5-2024

Machine Learning Predictive Models for Load Identification of Railroad Bridges Subjected to Train-Induced Vibration

Md Masnun Rahman
The University of Texas Rio Grande Valley

MACHINE LEARNING PREDICTIVE MODELS FOR LOAD IDENTIFICATION OF RAILROAD BRIDGES SUBJECTED TO TRAIN-INDUCED VIBRATION

A Thesis

by

MD MASNUN RAHMAN

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Civil Engineering

The University of Texas Rio Grande Valley

May 2024

MACHINE LEARNING PREDICTIVE MODELS FOR LOAD IDENTIFICATION OF RAILROAD BRIDGES SUBJECTED TO TRAIN-INDUCED VIBRATION

A Thesis
by
MD MASNUN RAHMAN

COMMITTEE MEMBERS

Dr. Mohsen Amjadian
Chair of Committee

Dr. Philip Park
Committee Member

Dr. Constantine M. Tarawneh
Committee Member

Dr. Fatemeh Nazari
Committee Member

May 2024

ABSTRACT

Rahman, Md Masnun, ML Predictive Models for Load Identification of Railroad Bridges Subjected to Train-Induced Vibration. Master of Science (MS), May 2024, 77 pp., 2 tables, 29 figures, 44 references, 6 titles.

The U.S. rail network is among the world's largest, safest, and most efficient. However, many its railway bridges, which play an essential role in its connectivity, are over 100 years old, showing signs of structural deterioration exceeding their practical service life. The increased weight and speed of modern trains also pose a risk to these bridges. Traditional load identification and monitoring technologies like Weigh-in-Motion (WIM) systems are rarely used on railway bridges due to their cost and practical limits. This thesis explores a cost-effective approach to utilize vibration sensors on railway bridges to identify and predict train load features (speed and axle load), using machine learning (ML). The study involves finite element modeling of railway bridges, followed by applying ML algorithms like LSTM and RNN to estimate axle loads and speeds from vibration data. The results show the acceptable performance of developed time series ML models in predicting these parameters.

DEDICATION

This master's thesis is dedicated to my parents and siblings, whose unwavering love and encouragement have been my driving force. My mother, Laila Parvin, my father, Md Mobassar Ali, and my siblings, Md Manjurul Alam and Md Mehesum Rahman, are the reasons I am here today. Their belief in my abilities and the sacrifices made for my education inspire me every day.

I dedicate this work to all individuals striving for knowledge and positive change. May this thesis contribute, even in a small way, to the collective pursuit of understanding and progress.

To my family, friends, and mentors, thank you for your unwavering support and belief in me.

ACKNOWLEDGMENTS

The joy of finishing a big task isn't complete without thanking the people who helped. They guided and encouraged me all along, making my success possible.

I am immensely grateful to Dr. Mohsen Amjadian, chair of my thesis committee and my supervisor for the master's program, for his unwavering support, professional advice, mentoring, scholarly guidance, and critical discussions. From research design and data processing to manuscript editing, he continuously encouraged me to navigate this process with his infinite patience and guidance. His constant scrutiny and suggestions supported me during critical stages.

My sincere thanks and deep gratitude also go to my thesis committee members: Dr. Philip Park, Dr. Constantine M. Tarawneh, and Dr. Fatemeh Nazari. Their advice, input, and feedback on my thesis significantly contributed to ensuring the quality of my intellectual work.

I am also grateful for the support and assistance provided by The University of Texas Rio Grande Valley (UTRGV) through the Graduate Research Assistantship. Additionally, I extend my thanks to Dr. Constantine M. Tarawneh for facilitating my receipt of the prestigious Dwight David Eisenhower Transportation Fellowship (DDETFP) from the U.S. Department of Transportation and Federal Highway Administration.

Finally, I want to express my gratitude to my colleagues and friends at the University of Texas Rio Grande Valley who supported me directly and indirectly by providing information and documents for my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1. Introduction

The railroad network in the United States, which is among the largest and most efficient in the world, plays a crucial role in the nation's infrastructure, serving as a key component in both freight and passenger transportation. However, a significant portion of this network relies on bridges that are over a century old. With the passage of time and the evolution of rail transport, these aging structures are increasingly subjected to stresses beyond their original design parameters.

1.2. Structural Health Monitoring

With the advancement of modern computational power and technology, techniques such as structural health monitoring are becoming increasingly important, particularly in the context of railways and train infrastructure (Barke and Chiu 2005). Among these technologies, accelerometer and piezoelectric sensor systems are particularly well-suited for monitoring the health of rail structures (Hodge et al. 2014). These sensors can effectively measure vibrations and stresses on rail tracks and bridges, providing critical data for maintenance and safety assessments. Additionally, for comprehensive inspection of rail infrastructure, technologies like LiDAR and infrared thermography are highly beneficial (Taheri Andani et al. 2018). They can

detect structural damage along the tracks or on bridges that might not be visible otherwise. Ultrasonic testing and acoustic emission techniques are also valuable in this area,

especially for identifying internal flaws or cracks in rails, which are crucial for preventing accidents and ensuring smooth operation (Mariani et al. 2017). These advanced technologies are key to maintaining the safety, efficiency, and longevity of railway infrastructure.

The techniques mentioned above primarily concentrate on monitoring the condition of rail tracks and bridge infrastructures. However, sometimes it becomes more critical to monitor the load imposed by modern trains and their speed. This is because understanding the weight and speed of trains directly helps in assessing whether a particular train is suitable for a specific bridge. This kind of monitoring is crucial in preventing bridge collapse due to heavy loads or high speeds that the bridge was not originally designed to withstand. By accurately measuring these factors, engineers can ensure that the structural integrity of bridges is not compromised, thereby maintaining safety and preventing potential disasters. This approach is particularly important in the context of older bridges, which may not have been built to accommodate the increased demands of contemporary train traffic.

1.3. Weigh-in-Motion (WIM) Systems

To address the above challenges, a method already established and operational on highways can provide valuable insights. On many highways, systems are in place to limit and monitor the weight of vehicles to protect road integrity and safety. One such system is the Weigh-in-Motion (WIM) System, which has been designed primarily for highway use.

WIM Systems are advanced technological setups that measure the weight of vehicles as they pass over a sensor at normal driving speeds. Unlike traditional static weighing stations, where vehicles must stop to be weighed, WIM systems allow for continuous traffic flow and real-time data collection. These systems typically consist of various sensors embedded in the roadway, such as piezoelectric cables or load cells, which detect and record the weight of a vehicle's axles and total weight as it moves over them. The data from these sensors are then processed by computer systems to calculate the vehicle's weight.

Recently, WIM technology has been adapted for use in railway infrastructures (Ye, Su, and Han 2014), taking cues from its highway counterparts but tailored to the specific needs of railroads. In this setup, sensors like strain gauges or load cells are installed on the tracks to measure the weight of trains in motion. These sensors accurately capture the weight of each axle as the train passes over, providing essential data on the train's load. However, the effective functioning of this system is primarily limited to ballasted tracks, where the railway tracks are connected directly to the ground. This limitation arises because the strain gauge and piezoelectric sensors require a solid and flat surface for optimal operation. This presents a challenge when it comes to railway bridges, where installing these sensors is more complex due to the different structural characteristics of railway bridges. As of now, there are no fully promising technologies available that can accurately and reliably predict the load on a railway bridge, making it a significant area in need of further research and technological development. This gap highlights the necessity for innovative solutions tailored specifically for railway bridge monitoring to ensure the safety and integrity of these critical structures in the railway network.

1.4. Objectives and Research Scope

This research aims to utilize Machine Learning (ML) and Artificial Intelligence (AI) for analyzing and predicting train loads and speeds from data collected via affordable sensors such as accelerometers and strain gauges on railway bridges. By integrating sensor data with advanced ML techniques, including Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN), the study focuses on accurate load identification and speed measurement. The project involves modeling both simple and complex railway bridge structures using analysis software like OpenSEESPy and SAP2000, examining their response to various train speeds and axles loads. The findings are expected to enhance real-time monitoring and maintenance practices, thereby improving the safety and efficiency of railway systems, especially on older bridges that are challenged by heavier, faster trains.

CHAPTER II

LITERATURE REVIEW

2.1. Load Identification as an Inverse Problem

The inverse problem in load identification is about figuring out the unseen forces or loads on a structure by observing how the structure moves or responds. Unlike the direct approach of measuring forces, which can be difficult or not possible, this method works backward: starting with the outcomes (the structure's responses) to find the causes (the forces).

This issue is called "inverse" because it flips the usual way of analyzing problems. In a typical "forward" problem, we would know the forces applied and try to predict the structure's reaction. However, in real-life situations like monitoring the condition of bridges or understanding the effects of moving trains on railway bridges, the actual forces can't be directly seen. What can be observed are the structure's responses, such as vibrations or changes in materials (Xu et al. 2018; Yang et al. 2004).

Solving this inverse problem is challenging and requires complex analysis or computer-based methods. The relationship between the forces and their effects can be complicated, especially for structures that don't behave in simple, linear ways. This task is made even harder

by issues like noise in the measurements, dealing with several unknown forces at once, and the unique characteristics of the materials and design of the structure (Mousavi, Holloway, and Olivier 2019). There are studies built on existing theories to tackle the inverse problem by aiming to reduce the difference between what's observed and what models predict. The problem's complexity is also due to its "ill-posed" nature, meaning it's tricky because of incomplete information about the structure's condition or initial state. In addressing the intricate inverse problems inherent to the load identification of railroad bridges subjected to train-induced vibration, these studies integrate sophisticated ML techniques, building upon recent advancements in AI. These methods include Gaussian process-based approaches (Su et al. 2021a, 2021a; Wan and Ni 2018; Wang et al. 2021). In addition, methods like Quantile Random Forest (QRF) (Zhang et al. 2021a), Artificial Neural Networks (Yang et al. 2004; Zhong-Xian, Feng, and Bo 2008a), and Genetic Algorithms (GA) (Jiang, Au, and Cheung 2003a), each contributing uniquely to the solution of complex inverse problems through their predictive capabilities and computational efficiency.

Specifically, Zhang et al. (Zhang et al. 2021b) introduced a probabilistic model that uses QRF in conjunction with Bayesian optimization, aiming at the prediction of typhoon-induced responses in long-span bridges. This model is distinguished by its superior prediction accuracy and reduced computational demands when compared to traditional optimization algorithms and models. Similarly, Yang et al. (Yang, Yan, and He 2016a) employed a Back Propagation (BP) neural network to precisely identify characteristics such as the position, wheelbase, and weight of vehicles on bridges. Their method demonstrated both feasibility and high accuracy in the identification of moving loads at uniform speeds. Further, Li et al. (Zhong-Xian, Feng, and Bo

2008b) explored the use of neural networks to enhance the reliability and accuracy of load identification, underscoring the potential of these ML strategies to revolutionize traditional approaches to structural monitoring and analysis.

In the case of a Simply Supported Beam (SSB) treated as a bridge subjected to a single point load moving at a constant speed, the dynamic response can be accurately described by the classic Duhamel integral if the beam's response is linear (Yang et al. 2021a). The Duhamel integral is generally given by:

$$y(t) = \int_0^t h(t-\tau)f(\tau)d\tau \tag{2.1}$$

In this equation, $y(t)$ represents the dynamic response of the beam at time t, $h(t-\tau)$ is the unit impulse response function, which characterizes how the beam responds over time to a load applied at a single instant, and $f(\tau)$ is the magnitude of the applied load at time τ. The integral in Equation (2.1) accumulates the effects of the load over time up to the current time t.

For a beam that does not conform to linear or time-invariant principles, a different function, say F_{ij}, is required to model the relationship between the input (the load) and the output (the response). This is especially true for nonlinear structures, where the dynamic response to a load can change over time or depending on the location of the load. The mapping function can be expressed as (Thomas, Wilson, and Wilson 1973):

$$y^j(t) = \boldsymbol{\mathcal{F}}^{i,j}\left[f^i(t)\right] \quad i,j = 1,2,\dots N \tag{2.2}$$

where y^j denotes the rate of change of the dynamic response at point j on the beam, $\boldsymbol{\mathcal{F}}^{i,j}$ is the transfer function that maps the relationship between the rate of change of the applied load, and i and j indicate specific points or degrees of freedom on the beam.

The inverse problem aims to find an inverse function M_{ij} that can estimate the applied load based on the observed response, within a certain error threshold:

$$\boldsymbol{M}^{i,j}\left[y^j(t)\right] = \tilde{f}(t) \approx \boldsymbol{f}^i(t) \text{ and } |\boldsymbol{f}^i(t) - \tilde{f}(t)| < \varepsilon \tag{2.2}$$

where *i, j = 1, 2,…N*. The challenge of deriving M_{ij} for nonlinear structures necessitates the use of more sophisticated and reliable techniques. This study introduces the use of LSTM networks, a type of RNN capable of capturing temporal dependencies, to approximate M_{ij}. With extensive supervised learning, the LSTM network is trained to predict the dynamic responses of the beam, effectively substituting the need for an explicit analytical expression of the inverse function. This approach allows for a more feasible solution to the inverse problem in scenarios where direct measurement of the load is not possible.

2.2. Traditional Load Identification Methods

Accurately measuring and predicting dynamic loads like train impacts on railway bridges is challenging in structural engineering. Direct measurement is often impractical due to limitations in testing technology and the complexity of large structures, which makes sensor installation difficult. As a result, identifying different types of dynamic loads has become a complex issue discussed extensively in the literature (Yang et al. 2021a).

