficient learning and faster convergence. Step B feeds the second network (DNN 2) that is composed of long short-term memory (LSTM) memory cells. These cells operate on the K length input sequence $x_m^{i+L+1:i+L+K}$ and the initial state estimation from Step A. The LSTM cell translates the acceleration sequences to the strain sequences from time $i + L + 1$ to $i + L + K$ and updates both the internal and hidden states. Step C utilizes the time series $x_m^{i+K:i+L+K}$ to help DNN 1 to make better initial state estimations for the next iterations. In other words, Steps A and C share the same parameters. The goal of the proposed approach is to minimize the loss between the cell states from Steps B and C ($\mathcal{L}_{state}$), as well as the true and predicted strain time series ($\mathcal{L}_{pred}$) at the same time.

During testing, all of the sequence is predicted by sliding the indexes used while forming the parts. Figure 2c shows the first subsequence $x_m^{1:L+K}$ with a length $L + K$ that estimates the strain response of length K. Then, the subsequence $x_m^{K:L+2K}$ starts estimating the next part of the sequence of length K. By repeating this, the whole time series is predicted. During testing, the first K length response estimation needs high-level representation for the first L points. Thus, the total number of subsequences necessary to estimate the strain response is the total length of samples T divided by $L+K$ and subtracted by the first L points.

Architechture and training

During training, inputs are designed as tensors that contain the features of the input sequence with the shape of [minibatch size, subsequence length, number of sensors]. For Step A, the N batch of input series with shape $[N, L, R]$ is fed into the DNN 1 to obtain a high-level representation of the input. DNN 1 is designed to have multiple fully connected (FC) layers to create initial estimation for the internal (c) and hidden (h) states at time $i+L$. The ReLU function, $f(x)=\max(0, x)$, is adopted as nonlinear activation. The shapes of both states are defined as [mini-batch size, number of LSTM layers, hidden layer size].

The input of Step B with the shape $[N, K, R]$ is fed through DNN 2, which is composed of LSTM cells. In this step, the output of the LSTM is fed through an FC layer to obtain a strain sequence with the same size of the input. Furthermore, the calculated internal and hidden states are updated from time $i+L+1$ to time $i+L+K$. In the last step, the network takes $x_m^{i+K:i+L+K}$ to retain the nonlinear representation of the context and updates the internal and cell states found by DNN 1.

The loss function is defined as the summation of mean squared error of the predicted and true values of the acceleration sequences, and internal and hidden states from Step B and Step C. The network is trained by ADAM optimizer, which is an adaptive learning rate algorithm (Kingma & Ba, 2014). The computations are performed on NVIDIA Tesla K80 GPUs. PyTorch is used as the optimized tensor library for deep learning (Paszke et al. 2019).

EXPERIMENTAL SETUP FOR VALIDATING DNN FRAMEWORK

To address geometric design of transportation infrastructure such as complex geometries, site constraints, and dense highway infrastructure systems, horizontally curved highway bridges are widely constructed despite design challenges stemming from torsional loads and flange lateral buckling (Putnam 2010). These bridges commonly consist of either I-girders or box girders that suffer from torsional weakness and/or costly fabrication. A study by Dong (2008) offered tubular flange girder (TFG) that replaces the flanges of the I-girder with tubular flanges to have higher torsional stiffness, higher strength, and easy manufacture.

Dong (2008) designed a full-scale curved TFG according to AASHTO (2005). This was followed by a half-scale test specimen built by Putnam (2010) to analyze the behavior of TFG subjected to the loads in the elastic regime. In this project we use the same test specimen to evaluate the performance of the proposed methodology on complex geometry. The following sections present the details of the test setup, procedure, and vibration analysis of the specimen.

Test setup

The test setup includes the main components of support beams, end overturning brackets, and bearings, as shown in Figure 4. A simple span horizontally curved girder is placed on top of two W14×167 steel beam pedestals, which are elevated such that the north pedestal is on a same level with the south pedestal. One-inch stiffeners are welded to the support pedestals with the approximate length of 1.83 m (6 ft). The curved girder is restrained by two 10.16-cm (4-inch) diameter pin bearings located on top of the support beams. The end overturning brackets at the bearings are positioned to resist the torques resulting from vertical loads acting on the shear center of the girder and prevent it from rolling outside. These brackets are attached to the support beams with bolts.