Traditional methods mainly focus on first determining a structure's dynamic characteristics, a task constrained by technological limitations and the intricate nature of these structures. In addition, these conventional approaches typically assume linear and predictable load responses, overlooking real-world nonlinear behaviors. For example, the vibrations from

high-speed trains demonstrate the inadequacy of traditional models to account for complex dynamic interactions within structures, potentially compromising infrastructure safety. This highlights the urgent need for more advanced methodologies to accurately predict and manage dynamic loads (Jiang et al. 2020).

In the context of this, this paper aims to explore the theory of moving load identification in the time domain on a continuous beam, without prior knowledge of the load characteristics. Understanding the forces from vehicle-bridge interactions is critical for bridge design, as they significantly contribute to the live load. However, direct measurements can be costly and biased, and simulations may have modeling errors. Therefore, there is a need for a method to indirectly determine vehicular loads by analyzing the vibration responses of the bridge structure such as accelerations, velocities, displacements, and strains. This approach should account for the various parameters of the bridge and vehicle system in the measured responses, offering a more accurate and reliable means of assessing dynamic loads (Zhu and Law 2002).

2.3. Load Identification using AI

Methods for identifying dynamic loads have developed in three main categories: 1) frequency domain techniques, 2) time domain techniques, and 3) intelligent algorithms.

Frequency-domain methods are the earliest and are considered the classical methods in this field. They primarily use direct inversion, least squares, or modal coordinate transformation techniques. These methods require inverting the frequency response function matrix to identify loads. However, matrix inversion often experiences significant ill-conditioning problems that

degrade solution accuracy (Zhang and Zhou 2023). Despite advancements in various techniques, industry professionals often favor frequency-domain methods for their maturity and reliability, making them a popular choice for identifying excitations like wind load, six-force-factor, and loads on mining machinery in numerous engineering applications (Hwang et al. 2016; Wang, Yang, and Xiao 2013; Wang et al. 2013; Zhang and Li 1986).

Time-domain methods on the other hand directly incorporate time as a variable, making them more intuitive. The general procedure first uses a structure's model parameters to establish an inverse system model. Then, the input is identified from the system output(Jie Liu et al. 2016). Many common time-domain methods employ modal decomposition or Duhamel integral techniques (CHE, Liu, and Jiang 2013; Li, Jiang, and Mohamed 2021).

However, several factors limit the accuracy of dynamic load identification. These methods can experience ill-posedness, cumulative errors, and unclear system parameters (Beghini et al. 2023). Noise, complex structures, repeated frequencies, and resonance issues also substantially impact the accuracy of load identification.

One of the early works in this field is the paper presented by O'Connor and Chan (1988), who introduced an interpretive time domain method to detect moving loads from bridge strain data. By modeling the bridge with lumped masses linked by massless elastic beams and analyzing the strains in a time series, they identified the loads (O'Connor and Chan 1988). Law et al. (1997, 1999) derived exact solutions for detecting moving forces on SSB (Law, Chan, and Zeng 1999). Chan et al. (1999) identified axle loads of a two-axle vehicle model using simulated

bridge responses (Chan et al. 1999). Zhu and Law (1999) combined modal superposition and optimization to locate forces on continuous multi span bridges (Zhu and Law 1999). However, these methods showed fluctuating force histories and noise sensitivity due to the ill-posed nature of the problem. To address this, Law et al. (1997, 1999) applied regularization techniques to time domain methods (Law et al. 2001), greatly improving result accuracy (Zhu and Law 1999). Regularization methods, notably direct techniques like Tikhonov regularization, play a pivotal role in stabilizing ill-posed problems by imposing constraints that guarantee stable and unique solutions. These strategies effectively balance the accuracy of data representation against the smoothness of the solution, thereby significantly boosting the dependability of mathematical models in various practical scenarios. Jacquelin (2003) (Jacquelin, Bennani, and Hamelin 2003) highlighted that while these methods are broadly applicable and stable, with rapid convergence rates, identifying suitable values for regularization parameters remains a challenge. Moreover, the Tikhonov regularization method, alongside Singular Value Decomposition (SVD) techniques, faces computational inefficiencies when applied to large-scale inverse problems. To address these limitations, several indirect regularization approaches have been developed, including the Landweber regularization method (Jang et al. 2011) and the conjugate gradient method (Jang 2013), offering alternative solutions for handling complex, large-scale computational challenges.

The advancement of artificial intelligence has significantly impacted the field of structural engineering, particularly in addressing complex inverse problems like dynamic load identification. Utilizing a combination of ML and deep learning techniques has yielded notable

improvements in prediction accuracy, computational efficiency, and the management of time-varying data characteristics.

ML algorithms such as Gaussian process-based methods (Kyprioti and Taflanidis 2021; Su et al. 2021b, 2021b; Wan and Ni 2018; Wang, Liu, and Bian 2022). Quantile Random Forest (QRF) (Zhang et al. 2021c), and Genetic Algorithms (GA) (Jiang, Au, and Cheung 2003b) have proven effective in predicting structural responses under variable loads, like those from typhoons on long-span bridges. For instance, a model combining QRF and Bayesian optimization proposed by Zhang et al. (Zhang et al. 2021d) demonstrated enhanced prediction accuracy with reduced computational demands. ANN with early applications by Cao et al. (1998) (Cao, Sugiyama, and Mitsui 1998) in aircraft wing load identification laying the groundwork for future advancements. Back Propagation (BP) ANNs have been employed for accurately identifying vehicle characteristics on bridges (Yang, Yan, and He 2016b; Yang et al. 2016b; Zhong-Xian et al. 2008a), though they face challenges such as slow convergence and susceptibility to local minima. The adoption of RNN models has addressed the need to capture the dynamic nature of traffic loads, offering an improved understanding of time sequence data despite challenges with long-term temporal correlations (Wu et al. 2018a). The field has seen a shift towards more sophisticated deep learning approaches to enhance the identification process. Support vector regression (Liu, Wang, and Gu 2021), deep regression adaptation networks (Wang et al. 2021), and deep RNNs (Zhou et al. 2019a) represent a leap forward in tackling dynamic load challenges with higher accuracy and efficiency. Deep Learning's role in dynamic load identification is viewed as a regression problem, with the model choice tailored to specific data features and objectives. Techniques ranging from Multilayer Perceptron (MLP) for table data, Convolutional

Neural Networks (CNNs) for image processing, to Support Vector Machines (SVMs) for limited sample learning, showcase the adaptability and potential of intelligent algorithms in structural analysis (Zhou et al. 2019b).

This application of machine and deep learning methods marks a promising direction for solving inverse problems in engineering. It highlights the potential for more accurate, efficient, and adaptable solutions in structural analysis and beyond, paving the way for innovative applications in the field.

2.4. Application of RNN and LSTM in Load Identification

The incorporation of advanced ANN models, specifically RNN and their enhanced variant, LSTM networks, has marked a significant evolution. These models address the intricate problem of capturing and analyzing time-varying characteristics inherent in structural loads, utilizing their unique architectures to manage the sequence data effectively. RNNs are fundamentally designed to handle sequential data like time series and sensors signals, making them useful for dynamic load identification, a problem characterized by its time series nature. However, conventional RNNs encounter significant challenges, such as gradient vanishing or exploding, which impair their ability to learn long-term dependencies within the data (Ni et al. 2016, 2016; Wu et al. 2018b). This limitation is crucial in structural engineering, where the temporal dimension of load data can significantly influence prediction accuracy and model reliability.

To mitigate these issues, LSTM networks introduce a sophisticated memory mechanism that includes three specific gates: the input, forget, and output gates. This design allows LSTMs to selectively remember and forget information, a feature that significantly enhances their capacity to model long-term temporal dependencies without succumbing to the gradient problems that plague traditional RNNs (Hochreiter and Schmidhuber 1997; Wu et al. 2018c). The introduction of bidirectional LSTMs (BLSTMs) further extends this capability by processing data in both forward and backward directions, thereby capturing dependencies across the entire time series more effectively (Graves and Schmidhuber 2005a).

The application of LSTM networks transcends dynamic load identification. Initially developed for natural language processing, LSTMs have found utility in diverse fields, demonstrating their versatility and effectiveness. For instance, they have been employed in phoneme classification (Graves and Schmidhuber 2005b), human activity recognition (Ordóñez and Roggen 2016), traffic flow prediction (Jiang et al. 2016), sequential data reliability learning (Jun Liu et al. 2016), and financial market trend analysis (Li, Zhang, and Zhao 2019). This wide range of applications underscores the LSTM's robustness and adaptability in handling complex time series data across different domains.

In the context of structural engineering, LSTMs offer a data-driven approach to inverse problem-solving, avoiding the simplifications and assumptions inherent in traditional physical models. Yang et al. present a robust RNN with LSTM for dynamic load identification on SSBs, achieving high accuracy with position error rates of 1.27% for sinusoidal and 1.26% for random excitations, and signal-to-noise ratios of 36.42 and 46.28. The model, featuring a BLSTM and

two fully connected layers, demonstrates adaptability and minimal sensitivity to sensor placement, advancing applications in structural dynamics and ML (Yang et al. 2021b).

A similar concept applied by Zhou et al. to predict time series problems utilized a deep RNN equipped with LSTM and BLSTM layers for dynamic impact load identification on nonlinear structures. Their approach, validated by an extensive dataset including training, validation, and testing sets, achieved stable performance metrics such as mean peak error, mean impulse error, and mean root-mean-square error (RMSE), demonstrating the method's insensitivity to sensor placement and hyperparameter variations.

Zhang and Zhou (2023) developed an AI-driven methodology using an array of lead zirconium titanate sensors and LSTM networks to identify moving loads on bridges, aiming to improve structural management and maintenance. The system's dataset includes simulated moving speeds and loads, achieving identification with high accuracy, demonstrated by mean absolute errors as low as 0.015 m/s for speed and 0.335 N for magnitude, confirming its efficacy in real-time traffic load monitoring (Zhang and Zhou 2023).

The integration of RNN and LSTM models into dynamic load identification represents a paradigm shift towards more intelligent, efficient, and accurate analysis techniques in structural engineering. Their ability to process and learn from time series data in a nuanced manner offers a promising avenue for enhancing the predictive modeling and understanding of complex structural behaviors under variable loading conditions.

CHAPTER III

BASIC THEORY OF LSTM NETWORK AND APPLICATION IN INVERSE PROBLEM

3.1. RNN Network

Recurrent Neural Networks (RNNs) are particularly well-suited for modeling sequential data, such as time series from accelerometers for load prediction. The unique architecture of RNNs, which includes feedback loops, enables them to maintain a state or 'memory' of previous inputs, allowing for the retention and utilization of historical data over time. This characteristic is illustrated in Figure 1, which depicts the process of unfolding an RNN. The figure transitions from a single, compact RNN representation on the left to an extended sequence that showcases the network's operation over discrete time steps, labeled from 1 to t. Each time step consists of an input node (in grey), a hidden node (in green), and an output node (in orange), with arrows indicating the flow of information. Notably, the hidden nodes pass information both to the output and to the next time step, highlighting the recurrent nature of the network. The unfolding of the RNN across time steps demonstrates the network's capability to integrate current input with past information to make informed predictions, a process vital for complex tasks like load forecasting in dynamic environments (Yang et al. 2021c):

$$H(t) = W_{hi} \cdot X(t) + W_{hh} \cdot S(t-1) + b_h \tag{3.1}$$

$$S(t) = f(H(t)) \tag{3.2}$$

$$O(t) = W_{\text{oh}} \cdot S(t) \tag{3.3}$$

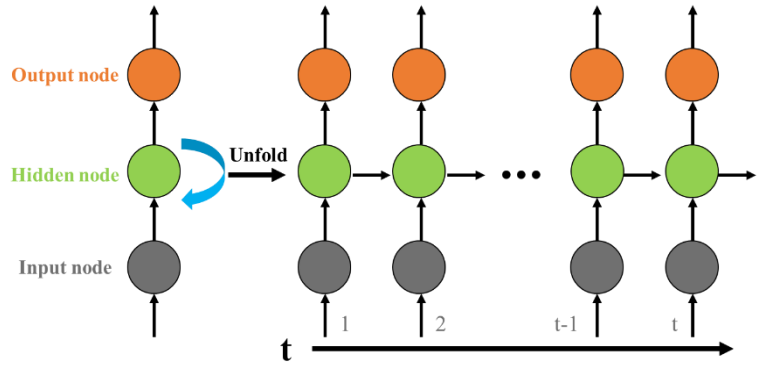


Figure 1: Traditional RNN cell architectures for sequential data processing (Alzubaidi et al. 2021)

In these equations, *X(t)* represents the data input at time t, such as readings from an accelerometer. $H(t)$ is a combination of the input and the previous hidden state, reflecting the network's memory. The nonlinear function f creates the hidden state $S(t)$, which in turn determines the output $O(t)$. This structure allows RNNs to capture and process the temporal patterns inherent in sequential data.

The recurrent connections in the hidden layer are what set RNNs apart. They enable the network to adapt to various sequence lengths, making them flexible and capable of modeling complex dynamics. This adaptability is crucial for predicting loads from time series data, where capturing the flow and trends in the sequence is key.

3.2. LSTM Network

LSTM networks have emerged as an evolution of RNNs, addressing the shortcomings of their precursor in handling sequential data. Originating from the multi-layer perceptron, RNNs were designed with the capability to process sequences by utilizing a hidden state as memory, allowing them to preserve temporal dependencies. The structure of a simple RNN features loops where the output of the hidden layer at one time step becomes part of the input at the subsequent step (Van Houdt, Mosquera, and Nápoles 2020). However, the vanishing gradient problem, as identified by Bengio et al. (1994), presents a significant obstacle to neural network training, particularly in learning long-term dependencies within extensive time series data (Lipton, Berkowitz, and Elkan 2015). This issue hinders the network's ability to effectively propagate information through many layers, affecting its learning capability for tasks requiring memory of long sequences. especially for the long-time series. Figure 2 is a standard LSTM cell where x_t is the input information at time t, h_t is the output information at time t, and the hidden layer A transports the information at time t-1 to time t recurrently in order to maintain the previous information at the present moment.