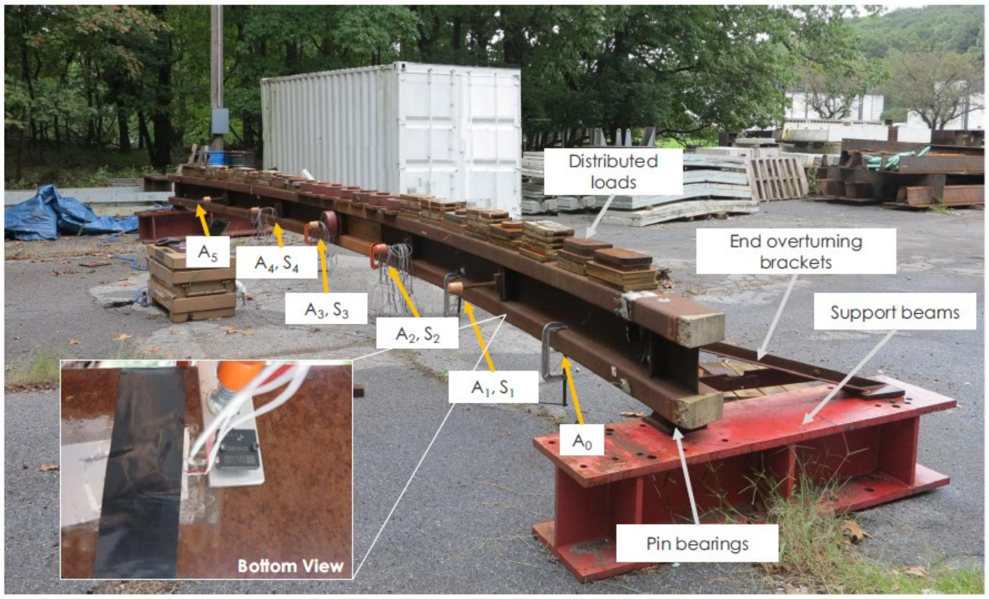


Figure 4. The laboratory test setup along with the instrumentation plan.

The TFG test specimen is approximately 9.14 m (30 ft) long and 40.64 cm (16 ft) deep and consists of tubular flanges welded to the web. The top and bottom tubes are ASTMA500 Grade B with a minimum yield stress of $F_y = 317$ MPa and ultimate tensile stress of $F_u = 400$ MPa. The girder has five transverse stiffeners and two bearing stiffeners that increase the shear resistance of the web and control web and tube distortion. The three faces of contact between stiffener and girder are welded. A concrete infill is injected in the region of the bearings to stiffen it for the reaction forces. The total self-weight of the test specimen consisting of the top and bottom tubes, web, and transverse stiffeners is approximately 670 kg.

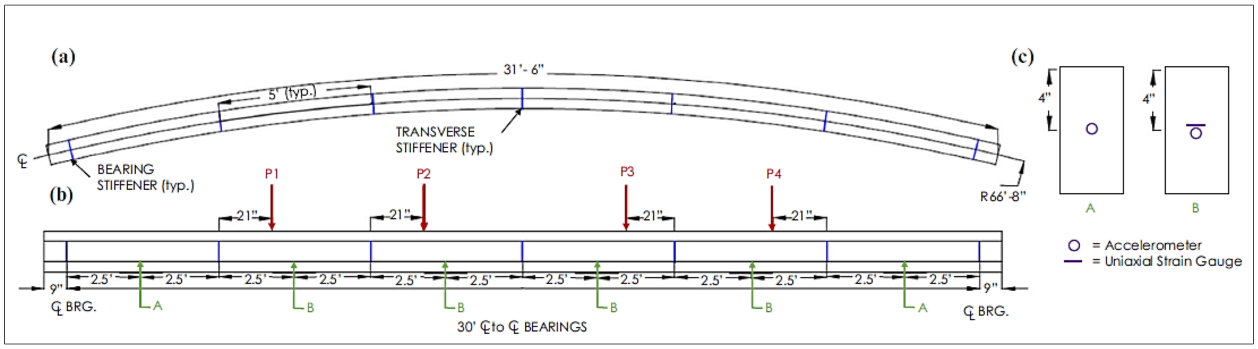


Figure 5. (a) Plan view, (b) location of the loadings and sensors, (c) strain gauge location details for the bottom of bottom tube.

Instrumentation

The girder is measured by a network of six accelerometers ($A_0 - A_5$), four strain gauges ($S_1 - S_4$), and a portable data acquisition system. Figure 4 shows the labeling and an overview of the instrumentation locations. Both the accelerometers and strain gauges are laid out across the length of the girder between the transverse stiffeners, along the midway of the bottom flange, as in Figure 5c.

The vertical bridge acceleration is usually significant for bridges because of its relevance to the stress of the members and their fatigue lives (Chen and Cai 2007). The response of the test specimen is measured by using six uniaxial DC accelerometers that are manufactured by Silicon Designs, Inc. (model 2210-002). The sensors have a characteristic noise floor of 10 μg/√Hz for ±2 g. The accelerometers are attached to the metal plate that is clamped to the girder.

The instrumentation consists of four weldable strain gauges LWK-06-W250B-350 with an active grid length of 0.25 inch. The strain gauges are spot welded to the tested structure in the field such that it measures strain in the axial direction. After installation, the gauges are covered with a multilayer system and then sealed with a silicon type agent. A Campbell Scientific CR9000 Data Logger is used for the collection of the data throughout the long-term monitoring. The logger is a high speed, a multichannel 16-bit system that is configured with digital and analog filters. In this setup, the data are filtered using a filter card at a frequency of 100 Hz.