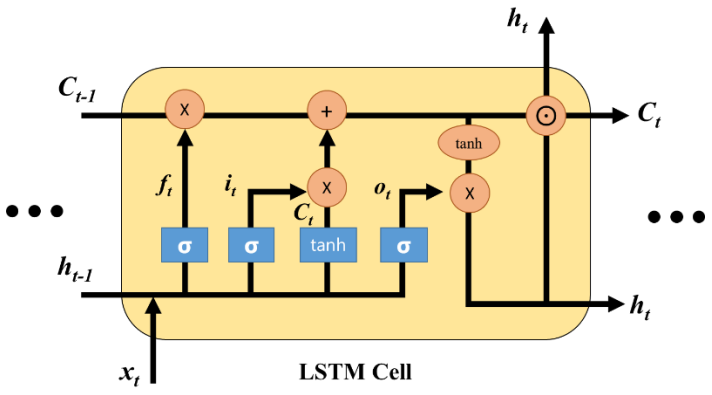


Figure 2: LSTM cell architectures for sequential data processing (Liu 2020)

Figure 2 shows the process of a traditional RNN which can be expressed mathematically as follows:

$$h_t = tanh(W_h W_{t-1} + W_x x_t + b) \tag{3.4}$$

where tanh(.) is an activation function defined as follows:

$$tanh\ (x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.5}$$

LSTM networks were conceived to overcome this limitation of the vanishing gradient problem by introducing a more sophisticated hidden layer structure. Hochreiter and Schmidhuber (1997) introduced mechanisms within the LSTM that enable the network to selectively retain or discard information through a series of gated operations (Hochreiter and Schmidhuber 1997). A typical LSTM unit comprises a memory cell state that traverses the entire chain, along with three gates: the forget gate, input gate, and output gate (Gers, Schmidhuber, and Cummins 2000).

The memory cell state is central to the LSTM's design, maintaining the flow of information and ensuring the network's capability to handle long sequences effectively. Each gate within the LSTM performs distinct functions as follows: Forget Gate *(f_t)*, Input Gate *(i_t)*, Cell State Update *(c_t)*, Output Gate *(o_t)*.

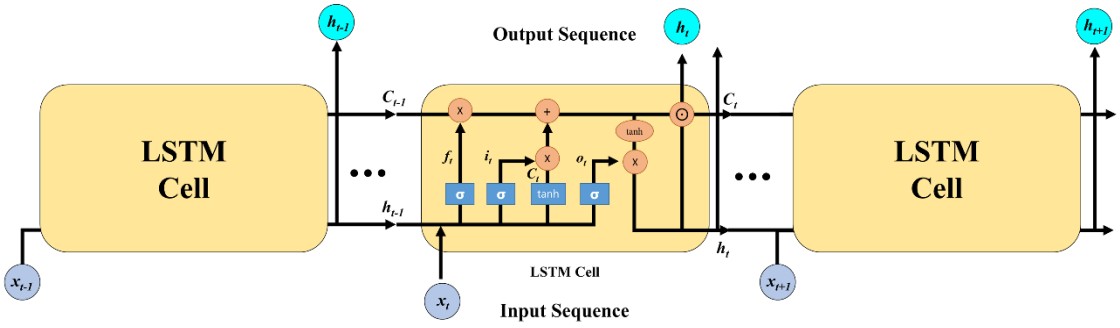


Figure 3: Schematic diagram of LSTM cell operations and information flow (Sanatel and Günel 2023)

3.2.1. Forget Gate *(f_t)*

The forget gate in an LSTM cell is responsible for determining which information from the cell state is retained or discarded as the network processes data. It is a crucial component that helps in mitigating the vanishing gradient problem commonly faced in traditional RNNs by selectively maintaining relevant information through long sequences (Yang et al. 2021c).

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (3.6)$$

f_t is the output of the forget gate at time step, σ is the sigmoid activation function that outputs a value between 0 and 1, W_{xf} is the weight matrix associated with the input, x_t is input vector at the current time step, W_{hf} is weight matrix associated with the hidden state, h_{t-1} is hidden state vector from the previous time step, and b_f is bias term for the forget gate.

The output f_t of the forget gate is a vector with values between 0 and 1. This output is element-wise multiplied by the previous cell state c_{t-1} to determine which information is to be retained (values close to 1) or forgotten (values close to 0). This mechanism allows the LSTM to preserve long-term dependencies while removing information that is no longer relevant for future steps.

3.2.2. Input Gate (i_t) and Candidate Layer (g_t)

The input gate in an LSTM cell is denoted as $\mathrm{i_t}$ and it plays a pivotal role in updating the cell state with new information. The gate uses a sigmoid activation function to carefully decide which values will be allowed to pass through (Yang et al. 2021c). The mathematical representation is:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3.7)$$

In this equation, W_{xi} is the weight matrix that interacts with the input at the current time step (x_t), W_{hi} is the weight matrix applied to the output of the previous time step's hidden state, and b_i is s the bias term, adding an offset to the gate's activity.

In conjunction with the input gate, the candidate layer, represented by:

$$g_t = \varphi\left(W_{xg}x_t + W_{hg}h_{t-1} + b_g\right) \tag{3.8}$$

φ typically signifies the tanh activation function, which outputs values in the range of -1 to 1, providing a regulated input to the cell state. In Equation (3.8), W_{xg} and W_{hg} are weight matrices for the input vector x_t and the previous hidden state h_{t-1} respectively, b_g is the bias term for the candidate layer.

The output of the input gate i_t and the candidate layer g_t are crucial components in updating the cell state, combining to form a filtered addition to the cell state. This process allows the network to maintain a memory of previous inputs in a controlled manner, contributing significantly to the LSTM's ability to capture long-term dependencies in data.

3.2.3. Cell State Update (c_t):

The cell state update is a fundamental part of an LSTM cell's functionality, representing the cell's memory at each time step. The cell state, denoted as c_t, is updated by combining past information, held in the previous cell state c_{t-1}, with new candidate values proposed by the candidate layer g_t as shown below,

$$c_t = f_t \odot c_{t\text{-}1} + i_t \odot g_t \tag{3.9}$$

where c_t is the new cell state at the current time step, f_t is the output of the forget gate at the current time step t, applied to the previous cell state, c_{t-1} is the previous cell state from the time step $t-1$, and i_t is the output of the input gate at the current time step t, regulating the addition of new information. In Equation (3.9), g_t is the vector of candidate values created by the candidate layer, which could be added to the cell state, and $\odot$ is the Element-wise multiplication, also known as the Hadamard product, ensuring that each element of the forget gate output and the input gate output is applied to corresponding elements of the previous cell state and the candidate values, respectively.

The update process of c_t is critical for the LSTM's ability to carry relevant information throughout the processing of sequences, allowing the network to maintain long-term dependencies. This is particularly useful in tasks where context from earlier in the sequence is necessary to understand later elements, such as in language modeling or time-series prediction. The careful modulation of information flow by the forget and input gates allows the LSTM to manage its memory effectively, preventing the vanishing gradient problem and enabling the network to learn over many time steps.

3.2.4. Output Gate (o_t) and Hidden State (h_t)

The output gate and hidden state within an LSTM cell play crucial roles in determining the final output at each time step, taking into account the current cell state and the previous outputs. The output gate decides which parts of the cell state will contribute to the hidden state, which in turn influences the cell's output at the current time step and the state passed to the next time step.

The output gate's activation, o_t, is computed as follows:

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{3.10}$$

where o_t is the output of the output gate, which regulates the flow of information from the cell state to the hidden state. In this equation, W_{xo} is the weight matrix for the output gate applied to the input vector x_t at the current time step, x_t is the input vector at the current time step, W_{xo} is the weight matrix for the output gate applied to the previous hidden state h_{t-1} from the previous time step, and $\mathrm{b_o}$ is the bias term for the output gate.

The hidden state $\mathrm{h_t}$ is updated using the output of the output gate and the current cell state $\mathrm{c_t}$, passed through a $\tanh$ function to normalize its values:

$$h_t = o_t \odot tanh(c_t) \tag{3.11}$$

where $\mathrm{h_t}$ is the hidden state vector at the current time step, $\mathrm{o_t}$ is the output from the output gate, which decides how much of the cell state is passed to the hidden state, and $\tanh(\mathrm{c_t})$ is the hyperbolic tangent of the current cell state, normalizing it to values between -1 and 1. In this equation, c_t is the current cell state, representing the memory of the network up to the current time step.

The interplay between the output gate and the updated cell state to produce the hidden state allows the LSTM to effectively control the amount of memory information influencing the output and being carried forward. This mechanism enables LSTMs to make decisions based on long-term dependencies, addressing challenges in sequence prediction tasks such as language modeling, time-series analysis, and more.

CHAPTER IV

NUMERICAL MODELING OF DYNAMIC SYSTEM SUBJECTED TO MOVING LOAD

In Chapter 4, the examination of numerical modeling of dynamic systems subjected to moving loads focuses on two models: a simple beam and a bridge. These models were selected to illustrate both fundamental response characteristics and complex interactions under dynamic conditions. Through simulations, the aim is to enhance understanding and predictive capability of structural behavior, facilitating advanced analysis and design in practical engineering applications.

4.1. Simply Supported Beam (SSB)

4.1.1. Single Axle Load

In this section, we introduce and validate a Finite Element (FE) model of a Simply Supported Beam (SSB) bridge through comparison with analytical solutions. Specifically, as shown in Figure 4, we focus on a single span simply supported railroad bridge subjected to a single axle load *(P)* traveling at a constant speed *(V).* This model simplifies the bridge as a linear elastic system, assuming it has a uniform cross-section along with constant mass and flexural rigidity *(EI).*

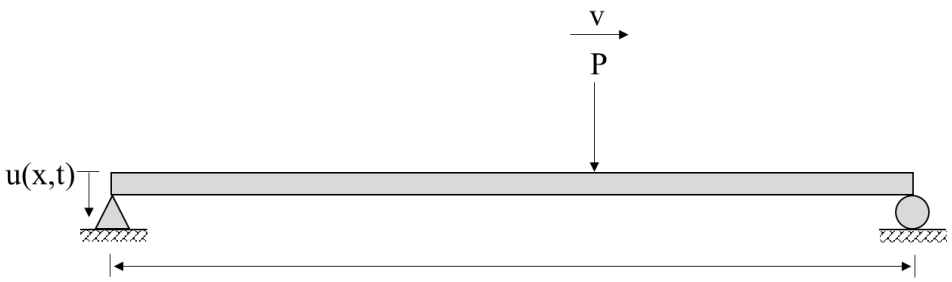


Figure 4: SSB model under transverse concentrated moving load.

The vertical displacement of the simply supported single-span bridge mentioned previously can be calculated analytically as follows. Let *u (x, t)* represent the displacement of the beam in the y-axis at position x and time *t*, with *L* denoting the bridge length, m the mass per unit length, c_e the external damping coefficient, c_i the internal damping coefficient, *E* the modulus of elasticity, and *I* the moment of inertia. To simplify the model while maintaining accuracy, we make several assumptions: (i) the bridge is homogeneous with a constant cross-section, (ii) only one moving load is present on the bridge at any time, (iii) only the moving force is considered, ignoring inertia effects, (iv) the load moves at a constant speed *v* across the entire bridge, (v) the bridge starts from a state of rest, and (vi) we disregard any surface irregularities of the bridge and track. The motion equation for a SSB subjected to a moving force *P* at a constant speed *v* is then introduced as follows (Pokhrel 2023):

$$m\ddot{u} + c_e\dot{u} + c_iI\dot{u}'''' + EIu'''' = P\delta(x - vt), \qquad 0 \le vt \le L \tag{4.1}$$

Where, primes (′) and dots (˙) denote differentiation with respect to coordinate x and time t, respectively, and δ is the Dirac delta function. The boundary conditions are.

$$u(0,t) = 0,\ u(L,t) = 0,\ EIu''(0,t) = 0,\ EIu''(L,t) = 0 \tag{4.2}$$

The beam is assumed to be at rest prior to the arrival of the moving vehicle so the initial conditions are.

$$u(x,0) = 0,\ \dot{u}(x,0) = 0 \tag{4.3}$$

The transverse deflection of the beam u (x, t) due to only the nth mode of vibration can be written as,

$$u(x,t) = \emptyset_n(x)q_n(t) \tag{4.4}$$

Where, Φ_n is the nth vibration mode which satisfies boundary conditions. $q_n(t)$ represents the generalized coordinate for the nth vibration mode.

Substituting value of Equation (4.4) in Equation (4.1), Multiplying both sides of Equation (4.1) by Φ_n and integrating with respect to x from 0 to L, the generalized equation of motion of the beam is obtained as

$$m\ddot{q}_n(t)\int_0^L [\emptyset_n(x)]^2 dx + \dot{q}_n(t) \times \left\{ c_e \int_0^L [\emptyset_n(x)]^2 dx + c_i I \int_0^L \emptyset_n{}^{\langle 4 \rangle}(x)\emptyset_n(x)dx \right\} + EIq_n(t)\int_0^L \emptyset_n{}^{\langle 4 \rangle}(x)\emptyset_n(x)dx = p\emptyset_n(vt) \tag{4.5}$$

where,

$$\int_0^L \delta(x-a)\emptyset_n(x)dx = \emptyset_n(a) \tag{4.6}$$

Let us denote the vibration frequency ω_n of the nth mode of the beam as

$$\omega_n{}^2 = \frac{EI}{m}\frac{\int_0^L \emptyset_n{}^{\langle 4 \rangle}(x)\emptyset_n(x)dx}{\int_0^L [\emptyset_n(x)]^2 dx} \tag{4.7}$$

We can write, $c_e = \alpha_e m$, and $c_i = \alpha_i E$ and ξ_n is the corresponding damping coefficient for nth mode of vibration given by,

$$\xi_n = \frac{1}{2}\left(\frac{\alpha_e}{\omega_n} + \alpha_i \omega_n\right) \tag{4.8}$$

Then Equation (4.5) becomes

$$\ddot{q}_n + 2\xi_n\omega_n\dot{q}_n + {\omega_n}^2 q_n = \frac{p\emptyset_n(vt)}{\int_0^L m[\emptyset_n(x)]^2 dx} \tag{4.9}$$

This equation of motion is valid only when the acting position vt of the moving load is located within the beam, i.e., $0 \leq vt \leq L$. Once the moving load leaves the beam, only free vibration remains.

For a simply supported beam, the n^{th} mode shape of vibration is given by,

$$\emptyset_n(x) = sin\frac{n\pi x}{L} \tag{4.10}$$

and the frequency of vibration ω_n obtained from Equation (4.7) is,

$$\omega_n = \left(\frac{n\pi}{L}\right)^2 \sqrt{\frac{EI}{m}} \tag{4.11}$$

Substituting the value into Equation (4.9) yields the equation of motion for the n^{th} mode of the SSB as

$$\ddot{q}_n + 2\xi_n\omega_n\dot{q}_n + {\omega_n}^2 q_n = \frac{2p}{mL}\sin\left(\frac{n\pi vt}{L}\right) \tag{4.12}$$

which is uncoupled from the other modes of vibration. From this equation, the generalized coordinate of q_n for the n^{th} mode can be solved as

$$\begin{aligned} q_n &= \frac{2pL^3/(EIn^4\pi^4)}{(1-{S_n}^2)^2 + (2\xi_n S_n)^2} \\ &\times \left\{ \left(1-{S_n}^2\right)\sin\Omega_n t - 2\xi_n S_n \cos\Omega_n t \right. \\ &\left. + e^{-\xi_n\omega_n t}\left[2\xi_n S_n \cos\omega_{dn} t + \frac{S_n}{\sqrt{1-{\xi_n}^2}}(2{\xi_n}^2 + {S_n}^2 - 1)\sin\omega_{dn} t \right] \right\} \end{aligned} \tag{4.13}$$

Where, ω_{dn} is the damped frequency of vibration of the beam and given by

$$\omega_{dn} = \omega_n\sqrt{1-\xi_n{}^2} \tag{4.14}$$

Ω_n is the excitation frequency.

$$\Omega_n = \frac{n\pi v}{L} \tag{4.15}$$

S_n is the non-dimensional speed parameter defined as the ratio of the frequency of excitation of the moving load to the n^{th} frequency of vibration of the beam.

$$S_n = \frac{\Omega_n}{\omega_n} = \frac{n\pi v}{\omega_n L} \tag{4.16}$$

Consequently, the total displacement u(x, t) of the beam caused by all the vibration modes can be summed as follows:

$$\begin{aligned} u(x,t) = \sum_{n=1}^{\infty} & \frac{\frac{2pL^3}{EIn^4\pi^4}}{\left(1-S_n{}^2\right)^2 + (2\xi_n S_n)^2} \\ & \times \left\{ \left(1-S_n{}^2\right)\sin\Omega_n t - 2\xi_n S_n \cos\Omega_n t \right. \\ & \left. + e^{-\xi_n\omega_n t}\left[2\xi_n S_n \cos\omega_{dn} t + \frac{S_n}{\sqrt{1-\xi_n{}^2}}\left(2\xi_n{}^2 + S_n{}^2 - 1\right)\sin\omega_{dn} t \right] \right\} \\ & \times \sin\frac{n\pi x}{L} \end{aligned} \tag{4.17}$$

This is the solution for displacement of the beam caused by a single moving load while considering the effect of damping.

In the above Equation (4.16), the terms containing $\Omega_n t$ represents the forced vibration of the bridge induced by the moving load, and the terms with $\omega_{dn} t$ are the free vibration. This

equation applies within the limit of the beam length. After the load has passed, the response is a free vibration with initial conditions equal to the conditions in the beam at the moment when the force leaves the span.

Practically, the effect of damping on the bridge is so small, due to the short acting time of the moving loads. So, it can be ignored completely. By neglecting the effect of damping, the total displacement u (x, t) can be written as;

$$u(x,t) = \frac{2pL^3}{EI\pi^4}\sum_{n=1}^{\infty}\frac{1}{n^4}\sin\frac{n\pi x}{L} \times \left(\frac{\sin\Omega_n t - S_n \sin\omega_n t}{1 - {S_n}^2}\right) \quad (4.18)$$

This is the equation of displacement of the simple beam at section x subjected to the moving load p acting at position vt without damping effect.

The result obtained for the midpoint displacement by considering the first mode only is sufficient, because all the anti-symmetric modes of vibration have zero contribution to the midpoint displacement. This gives the sense of using only the first mode which can yield generally good approximate solutions for vehicle-induced response (Yang et al., 2004).

4.1.2. Multi Axle Load

To assess the impact of various loading scenarios on railway structures, a scaled beam model with two moving point loads was utilized as shown in Figure 5. This theoretical framework allows for the addition of multiple axles, accurately simulating a full-scale train with variable axle spacings. This method enables precise representation and analysis of complex loading dynamics in railway systems, ensuring their structural integrity under diverse conditions.

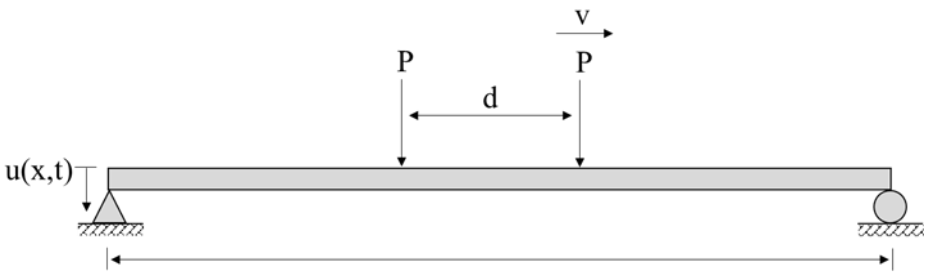


Figure 5: SSB subjected to double moving point loads (N=2)

Let's examine a simplified beam model of length L, subject to two-point loads at an axle distance d. This setup would be iteratively replicated in real-world applications to accommodate variable axle spacings. Within this context, the wheel load function *F(t)* characterizes the dynamic load distribution of a train moving along the beam. This function is crucial for determining the stress response of the beam under transient loading conditions, enabling the evaluation of structural performance and resilience. The function *F(t)* can be defined as (Pokhrel 2023):

$$F(t) = \sum_{j=1}^{N} P \times U_j(t, v, L) \tag{4.19}$$

where,

$$U_j(t, v, L) = \delta\left[x - v(t - t_j\right] \times \left[H\left(t - t_j\right) - H\left(t - t_j - \frac{L}{v}\right)\right] \tag{4.20}$$

In Equation 4.20 δ is dirac delta function, x is coordinate of the beam, H is step function, t_j is arriving time of the j^{th} load at the beam, $t_j = (j - 1)d/v$, and N - total number of moving loads. The action of the j^{th} moving load is effective by the term $H(t-t_j)$ once entering the beam, and will be disabled by the term $H(t - t_j - L/v)$ while it leaves the beam.

Then Equation (4.19) can be modified to.

$$F(t) = \sum_{j=1}^{N} P \times \left[U_j(t, v, L) + U_j(t - t_c, v, L)\right] \tag{4.21}$$

This is applicable once neglecting the effect of inertia of the moving masses and the train bridge interaction. Again, the forcing function $F_n(t)$ for the n^{th} generalized coordinate can be given by

$$F_n(t) = \frac{2P}{mL} \sum_{j=1}^{N} [f_n(t, v, L) + f_n(t - t_c, v, L)] \tag{4.22}$$

If only the effects of moving loads are considered, the equation of motion in terms of generalized coordinate q_n

$$\ddot{q_n} + 2\xi_n \omega_n \dot{q_n} + {\omega_n}^2 q_n = F_n(t) \tag{4.23}$$

$$q_n(t) = \frac{1}{m\omega_{dn}} \int_0^t F_n(\tau) e^{-\xi_n \omega_n (t-\tau)} \times sin\, \omega_{dn}(t - \tau) d\tau \tag{4.24}$$

$$= \frac{2PL^3}{EI\pi^4} [P_n(v, t) + P_n(v, t - t_c)] \tag{4.25}$$

where ω_{dn} is the damped frequency of vibration of the beam given by

$$\omega_{dn} = \omega_n \sqrt{1 - {\xi_n}^2} \tag{4.26}$$

The function $P_n(v, t)$ can be expressed as

$$\mathrm{P_n(v,t)} = \frac{1}{\mathrm{n}^4} \sum_{\mathrm{j=1}}^{\mathrm{N}} \frac{1}{(1 - {\mathrm{S_n}}^2)^2 + 4(\xi_\mathrm{n} \mathrm{S_n})^2} \times \left[\mathrm{A} \times \mathrm{H}(\mathrm{t} - \mathrm{t_j}) + (-1)^{\mathrm{n+1}} \times \mathrm{B} \times \mathrm{H}\left(\mathrm{t} - \mathrm{t_j} - \frac{\mathrm{L}}{\mathrm{v}}\right)\right] \tag{4.27}$$

Here the value of A and B are found as,

$$A = \left(1 - {S_n}^2\right)\sin\Omega_n\left(t - t_j\right) - 2\xi_n S_n \cos\Omega_n\left(t - t_j\right) + e^{-\xi_n\omega_n\left(t-t_j\right)} \times \left[2\xi_n S_n \cos\omega_{dn}\left(t - t_j\right) + \frac{S_n}{\sqrt{1-{\xi_n}^2}}\left(2{\xi_n}^2 + {S_n}^2 - 1\right)\sin\omega_{dn}\left(t - t_j\right)\right] \quad (4.28)$$

$$B = \left(1 - {S_n}^2\right)\sin\Omega_n\left(t - t_j - \frac{L}{v}\right) - 2\xi_n S_n \cos\Omega_n\left(t - t_j - \frac{L}{v}\right) + e^{-\xi_n\omega_n\left(t-t_j-\frac{L}{v}\right)} \times \left[2\xi_n S_n \cos\omega_{dn}\left(t - t_j - \frac{L}{v}\right) + \frac{S_n}{\sqrt{1-{\xi_n}^2}}\left(2{\xi_n}^2 + {S_n}^2 - 1\right)\sin\omega_{dn}\left(t - t_j - \frac{L}{v}\right)\right] \quad (4.29)$$

Where, Ω_n denotes the exciting frequencies of the moving loads, $\Omega n = n\pi v/L$, and the speed parameter, $S_n = n\pi v/(\omega_n L)$.

The terms $P_n(v, t)$ and $P_n(v, t - t_c)$ in Equation (3.20) represent the dynamic responses excited by the two sets of wheel loads, and the second set has a time lag t_c after the front set of loads. Neglecting the effect of damping and considering only the first mode of vibration, the dynamic response of the beam can be derived as

$$u(x,t) = \frac{2pL^3}{EI\pi^4} \times \frac{1}{1-{S_n}^2}\sin\frac{\pi x}{L}\left[P_1(v,t) + P_1(v,t-t_c)\right] \quad (4.30)$$

where the response function P1(v, t) for the first set of wheel loads is given by.

$$P_1(v,t) = \sum_{k=1}^{N} \left\{ \left[\sin \Omega_1 (t - t_j) - S_1 \sin \omega_1 (t - t_j) \right] \times H(t - t_j) + \sin \Omega_1 \left(t - t_j - \frac{L}{v} \right) - S_1 \sin \omega_1 \left(t - t_j - \frac{L}{v} \right) \times H \left(t - t_j - \frac{L}{v} \right) \right\} \quad (4.31)$$

The terms $P_1(v, t)$ and $P1(v, t - t_c)$ denote the contribution of the front and rear wheel loads, respectively. Depending on the bridge/car length ratio L/d, there may be different number of wheel loads or no loads acting on the railroad bridge during the passage of the train. So, The most severe case occurs when the front wheel load of the $(N - 1)^{th}$ car has left the bridge, and the front wheel load of the N^{th} car has entered the bridge, namely, when the rear wheel load of the $(N - 1)^{th}$ car and the front wheel load of the N^{th} car are simultaneously acting on the bridge.