Test procedure

The vibration and strain measurement of the curved girder are collected to establish the dynamic characteristics of the structure. Ambient vibrations are recorded, each time for over a 10-min duration using a sampling frequency of 250 Hz (1,500,000 samples/channel). The data are collected for 2 months on different days and at different time periods with varying temperatures of 9 ˚C – 29 ˚C. Distributed weights are added to the system such that the three lowest mode frequencies are between 5 and 50 Hz. The steel plate weights are placed such that they cover the whole surface of the girder except the loading locations. The total distributed load added to the system is measured as 120.84 kg/m.

Bridges are subjected to different types of vibration during their operation conditions such as low-amplitude ambient vibrations due to wind loading and car traffic, sudden high-amplitude impact-type ambient vibrations due to heavy trucks, and stronger vibrations. In order to perform similar excitation scenarios, four different loading types are applied on the girder to the locations illustrated in Figure 4. Table 1 summarizes the number of samples, the application locations, and maximum and minimum ambient vibration amplitudes of the load categories.

Table 1. Load types.

Type	Location	No. of Samples	Max. Accel. (mg)	Min. Accel. (mg)	Max. Strain (με)	Min. Strain (με)
Type I	P1 or P2 or P4	54	196.0	-242.6	37.31	-24.39
Type II	P1 and P4	20	306.6	-413.0	52.02	-31.53
Type III	Hammer and P1 or P4	16	478.0	-685.6	28.22	-13.09
Type IV	P1, P2, P3, and P4	12	516.9	-723.3	56.54	-52.74

For the low-amplitude ambient vibrations, a person taps on the girder with random frequencies without adding extra weight on the girder. The loading is applied on the structure at locations P1, P2, or P4 and called Type I loading. For Type I, the magnitude of the vibrations does not exceed 200 mg and strains are less than 40 με. A similar type of loading condition is generated by two people tapping the girder at points P1 and P4. Such loading is applied to get medium-amplitude vibrations and is called Type II loading. Heavier loading is simulated by hitting the structure with a hammer and stepping on different locations at the same time. One person hits the girder with different amplitudes and frequencies while another person taps the girder at the points P1, P2, or P4. This type of loading is referred to as Type III loading. Type III loading produces high acceleration but smaller strains relative to the other loadings. Finally, strong excitation is simulated by tapping the girder from four locations at the same time to obtain Type IV loading. For Type IV, high vibrations cause high strains in the girder.

Data analysis

Modal parameter identification is performed using the output-only Eigensystem Realization Algorithm (ERA) using software package SMIT (Chang and Pakzad 2013). In this study, only vertical mode shapes are an area of interest due to the sensor placement (as opposed to transverse and torsional modes). Table 2 summarizes the average modal frequency and damping ratios for each loading type. Vertical modal frequencies have an agreement for each loading category (i.e., the highest error is less than 5% except the fourth mode, which is not observable in all samples). However, the errors in the estimated damping ratios are relatively high, which is a common problem in the literature (Eshkevari and Pakzad 2020).

Table 2. Modal frequencies of the load types.

Type	Mode 1 Freq. (Hz)	Mode 2 Freq. (Hz)	Mode 3 Freq. (Hz)	Mode 4 Freq. (Hz)	Mode 1 Damping (%)	Mode 2 Damping (%)	Mode 3 Damping (%)	Mode 4 Damping (%)
Type I	6.81	26.31	47.90	67.97	2.16	1.23	2.11	1.67
Type II	6.84	26.03	47.52	61.00	2.12	1.24	2.02	3.42
Type III	6.70	26.30	48.42	61.44	2.32	1.30	2.11	3.77
Type IV	6.54	26.08	47.28	61.18	2.49	1.42	3.03	3.79

Proposed architechture

Figure 6 shows the proposed DNN architecture that is constructed after a design space exploration process. The network operates on time series that is normalized between $[-1,1]$ after dividing by the maximum of the training dataset. Then the sequences are downsampled by four to remove high-frequency noise existing in the data. Eighty-five percent of the post-processed time histories are randomly selected to form the training dataset. The remaining 15% of the samples are used to perform testing. During training, acceleration time histories collected from one location ($R = 1$) are used to estimate the strain responses. Powers of two are often selected to be the mini-batch size; thus, N is adopted as 128. The lengths of the subsequences formed in Step A (L) and Step B (K) are assigned as 50 and 100, respectively. In every iteration, one forward and backward pass is completed and parameters are updated for a mini-batch. Then the training and testing errors of the samples are computed. This process is repeated for 1,000 iterations per epoch. The network is run for 100 epochs each consisting of 1,000 iterations.