The dynamic response of the beam has been excited to the maximum by the former N − 1 cars that have passed the bridge. For this case, $t_N < t < (t_N + L/v)$ and the Equation (4.32) becomes

$$\begin{aligned} P_1(v,t) = {} & [\sin \Omega_1 (t - t_N) - S_1 \sin \omega_1 (t - t_N)] \times H(t - t_N) \\ & - 2S_1 \cos \frac{\omega_1 L}{2v} \times \sin \omega_1 \left(t - \frac{L}{2v} \right) \\ & + \sin \omega_1 \left(t - \frac{L}{2v} - \frac{t_N}{2} \right) \times \frac{\sin \omega_1 \left(\frac{t_N}{2} - \frac{d}{2v} \right)}{\sin \frac{\omega_1 d}{2v}} \times H \left(t - t_{N-1} - \frac{L}{v} \right) \end{aligned} \quad (4.32)$$

The term containing $H(t - t_N)$ represents the dynamic response of the beam induced by the motion of the N^{th} front wheel load of the train, and the term containing $H(t - t_N - 1 - L/v)$ is the free vibration caused by the former N−1 front wheel loads that have passed the bridge. Since the two response functions $P_1(v, t)$ and $P_1(v, t - t_c)$ are similar in nature, only the function $P_1(v, t)$ for the first set of wheel loads is considered in the above Equation (4.32)

4.2. Devon Railroad Bridge and Dynamic Analysis

4.2.1. Overview of Devon Railroad Bridge

The bridge analyzed is in Milford, Connecticut, and forms part of the Northeast Rail Corridor that connects Washington, D.C., and Boston, Massachusetts. This open-deck steel truss railroad bridge, constructed around 1900, is designed for high-speed trains and stretches 325.2 meters across the Housatonic River. It features a seven-span, double-track through truss structure (Malla et al. 2017).

The study focuses on a 66.15-meter-long span near the bridge's east abutment. This bridge has two tracks spaced 3.86 meters apart, with wooden ties and stringers that are 1.98 meters apart laterally. The selected truss span includes seven panels, where stringers are linked to floor beams set 9.45 meters apart along its length. The bridge utilizes a lateral wind bracing system, with built-up members forming the truss end-bracings, verticals, and top chords, while eye bars of varying counts make up the midspan diagonals and bottom chords. The lower lateral braces are single-angle sections, and the upper ones are channel sections.

Figures 6 to 9 illustrate the detailed elevation and plan views of the truss span under study. The stringers, organized into 14 panels, support floor beams placed 4.725 meters apart longitudinally. These beams are constructed from web plates, flanges, angles, and cover plates, with varying numbers and thicknesses. Rolled W 36×150 I-beams form the stringers, as described by Malla et al. (2017) (Malla et al. 2017).

Figure 6: Photograph of Devon railroad bridge, Milford, Connecticut. (Malla et al., 2017)

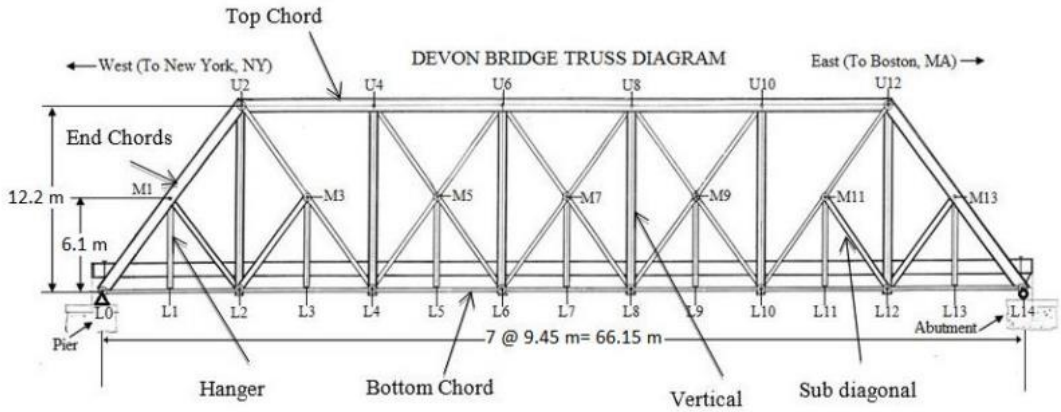


Figure 7: Arrangement of principle members of Devon bridge (Baniya et al., 2015)

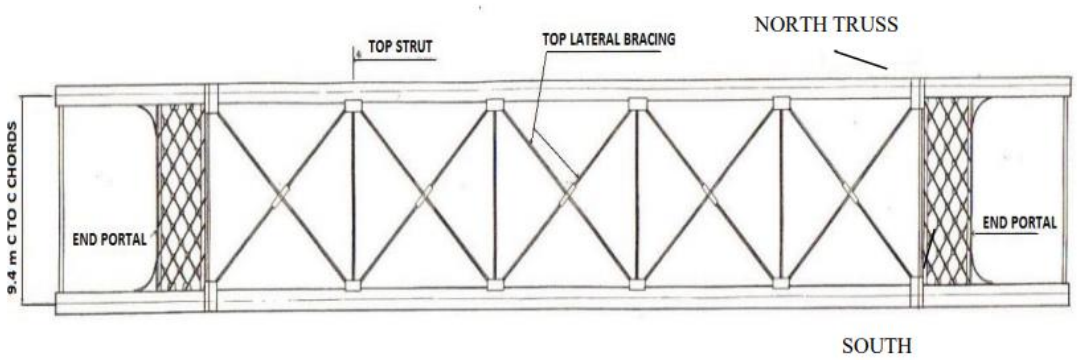


Figure 8: Top Chord plan (Baniya et al., 2015)

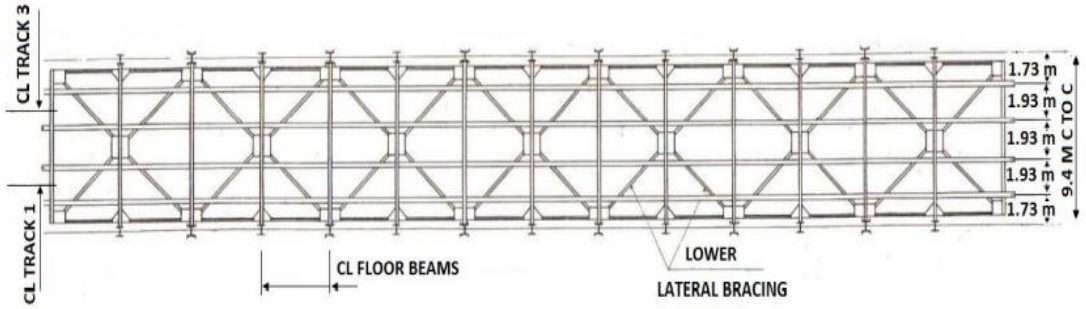


Figure 9: Bottom chord plan (Baniya et al., 2015)

4.2.2. Dynamics of Train Loading and FE Model in SAP2000

In modeling train loads, a methodology akin to vehicle systems was employed, focusing on the contact between the train's wheels and the track and overlooking the impact of mass inertia. The SAP2000 software was used, given its limitation to model only moving constant forces instead of a moving mass system. The chosen model for this study, depicted in Figure 10, is the Amtrak Acela train, which represents a critical load configuration on the infrastructure.

A detailed 3D FE model of the Devon railroad bridge's superstructure was developed within SAP2000, highlighting the direct transfer of loads from the train to the rail, and then to

stringers and cross girders. The model, which is displayed in Figure 11, simplifies the support conditions by defining the west end as a hinge and the east end as a roller. It is assumed that the intersection of all primary members' centerlines at the centroids of their cross-sections preserves the integrity of the bridge's complex steel truss structure without compromising the accuracy of the results. The categorization of bridge components into axial and bending members informs the analysis, as shown in Table 1, and plays a crucial role in understanding the bridge's behavior under train loading.

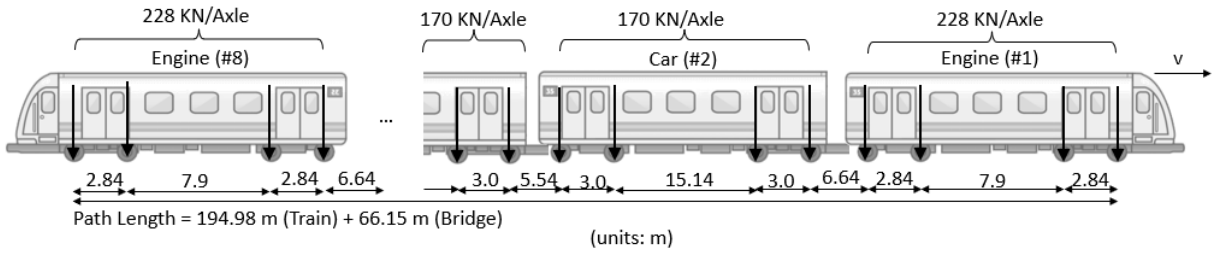


Figure 10: Configuration of loading from Amtrak Acela Train (Pokhrel 2023)

Amtrak Acela train and axle loads have been simulated in SAP2000 for FE modeling.

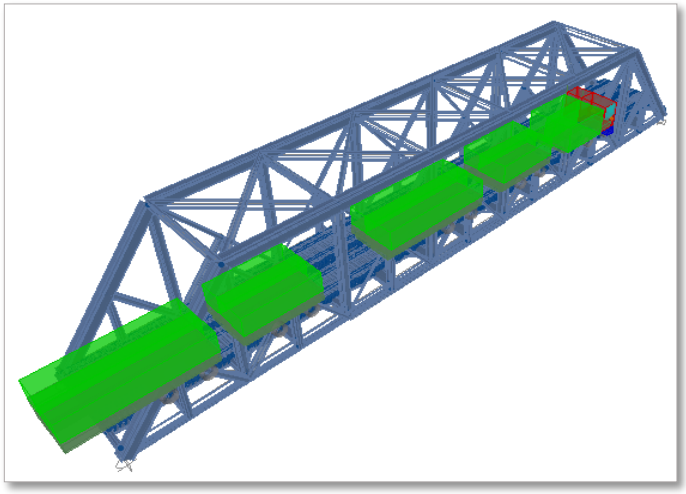


Figure 11: 3D model of Amtrak Acela train and axle loads in SAP2000 (Pokhrel 2023)

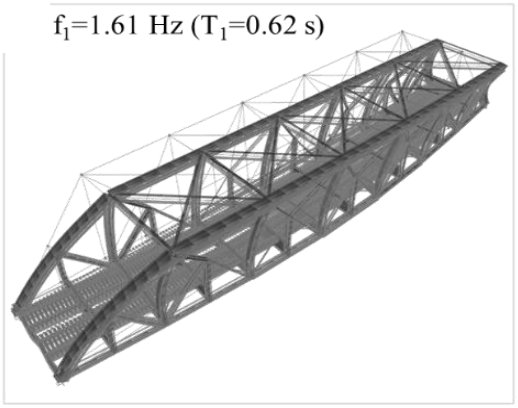
4.2.3. Mode Shapes and Model Validation

The model's validation was grounded in field test data, with measurements from accelerometers and Linear Variable Differential Transformers (LVDT) taken during train passages (Malla et al. 2017). The correlation between the natural frequencies from the FE model and the field data, presented in Table 1, demonstrates the model's validity. Moreover, a comprehensive modal analysis yielded 50 mode shapes and frequencies, substantially aligning with field data, underscoring the model's ability to capture the bridge's dynamic response. The key mode shapes are presented in Figure 12, with the first lateral and vertical modes being particularly significant for understanding the bridge's behavior.

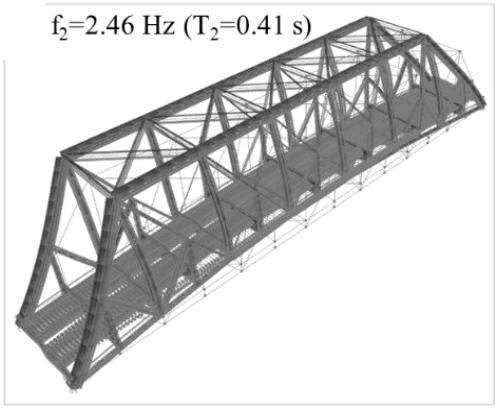
Table 1: Natural frequencies of different modes from FE model and field study (Pokhrel 2023)

Mode Number (Global)	Natural Frequency (Hz)	
	FE model	Field data (Malla et al., 2017)
First lateral mode (Mode 1)	1.6	1.6
Second lateral mode (Mode 2)	2.5	3.3
First vertical mode (Mode 4)	4.4	4.6

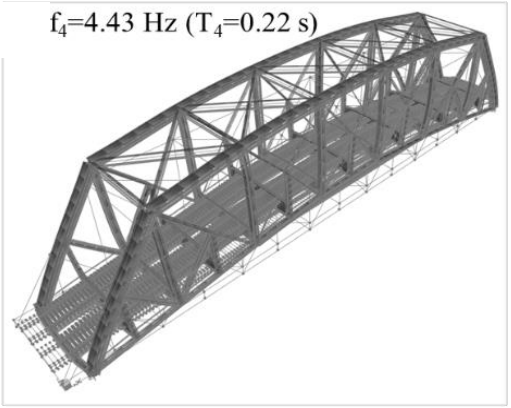
(b)



(a) Mode 1 (First lateral mode)



(b) Mode 2 (Second lateral mode)



(c) Mode 4 (First vertical mode)

Figure 12: Mode shapes of the bridge resulted from the FE model (Pokhrel 2023)

4.2.4. Transient Time History Analysis

The transient time history analysis, integral for assessing dynamic responses, was performed on the FE model to capture the acceleration response at various train speeds. This analysis examines how higher speeds affect the bridge's vibrational frequencies and identifies damage through anomalies in acceleration signals, particularly in the vertical acceleration, which is more sensitive to damage and stiffness loss in the bridge's deck members.

4.3. Comparative Study with Other Finite Element Software

The research starts with creating a model of a SSB based on the numerical validations provided by Yang et al (Yang et al. 2004). To confirm the accuracy of the results and to compile the datasets, the model is checked against various methods and sources. This beam model is then analyzed for deflection and acceleration when a moving point load is applied, using software like SAP2000 and OpenSEESPy. This detailed process helps validate the beam model, ensuring that the data collected for structural analysis is reliable.