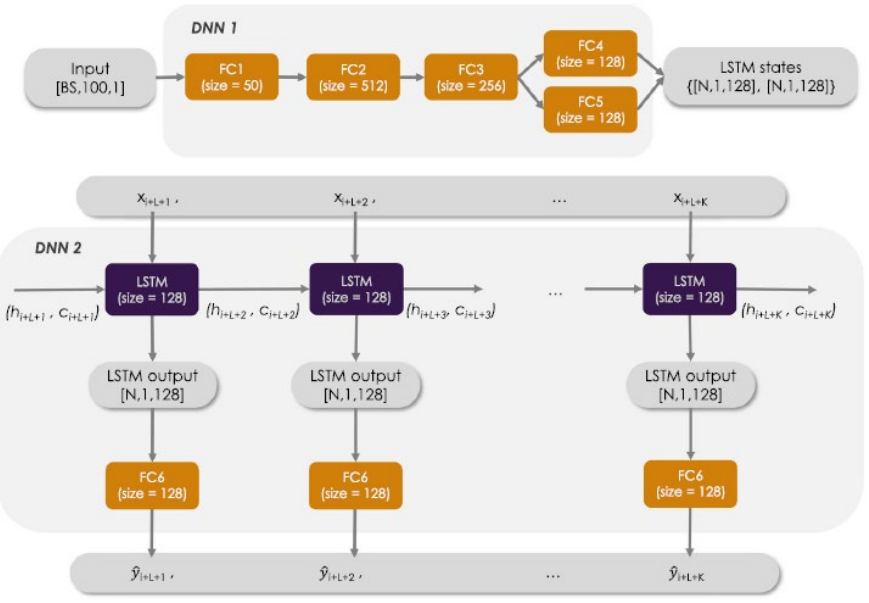


Figure 6. The proposed network topology.

For Step A, input size of [128, 50, 1] is stacked into [128, 200] shaped array to feed five connected layers. The resultant of these FC layers are used to create hidden and internal state estimates for Step B. The size of the LSTM hidden state is defined as 128. During this step, both strain sequences and LSTM state parameters are predicted. The output of LSTM cell is passed through another FC layer to obtain strain estimates in the selected location. In Step C, [128, 150, 1] sized inputs are fed into the same four FC layers to regularize the LSTM state parameter of Step B.

SOFTWARE DEVELOPMENT FOR ANDROID APPLICATION

The software solution has two parts, the android application and the database server. Let us describe each part separately.

The database server is used to store the data collected from the mobile applications. We have implemented a PHP web service, which after successful authentication will store the data pushed by the android application. The data are stored in a MySQL database. For each unit of data stored, the database will not only store the measured data but also a unique identifier of the phone that executed the android application. The database server also keeps a list of bridges that a user would like to collect data from.

The android application periodically checks for the list of bridges and allows users to confirm or decline the collection process for each offered bridge. For every bridge from which the user allows the application to collect data, we create a Geofence, https://developer.android.com/training/location/geofencing. The goal of this application programming interface (API) is to allow the application to be inactive when the user is far away from a bridge, and once the user gets into close proximity of the bridge, the application will wake up automatically and will start collecting data. Once the user leaves the bridge (geofence), the collection process is stopped. Every 15 minutes the application checks if there are any data collected ready to be uploaded to the database server (if the phone is not on wifi, nothing will be uploaded even if there are some measurements). Before uploading the data to the server, we get the current time from time server, using the Network Time Protocol (NTP) https://developer.android.com/reference/com/google/android/things/device/TimeManager. Subsequently, we compare the time with a local device time. The difference between local device time and the time server (time offset) is also stored with the data in the database.

Test procedure

Before testing the android application on real bridges, we first wanted to test the data collection process and the time synchronization with the time server. To accomplish this, we installed the collection application on two android phones and placed them next to each other on a shaker (see Figure 7).

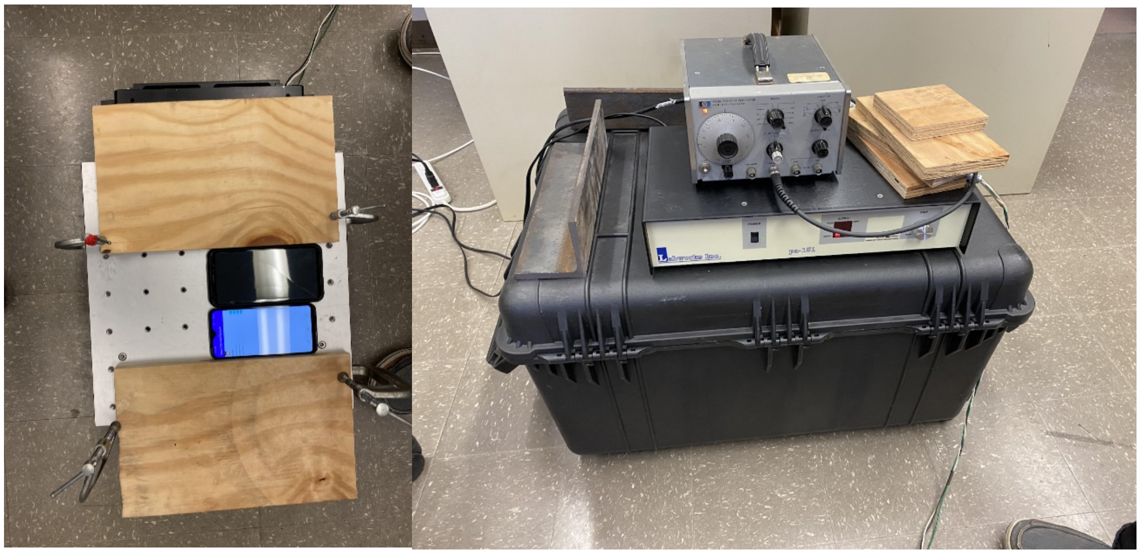


Figure 7. Experimental setup for testing the data collection API: (left) two smartphones attached to a shaking table, (right) signal generator for shaking table.