4.3.1. Formulation of Moving Load Scenario using OpenSEESPy

This study introduces a computational framework developed in the OpenSEESPy interface, part of the OpenSEES open-source software project, for simulating the dynamic response of SSBs subjected to moving point loads. The methodology involves discretizing the beam into finite elements and setting up nodes at uniform intervals along its length. It then applies boundary conditions that prevent horizontal and vertical movements at the beam's ends but permit rotation. The approach proceeds by applying a point load along the Y-axis to each node, paced according to the timestep intervals. This process is designed to simulate dynamic loading conditions by transferring the applied load from node to node and then removing it in accordance with the timestep intervals, ensuring that static loads do not accumulate. Consequently, larger timesteps may lead to a response resembling that of a static system, whereas smaller timesteps are utilized to ensure a behavior that closely approximates dynamic conditions. It defines elastic beam column elements with attributes such as cross-sectional area, Young's modulus, and moment of inertia, incorporating geometric transformations for realism.

Utilizing the transient analysis method with Newmark integration, the framework employs a BandGeneral system solver, Transformation constraints, and a Modified Newton algorithm for energy increment convergence testing. A key feature of the study is the dynamic application of moving loads, simulating the effect of load traversal over time. This enables a detailed analysis of displacement responses at critical beam points, enhancing our understanding of structural dynamics under such loading conditions. The chosen simulation parameters, including load speed, duration, and timestep, are carefully selected to capture the intricate behavior of beams under dynamic loads, offering valuable insights into structural performance and resilience.

4.3.2. Model Validation for the Single Point Load Case

The study examines a SSB characterized by specific parameters shown in Figure 13: Young's modulus E is set at 2.87 GPa, Poisson's ratio ν at 0.2, the moment of inertia I at 2.90 m^4, mass per unit length m at 2303 kg/m, with a suspended mass M_v of 5750 kg, moving at a speed v of 100 km/h (Yang et al. 2004).

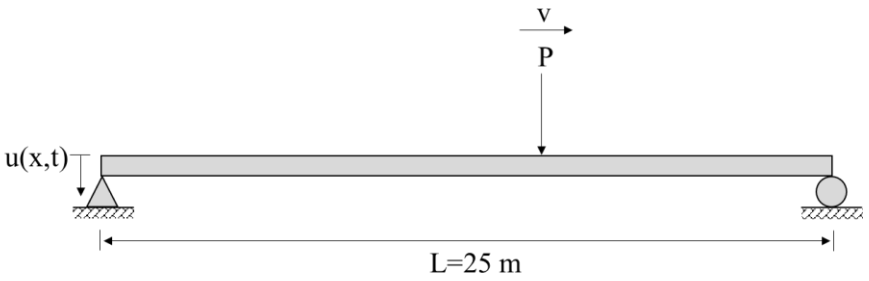


Figure 13: Diagram of a SSB with a single moving point load

To confirm the findings, a numerical simulation is carried out, supplemented by the analytical solution provided by c This is then paralleled with results from SAP2000 and

OpenSEESPy software. The central focus is on calculating the vertical displacement at the beam's mid-span. These calculations are then normalized against the maximum static displacement. The resulting graph delineates the impact of the moving load on both the displacement and acceleration at the mid-span of the beam.

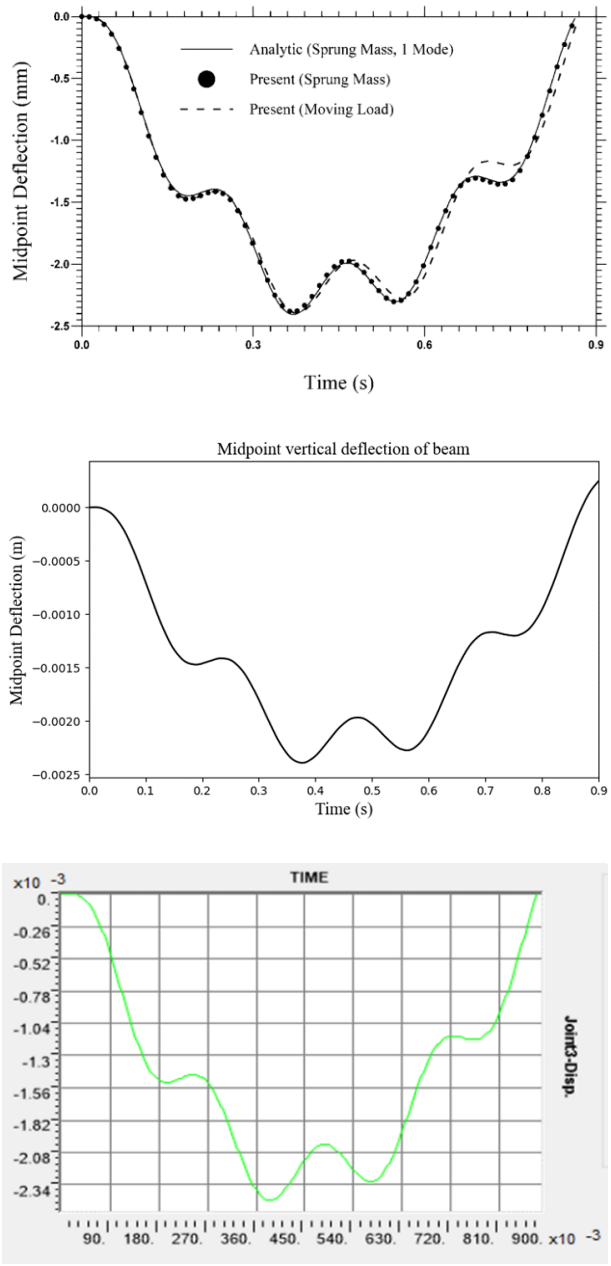


Figure 14: Comparative analysis of beam deflections (a) analytical solution (b) OpenSEESPy (c) SAP2000

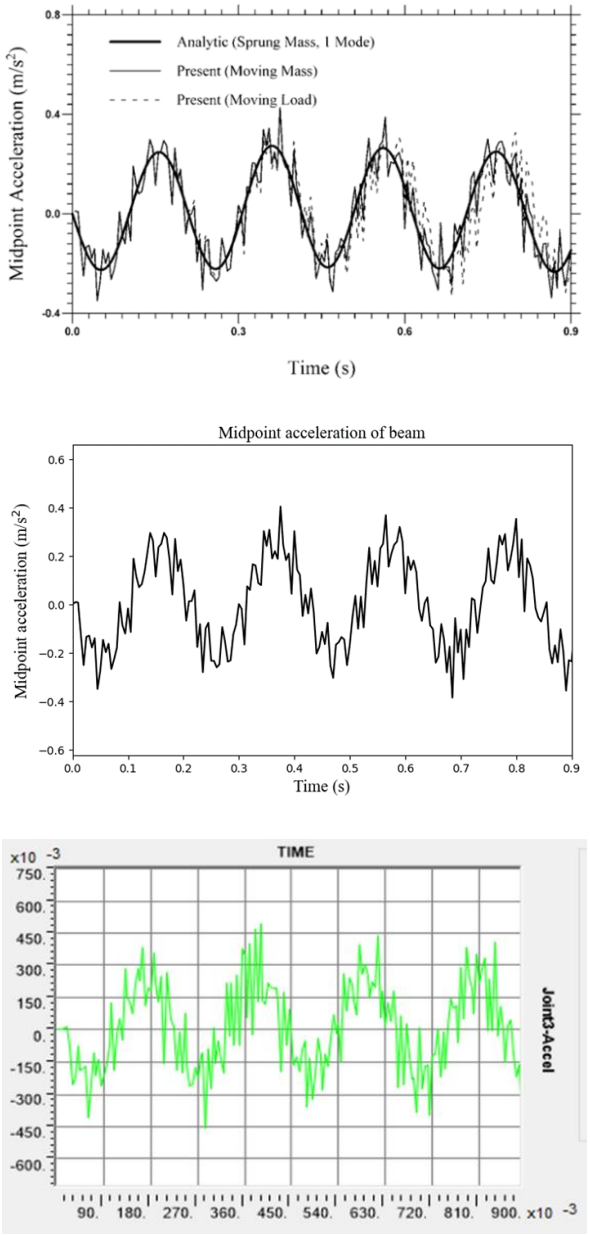


Figure 15: Comparative Analysis of Beam acceleration responses with time step, dt = 0.005 (a) analytical solution (b) OpenSEESPy (c) SAP2000

This analysis underscores the influence of timestep size on the acceleration responses observed in FE simulations. Utilizing smaller timesteps increases data point density, thereby unveiling notable variances across outcomes from various FE software. To more closely mirror analytical solutions and ensure uniformity among FE models, it is advisable to select timesteps ranging from 0.001 to 0.005. The Figure 15 presented was generated using a timestep (dt) of 0.005. Nonetheless, dt=0.005 was selected as it most closely approximated the graph presented in the analytical solution by Yang et al (Yang et al. 2004) as shown in Figure 15 (a). Across all scenarios, a discretization length of 0.01 m was employed, a decision arrived at through trial and error, to enhance the accuracy of the simulation outcomes.

4.3.3. Model Validation for the Double Point Load Case

Adopting a timestep of dt = 0.005 and a discretized beam length of 0.01 m, the study extended the algorithm to model scenarios with a double point load by setting the distance between the two loads at 8m, each with a force of 56,384 KN. Results from both SAP2000 and OpenSEESPy showed matching deflections around 4 mm and similar patterns in acceleration responses. As shown in Figure 16 and 17. These parallel findings for simulations with single and double point loads affirm the moving load algorithm's accuracy, suggesting its suitability for further analytical exploration.

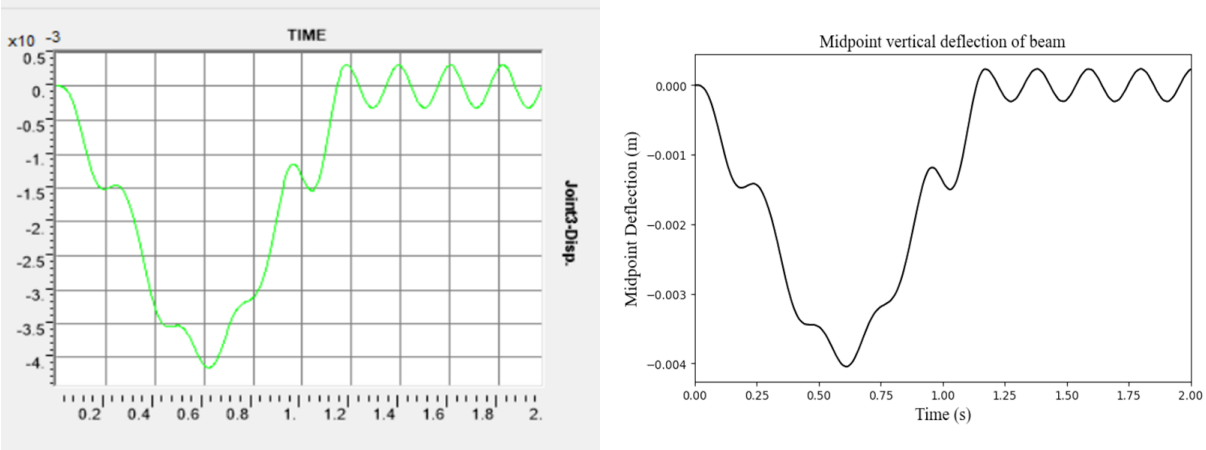


Figure 16: Comparative analysis of beam midpoint deflections with double point load (a) SAP2000 & (b) OpenSEESPy

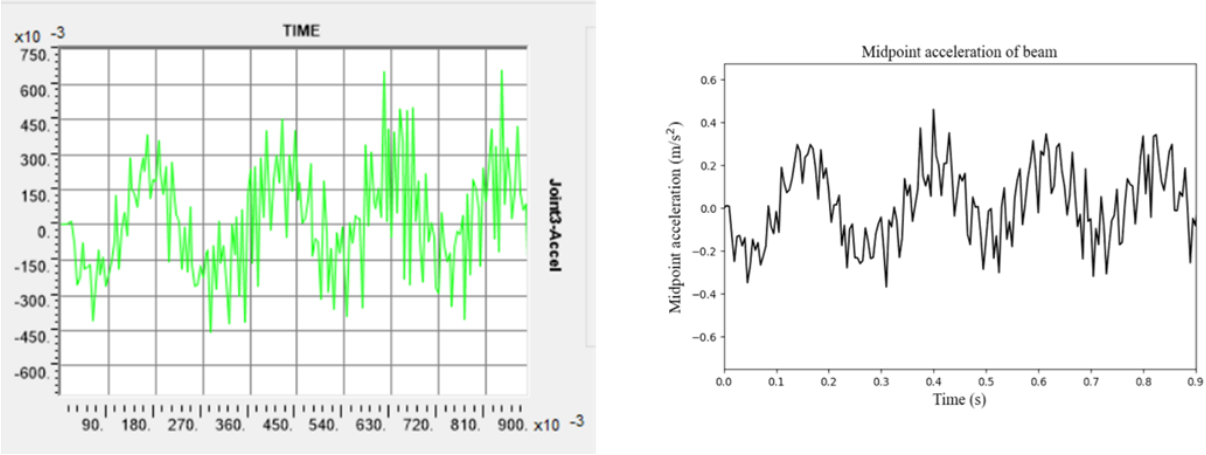


Figure 17: Comparative Analysis of Beam midpoint acceleration with double point load (a) SAP2000 & (b) OpenSEESPy

CHAPTER V

PARAMETRIC STUDY

In this chapter discussing the parametric study, the selection of a validated simply supported beam model served as the initial step, chosen for its simplicity and the robust capabilities of OpenSEESPy to execute iterative parameter loops, enabling the extraction of datasets across a broad spectrum of values. Subsequently, a finite element (FE) bridge model was utilized to gather datasets within a constrained range from the SAP2000 model. The optimal values identified from the LSTM model analysis will then be applied to assess its predictive performance, taking into account the optimal hyperparameters. This structured approach facilitates a comprehensive evaluation of model efficacy across varying structural scenarios.