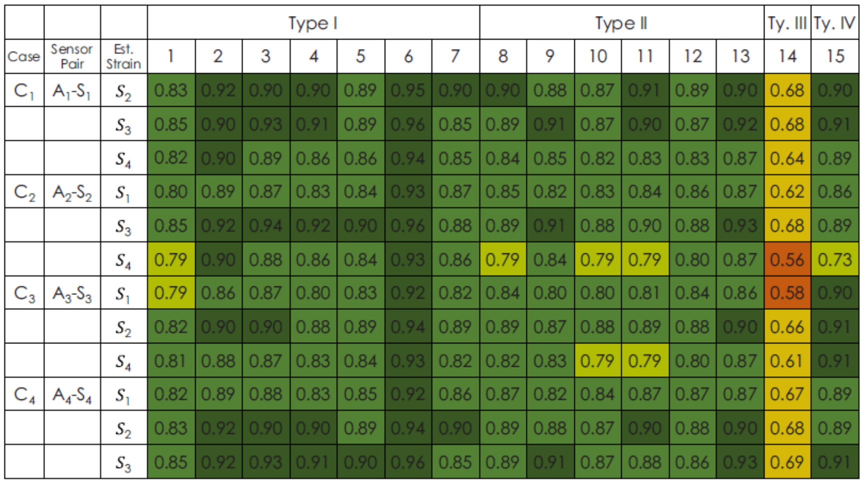
CHAPTER 3

Findings

RESULTS FOR DNN FRAMEWORK

This section presents the performance analysis of the proposed deep learning methodology. The applicability of the algorithm is verified through four cases, where each training case employs measured acceleration and strain from one location. Then, the testing phase is performed by using only measured acceleration collected from various desired locations. For instance, in Case 1, the acceleration–strain sensor pair of A_1 and S_1 is adopted for training. The saved model parameters and acceleration data A_2, A_3, A_4 are utilized to estimate strains where S_2, S_3, S_4 are located. The estimated strains are called $\hat{S}_2$, $\hat{S}_3$, $\hat{S}_4$. Note that only accelerometers ($A_1 - A_4$) are used during the assessment of the methodology. A_0 and A_5 are utilized only for the analysis of data (modal identification described earlier).

The error in the strain estimates is measured by using time–response assurance criterion (TRAC). TRAC number is a widely used measure in virtual sensing that gives the overall correlation of the time histories of true and estimated strain values. TRAC values close to unity indicate high correlation of the estimated strains.



			Type I							Type II						Ty. III	Ty. IV
Case	Sensor Pair	Est. Strain	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C_1	A_1-S_1	S_2	0.83	0.92	0.90	0.90	0.89	0.95	0.90	0.90	0.88	0.87	0.91	0.89	0.90	0.68	0.90
		S_3	0.85	0.90	0.93	0.91	0.89	0.96	0.85	0.89	0.91	0.87	0.90	0.87	0.92	0.68	0.91
		S_4	0.82	0.90	0.89	0.86	0.86	0.94	0.85	0.84	0.85	0.82	0.83	0.83	0.87	0.64	0.89
C_2	A_2-S_2	S_1	0.80	0.89	0.87	0.83	0.84	0.93	0.87	0.85	0.82	0.83	0.84	0.86	0.87	0.62	0.86
		S_3	0.85	0.92	0.94	0.92	0.90	0.96	0.88	0.89	0.91	0.88	0.90	0.88	0.93	0.68	0.89
		S_4	0.79	0.90	0.88	0.86	0.84	0.93	0.86	0.79	0.84	0.79	0.79	0.80	0.87	0.56	0.73
C_3	A_3-S_3	S_1	0.79	0.86	0.87	0.80	0.83	0.92	0.82	0.84	0.80	0.80	0.81	0.84	0.86	0.58	0.90
		S_2	0.82	0.90	0.90	0.88	0.89	0.94	0.89	0.89	0.87	0.88	0.89	0.88	0.90	0.66	0.91
		S_4	0.81	0.88	0.87	0.83	0.84	0.93	0.82	0.82	0.83	0.79	0.79	0.80	0.87	0.61	0.91
C_4	A_4-S_4	S_1	0.82	0.89	0.88	0.83	0.85	0.92	0.86	0.87	0.82	0.84	0.87	0.87	0.87	0.67	0.89
		S_2	0.83	0.92	0.90	0.90	0.89	0.94	0.90	0.89	0.88	0.87	0.90	0.88	0.90	0.68	0.89
		S_3	0.85	0.92	0.93	0.91	0.90	0.96	0.85	0.89	0.91	0.87	0.88	0.86	0.93	0.69	0.91

TRAC:	1.0-0.9	0.9-0.8	0.8-0.7	0.7-0.6	0.6-0.5	<0.5

$$TRAC = \frac{\left(\sum_{t=0}^{t=T} \hat{y}(t)y(t)\right)^2}{\left(\sum_{t=0}^{t=T} \hat{y}(t)\hat{y}(t)\right)\left(\sum_{t=0}^{t=T} y(t)y(t)\right)}$$

Figure 8. TRAC values of the testing dataset.

The TRAC analysis of 15 testing samples was performed for four evaluation scenarios and the results are summarized in Figure 8. Each case includes three strain predictions based on four different load types. The majority of the configurations have TRAC values above 0.8; hence, there is a high agreement between the predicted and true time histories.

Type III loading is a special case where high acceleration does not cause high strain values. Hence, in this caese the relationship between normalized acceleration and strains is not very easy to predict. Moreover, when the power spectral densities are investigated, it can be observed that some samples do not contain all frequency components under Type III loading. However, all TRAC values of the Type III loading are above 0.5, which indicates good generalization of the designed network architecture. Moreover, as expected, it can be seen that TRAC numbers are slightly higher when the trained sensor location is closer to the estimated strain locations.

Figure 9 illustrates the target and predicted strain time series for a representative sample selected from each loading case. These plots include denormalized strain sequences and predictions for the 10-min time history and zoomed time history where the maximum strain is observed. The proposed technique is able to provide strain response histories that accurately represent the actual measurements even for the unmeasured locations and the loading cases that are different than the training dataset.

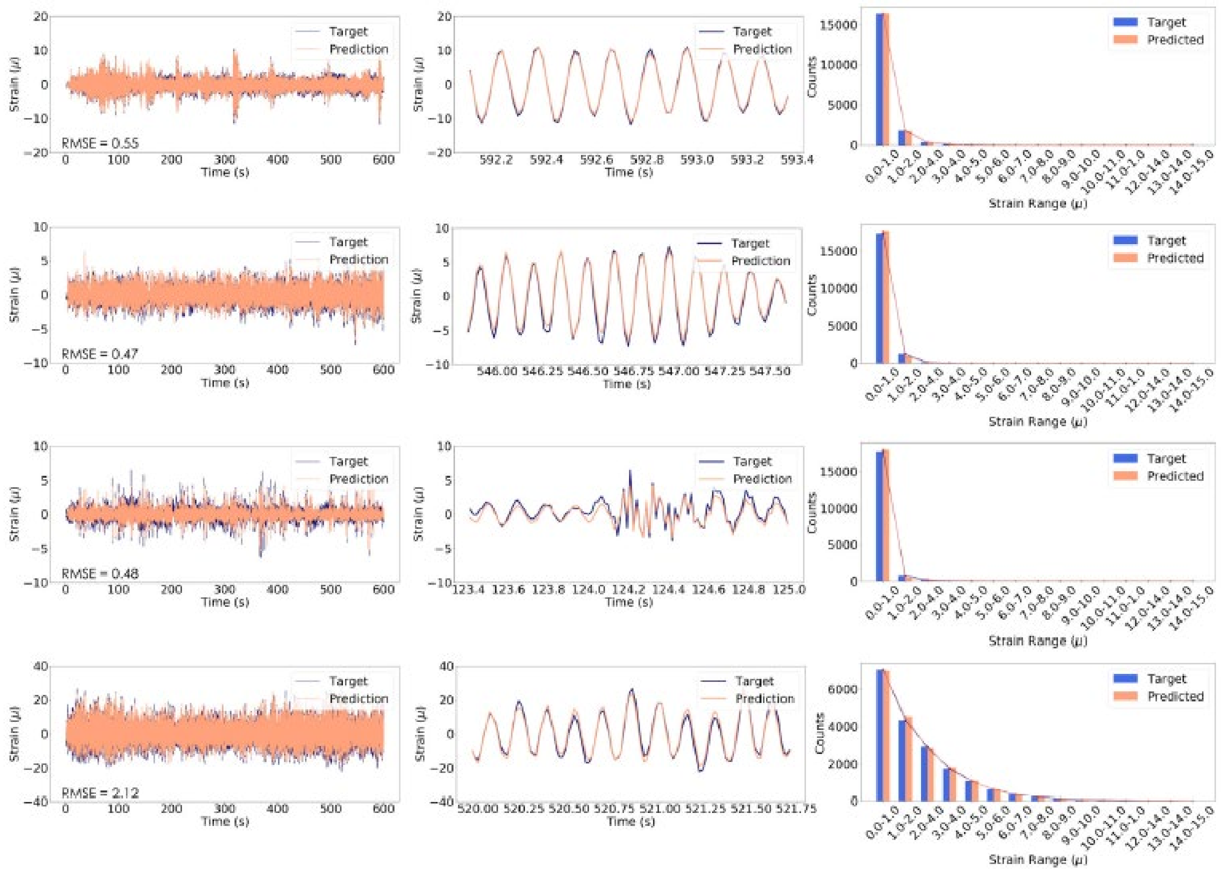


Figure 9. Strain predictions for sensor S_1 with training scenario C_4. From left to right: 10-min time history, zoomed time history where the maximum strain is observed, and rainflow counting histograms of the measured (blue) and predicted (red) strain signals.