5.1. LSTM for Simply Supported Beam under a Double Axles Moving Load

5.1.1. Dataset Preparation

The analysis began with a validated SSB, selected for its simplicity and the robustness of OpenSEESPy, which allows for efficient iteration over a range of parameters to gather extensive datasets. Initially, the dataset comprised responses from the simply supported beam under the influence of two-point loads, as detailed in section 4.3.3. Data points were collected from three designated points on the beam L/3, L/2, and 2L/3. For the initial dataset, speed values ranged from 20 m/s to 30 m/s in increments of 0.5 m/s, load values from 50 KN to 100 KN in

increments of 10 KN, and axle distances from 5 m to 10 m in 1m increments. This parameterization yielded a total of 720 data sets. The datasets were compiled into comma-separated values (CSV) files, with all response data both displacements and accelerations sequentially arranged in the same columns.

In this research, the assembly and integration of datasets into a singular comma-separated file was a deliberate process undertaken to explicate the usage of LSTM models for analyzing sequential data. These models are adept at handling time series where data points x_t are documented across a series of time steps $t = t_0 \ldots t_N$, with N symbolizing the sequence's extent. For data encompassing D dimensions, a sequence or "dataset" is characterized by the shape $N \times D$.

When sequences are of uniform length N, with B sequences available, they can be arrayed into a tensor shaped $B \times N \times D$, encapsulating the dataset for model utilization. In this tensor, B signifies the batch axis, marking the dimension for stacking independent sequences. Should the entire dataset be utilized for training simultaneously, it would be introduced into the model as a tensor of size $B \times N \times D$, defining the batch size as B.

If, however, sequences display differing lengths, the training approach must be adapted. One must ascertain whether the entirety of each sequence is crucial for training. If segments of the sequence, $n<N$, are sufficient for prediction, one could select b sequences of this length n, hereby defining a new discretionary batch size b.

In instances where only the full sequence length can facilitate accurate predictions, it is advised to process each sequence independently, effectively designating each sequence as an individual batch (b=1). This requires the model to be provided with tensors of dimension b×n×D, which takes into account the challenge of aligning sequences of diverse lengths.

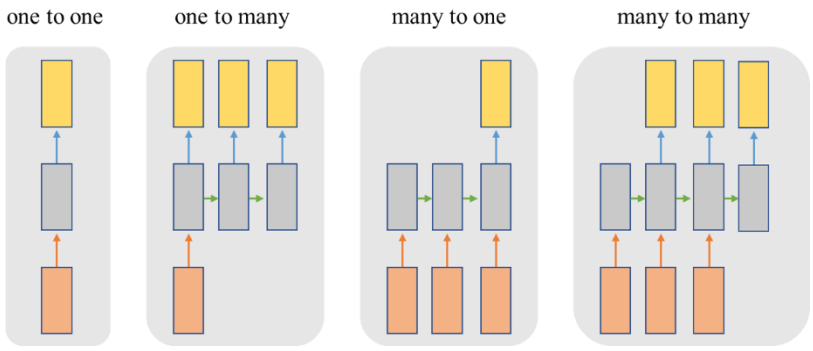


Figure 18: Types of RNN architectures: one-to-one, one-to-many, many-to-one, many-to-many, and many-to-many (Anon n.d.)

Turning to LSTM architectures and specifically addressing the many-to-one and many-to-many configurations, these are particularly relevant for their application in time series analysis. Within the Keras framework, "time steps" correspond to the number of LSTM cells and define one dimension in the shape of the 3D tensor [batch_size, timesteps, input_dim].

Illustrating this concept, a dataset comprising 2 features across 6 time steps serves as a reference. The many-to-one architecture necessitates the data to be reshaped such that predictions at a given time step t rely on a determined number of preceding time step features. The "input_dim" in the 3D tensor mirrors the number of features in the initial 2D dataset; in this

example, "input_dim" is 2. A specific function will be crafted to transform the 2D data into the desired 3D "sliding window" form.

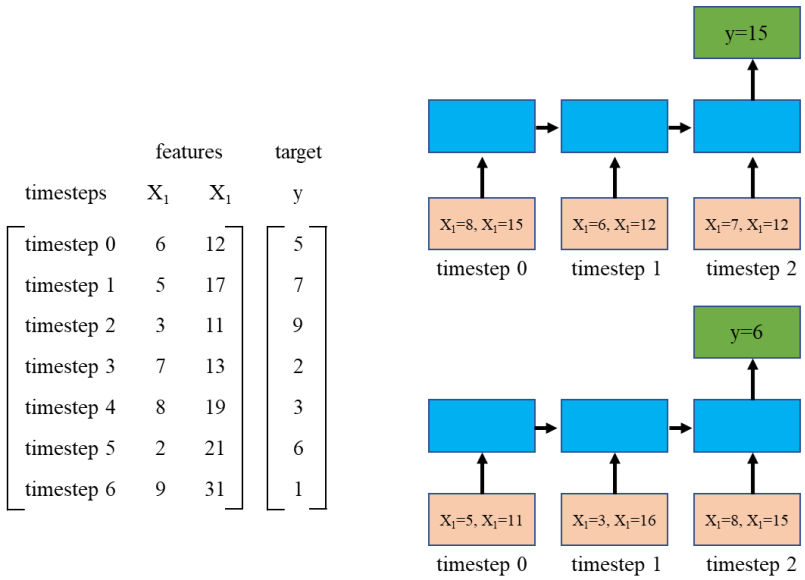


Figure 19: Many-to-one sequence prediction example using RNNs (Bikmukhametov 2020)

Addressing a many-to-many LSTM challenge, the Python code in our study adeptly preprocesses data by reformatting sensor measurements into a configuration suitable for LSTM training. This involves three sensors providing acceleration or displacement data, identified as features X1, X2, and X3, which are then associated with the corresponding targets Speed, Load, and Axle distance, labeled as Y1, Y2, and Y3, respectively. The processed datasets are structured with the batch size B and sequence length N, both discernible from the data's tensor shapes 720 batches each containing 200 sequential readings represented by 3 features. To enhance the model’s ability to perform, a StandardScaler transformation is employed, standardizing both the input features and target outcomes to have a mean value of zero and a standard deviation of one. Such standardization is crucial for the neural network's learning algorithm, as it helps ensure efficient and reliable convergence during training.

5.1.2. Hyperparameters of the LSTM Network

Regularization

Regularization is a technique used to prevent overfitting, a common problem in ML where a model learns the training data so well that it performs poorly on new, unseen data (Merity, Keskar, and Socher 2017). It works by adding a penalty for complexity to the loss function, encouraging the model to keep the weights small, thereby simplifying the model. This is essential because in the context of time series and inverse problems like predicting acceleration responses, we aim for a model that captures the true underlying patterns, not the noise or anomalies in the training set. In LSTMs, regularization can be particularly effective because these models have the potential to be highly complex due to their recurrent nature, which can lead to overfitting if not controlled.

Early Stopping

Early stopping is a regularization technique that mitigates overfitting by ceasing the training of a ML model at the moment when performance on a validation set starts to degrade (Reimers and Gurevych 2017a). The essence of early stopping is to track the error on a validation dataset during the training process and to stop the training when this error begins to increase, indicating that the model is no longer learning generalizable patterns. In the context of LSTMs, which are prone to overfitting due to their ability to capture long-term dependencies in time series data, early stopping is crucial for maintaining the ability to generalize from historical data to future data points.

The provided figure shows the trajectory of training and validation errors over successive epochs. As training progresses, the model's error on the training set typically decreases,

demonstrating that it is learning. However, if the validation error stops declining and starts to ascend, as depicted by the divergence of the two curves in the image, the model is likely beginning to overfit to the training set. Early stopping would be employed at the juncture just before the validation error rises, ensuring that the model retains its generalization capabilities and thus can predict future events with greater accuracy.

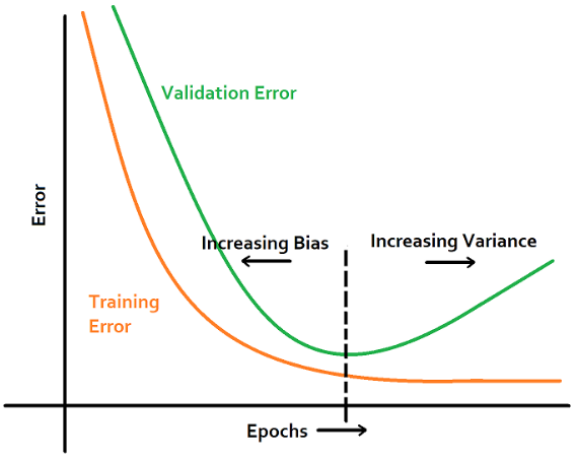


Figure 20: Optimal bias-variance tradeoff in model training illustrated by validation and training error curves (Tyagi, Rane, and Manry 2022)

The provided figure shows the trajectory of training and validation errors over successive epochs. As training progresses, the model's error on the training set typically decreases, demonstrating that it is learning. However, if the validation error stops declining and starts to ascend, as depicted by the divergence of the two curves in the image, the model is likely beginning to overfit to the training set. Early stopping would be employed at the juncture just before the validation error rises, ensuring that the model retains its generalization capabilities and thus can predict future events with greater accuracy.

Units

Units in an LSTM refer to the number of neurons in each cell's hidden layer, which directly affects the model's capacity to capture information. If there are too few units, the model

may not be able to learn complex patterns (underfitting), whereas too many units can lead to overfitting. The choice of units is thus a balancing act: sufficient to capture the intricacies of the data, such as the various factors affecting acceleration in a time series, but not so many that the model becomes excessively complex and tailored to the training set (Reimers and Gurevych 2017b).

Batch Size

Batch Size impacts the learning process by determining how many samples the network is exposed to before the weights are updated. A smaller batch size provides a more noisy gradient, which can be beneficial for finding global minima but may also make the training process less stable. Conversely, a larger batch size provides a more accurate estimate of the gradient, but can also lead to local minima and requires more memory. In the context of LSTMs used for time series analysis, the right batch size can significantly influence the speed and quality of the learning, balancing the need for efficient computation with the model's ability to generalize from the training data.

Learning Rate

Learning Rate is a critical hyperparameter in the training of neural networks, controlling the size of the weight updates during training. An optimal learning rate is one that is large enough to enable fast convergence to a good solution but small enough to prevent the weights from diverging. With LSTMs that are dealing with time series data, such as predicting acceleration responses, the right learning rate is especially crucial, as it needs to be set so that the network can navigate through complex temporal dynamics without overshooting important subtleties in the data.

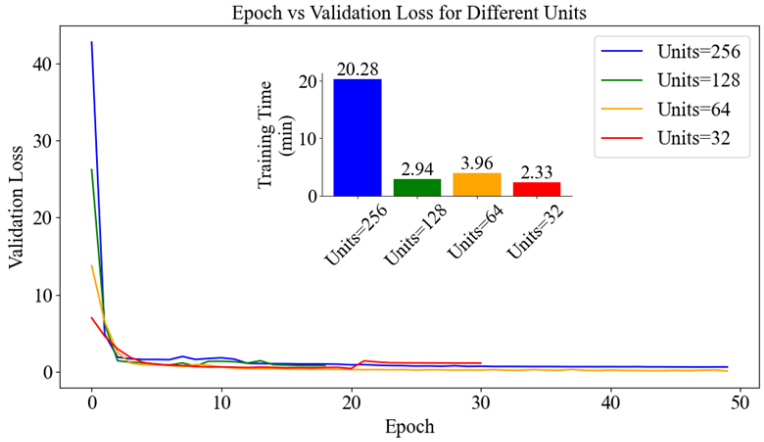
5.1.3. Hyperparameters Optimization

Influence of Units, Batch Sizes, and Learning Rate

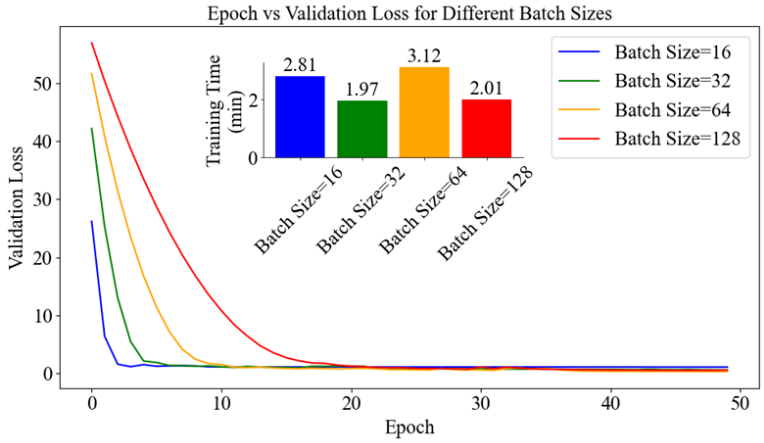
Figure 21 provides a succinct visualization of the effects of hyperparameter tuning on LSTM training efficacy, with kernel regularization applied. The top graph indicates that increasing the number of units decreases the validation loss but incurs longer training times, with diminishing returns on loss reduction beyond 128 units. The middle graph reveals that smaller batch sizes lead to faster convergence but longer training times, with batch size 16 striking an optimal balance between validation loss reduction and training efficiency. The bottom graph illustrates that a learning rate of 0.001 offers the most favorable balance, achieving low validation loss without the training time penalties observed at lower rates or the convergence issues present at higher rates. This composite analysis underscores the importance of careful hyperparameter selection to optimize LSTM performance, ensuring robustness and efficiency in predictive modeling of time series data.