In addition, the rainflow histograms are generated that are necessary to estimate the fatigue damage accumulation. The estimated and true rainflow histograms are compared. The histograms summarize the number of cycles for a given strain amplitude range where the bin size of $1\mu\varepsilon$ is employed. Good agreement is obtained between true and estimated counts for all load cases. The fatigue life of the specimen is calculated using Miner's Rule and the fatigue life of the specimen is determined as infinite for both target and prediction.

The performance of three-step training is compared with training using only Step B. The root mean squared error (RMSE) of the testing dataset is computed for both approaches. While training with Step B, LSTM network parameters are initialized by zeros. Figure 10 shows the average RMSE between predictions and strain gauges S_2, S_3, S_4. As there are three testing cases to estimate each strain gauge location (e.g., strains $\hat{S}_2$ can be predicted by using acceleration-strain pairs ($S_1 - A_1$, $S_3 - A_3$, $S_4 - A_4$)), the average of these cases is reported. According to the figure, RMSE of the predictions is smaller than $1\mu\varepsilon$. Additionally, it can be seen that average RMSE values are reduced by 5–10% by introducing Steps A and C, as the information from continuous dynamical behavior of the structures helps initialize the parameters more accurately and improve the convergence.

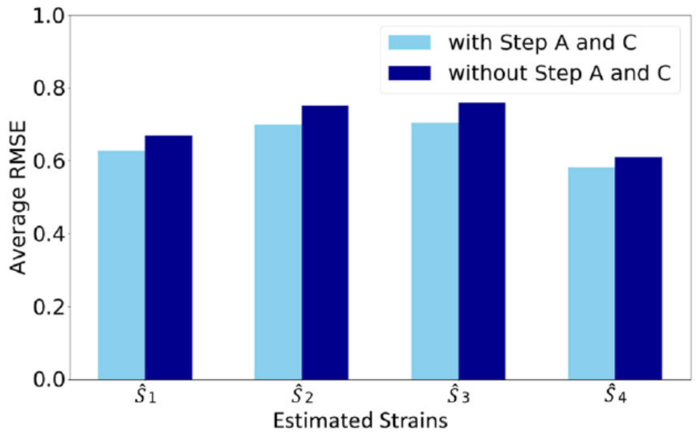


Figure 10. Root mean squared error (RMSE) of the strain predictions.

The training and testing time are also compared. The training time per iteration is stated as 0.061 and 0.053 ms for Steps A–C and only Step B, respectively. The total time for each epoch is found as 65.64 ms for the three-stage and 57.64 ms for only Step B. It can be seen that performing only Step B does not drastically reduce the training time. Also, as the good convergence is not satisfied easily, more epochs might be necessary that can increase the total training time. The time for the testing is reported as 0.826 ms for the proposed approach that is very close to utilizing only Step B (0.796 ms).

RESULTS FOR API TESTING

We tried various frequencies and shaking patterns. In Figure 11 left, we are showing the acceleration in one selected direction for LG and Samsung (SM) using only the device time-stamps. One can easily observe that both phones have different local times, shifted by many ms. Figure 11 right shows the adjusted data from LG and SM with the time offset obtained from the time server. Clearly, the data are much better aligned.

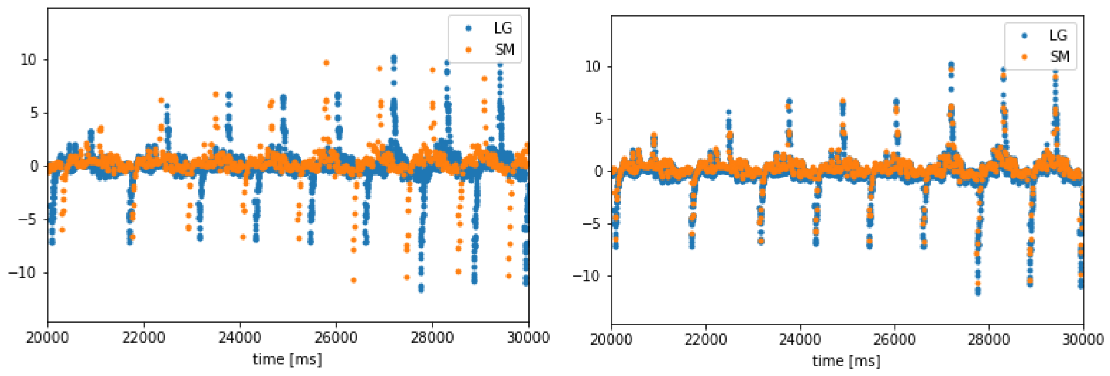


Figure 11. Comparison of (left) accelerations recorded in an LG and Samsung (SM) smartphone and (right) acceleration time histories post time synchronization.

A zoomed view of the time-aligned data is shown in Figure 12.

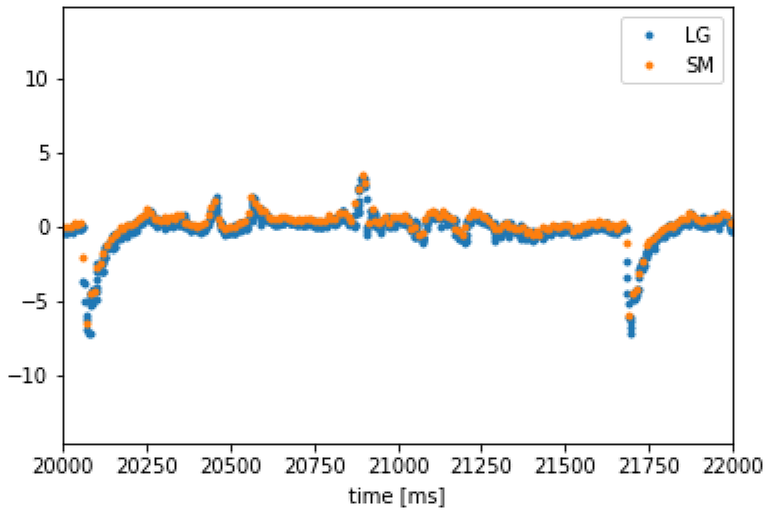


Figure 12. A zoomed view of Figure 11 right.

CHAPTER 4

Recommendations

FUTURE DIRECTIONS OF RESEARCH

In this project we demonstrate the efficacy of a DNN-based framework for estimating dynamic strain from acceleration measurements for complex bridge infrastructure. The proposed DNN-based framework attempts to establish the nonlinear map that translates acceleration to strain in the presence of noise. It accurately predicts the strain response from acceleration measurements. Furthermore, it also predicts the rainflow histograms very accurately, which enables an accurate and efficient fatigue life analysis. Additionally, we performed preliminary validation of a smartphone application that will aid in data collection.

In the future, the research team plans to further generalize the DNN framework and validate the data collection API by pursuing the following directions:

- Harness the power of further advancements in deep learning that will allow for modeling the various spatio-temporal dependencies of the problem.
- Enhance the existing networks such that with inclusion of more data, one can estimate strains in other directions.
- Augment deep learning frameworks with more involved physical principles associated with the problem at hand to enhance performance and facilitate interpretability from a physical standpoint.
- Test the developed API in real-world scenarios; for example, collect data inside vehicles while driving over a bridge.

References

Cerda, F., Garrett, J., Bielak, J., Bhagavatula, R., and Kovacevic, J. (2010). Exploring Indirect Vehicle-Bridge Interaction for Bridge SHM. *Proc. of 5th intl. conf. on bridge maintenance, safety & management*, Phil., 696–702.

Chang, M., and Pakzad, S. N. (2013). Observer Kalman filter identification for output-only systems using interactive structural modal identification toolsuite. *Journal of Bridge Engineering*, *19*(5), 04014002.

Chen, S., and Cai, C. (2007). Equivalent wheel load approach for slender cable-stayed bridge fatigue assessment under traffic and wind: Feasibility study. *Journal of Bridge Engineering*, *12*(6), 755–764.

Dong, J. (2008). *Analytical study of horizontally curved hollow tubular flange girders*. Bethlehem, PA: Lehigh University.

Downing, S. D., and Socie, D. F. (1982). Simple rainflow counting algorithms. *Intl. Journal of Fatigue*, 4(1), 31-40.

Eshkevari, S. S., and Pakzad, S. N. (2020). Signal reconstruction from mobile sensors network using matrix completion approach. *Topics in Modal Analysis & Testing*, *8*, 61–75.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471, 2000.

Graves, A., Mohamed, A. R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on* (pp. 6645-6649). IEEE.

Kingma, D., and Ba, J. (2014). Adam: A method for stochastic optimization. Preprint, arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

LeCun, Y., and Bengio, Y. (1995). Convolutional networks for images, speech and time series. *MIT Press.*

Matarazzo, T. J., and Pakzad, S. N. (2013). Mobile Sensors in Bridge Health Monitoring. *Proc. of the 9th International Workshop on Structural Health Monitoring*, Stanford, CA, 1881-1888.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high performance deep learning library. In *Advances in Neural Information Processing Systems* (pp. 8024–8035). Curran Associates, Inc.

Putnam, E. (2010). *Design, experimental, and analytical study of a horizontally curved tubular flange girder*. Bethlehem, PA: Lehigh University.

Sen, D., and Nagarajaiah, S. (2018). Data-driven approach to structural health monitoring using statistical learning algorithms. In *Mechatronics for Cultural Heritage and Civil Eng.* (pp. 295-305). Springer, Cham.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).

Yang, Y.-B., Lin, C. W., and Yau, J. D. (2004). Extracting bridge frequencies from the dynamic response of a passing vehicle. *Journal of Sound and Vibration*, 272(3-5), 471–493.