Influence of Regularization and Early Stopping

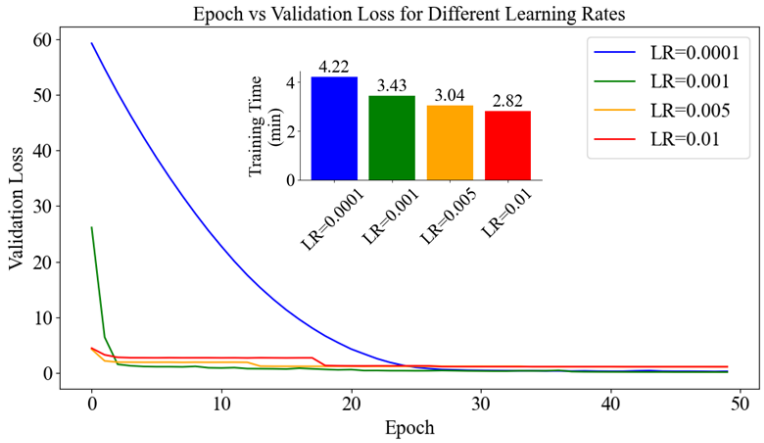
In Figure 22, the LSTM model's performance is analyzed in the absence of kernel regularization, employing only early stopping to prevent overfitting. This strategy's effectiveness is evident across varying hyperparameters, as seen in the convergence of validation loss over epochs. A meticulous assessment reveals that a batch size of 128, despite its longer training time, yields a stable and satisfactory reduction in validation loss. Similarly, a model with 16 units is optimal, minimizing validation loss effectively while keeping training time manageable. The learning rate of 0.001 proves advantageous, striking a balance between the extremes of underfitting and volatile convergence. The superior performance of these settings validates their selection for more in-depth exploration and fine-tuning in future work, emphasizing their contribution to achieving an efficient yet robust LSTM model for time series forecasting tasks.



(a)



(b)



(c)

Figure 21: Impact of hyperparameters on model performance: units, batch sizes, and learning rates with kernel regularization

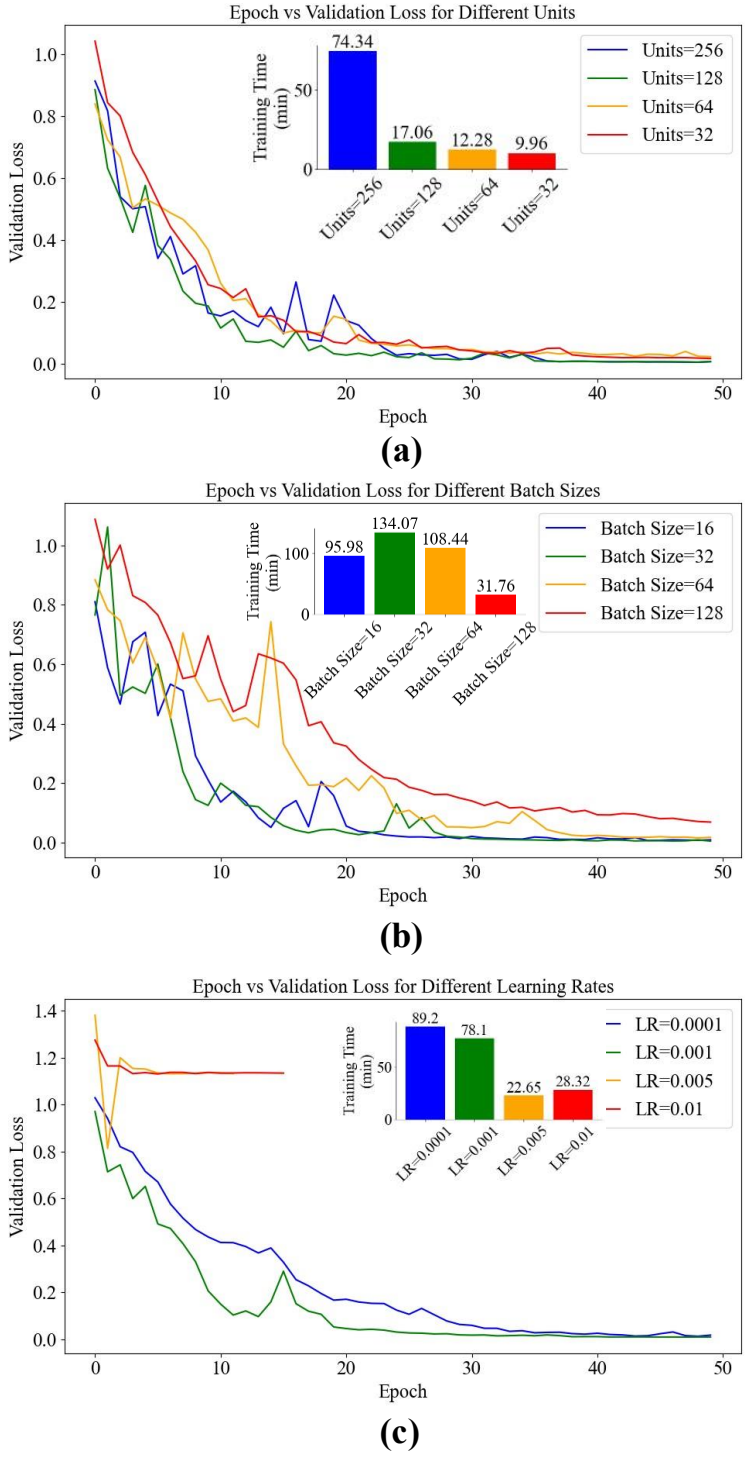


Figure 22: Model training dynamics: effects of unit count, batch size, and learning rate on validation loss without kernel regularization

5.1.4. Influence of the SSB's Parameters

Influence of the Number of Acceleration Sensors

Figure 23, shows the impact of sensor count on the validation loss of an LSTM model throughout 50 epochs. The dataset collected from a single sensor concludes with a loss of 0.0411, suggesting a less accurate model. With the addition of a second sensor, the loss is notably reduced to 0.0042, indicating a substantial improvement in model performance. The introduction of a third sensor further diminishes the validation loss to 0.0038, showcasing the lowest error and the best model performance among the configurations tested. This trend evidences the positive correlation between the number of sensors and the predictive accuracy of the model, with a higher quantity of sensors providing more comprehensive data for the model to learn from, thus optimizing the LSTM's ability to generalize and reducing the validation loss.

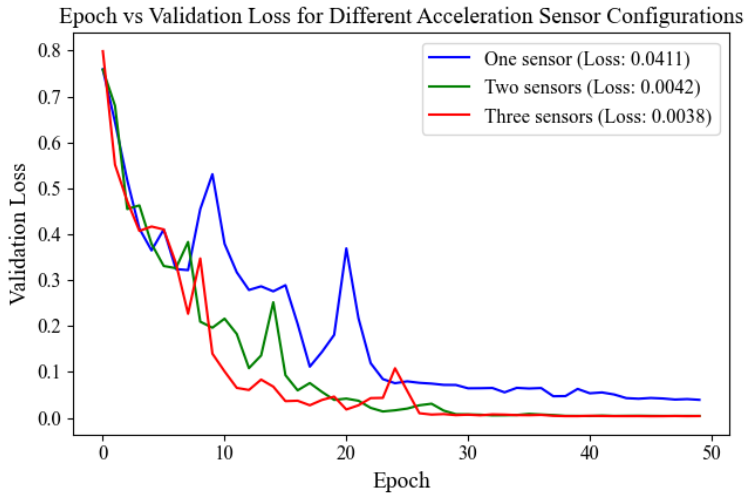


Figure 23: Validation loss comparison across different acceleration sensor configurations during training

Influence of Displacement vs Acceleration Sensors

Figure 24, displays the LSTM model's validation loss when predicting train car features such as speed, load, and the number of axles on railroad bridges, using different numbers of displacement sensors. A single sensor achieves a loss of 0.0058, while adding a second sensor reduces the loss to 0.0010, and employing three sensors further lowers it to 0.0016, indicating enhanced model performance with additional sensor data. However, despite the improved results with displacement sensors, the research proceeded with acceleration sensors. This decision was informed by practical considerations: installing WIM systems for displacement measurement on railroad bridges is often unfeasible due to structural and economic constraints. In contrast, acceleration sensors are significantly more viable to install and maintain. Consequently, while displacement sensors might provide better predictive accuracy for certain train features, the ease and feasibility of acceleration sensors offer a more practical solution for continuous monitoring and feature prediction on railroad bridges.

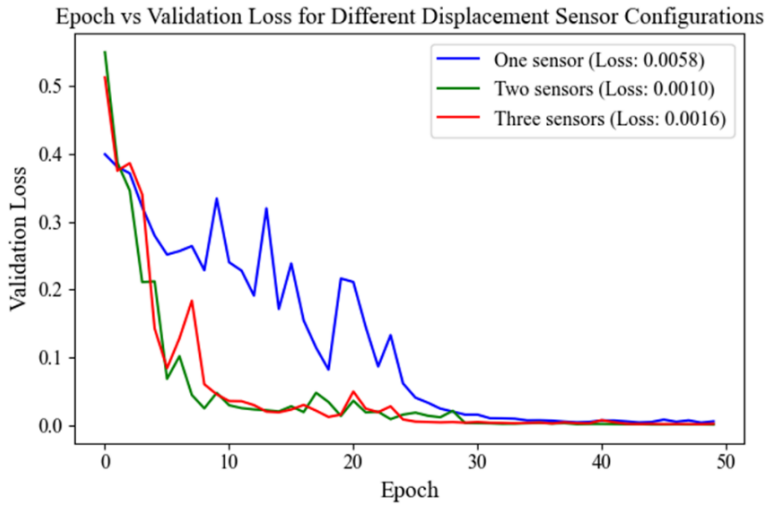


Figure 24: Validation loss comparison across different displacement sensor configurations during training for simply supported beam

Influence of Noise in the Acceleration Sensors

Figure 25 showcases the LSTM model's performance with three acceleration sensors, illustrating the model's ability to withstand various levels of noise a key consideration in real-world scenarios where data from railroad bridge sensors are often muddled with environmental and operational interference. As noise levels increase from 0% to 40%, the validation loss predictably escalates, reflecting the model's rising challenge to discern signal from noise. At no noise, the model achieves an impressively low loss of 0.009, indicating high precision in predicting train features such as speed, load, and axle number. With 10% noise, there is only a slight uptick in loss to 0.032, demonstrating robustness. However, beyond this point, the model experiences greater difficulty maintaining accuracy, with loss reaching 0.062 at 20% noise and progressively worsening to 0.113 at 40% noise. These increments illustrate the real-world practicality of the model, as it simulates conditions where sensors might encounter vibrations, temperature variations, or electrical noise, affirming the selected three-sensor array's effectiveness and the necessity for models capable of operating amidst such stochastic disturbances.

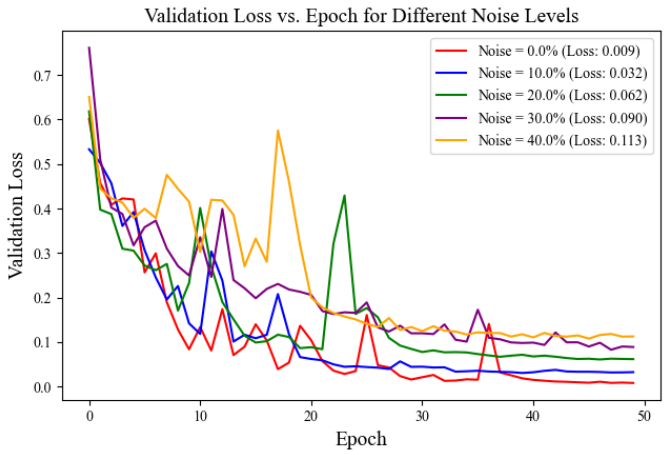


Figure 25: Effect of data noise levels on model validation loss over training epochs

5.1.5. Performance Evaluation of the LSTM Network

Finally, in this section, we evaluate the accuracy of the optimized LSTM network by calculating the following performance indices (Chicco, Warrens, and Jurman 2021):

The Median Absolute Error (MedAE) is a statistical measure used to evaluate the accuracy of a prediction model or estimator. It is particularly robust to outliers, as it uses the median rather than the mean. The MedAE is calculated by taking the median of all absolute errors between the predicted values and the actual values. Absolute error is simply the absolute difference between predicted and actual values. MedAE is calculated using the formula:

$$MedAE = median(|y_i - \hat{y}_i|) \tag{5.1}$$

Here, y_i represents the actual values, $\hat{y}_i$ represents the predicted values and $|y_i - \hat{y}_i|$ is the absolute error between the predicted and actual value for each observation

Mean Absolute Error (MAE) calculates the average of the absolute differences between the predicted values and the actual values. Here's how it's typically calculated

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|) \tag{5.2}$$

Mean Squared Error (MSE) is a common metric used to measure the accuracy of predictions in statistics and machine learning. It calculates the average of the squared differences between predicted values and actual values. Because errors are squared, it tends to give more weight to larger errors, making it sensitive to outliers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.3}$$

Mean Absolute Percentage Error (MAPE) is another widely used metric for evaluating the accuracy of predictions, especially in the fields of forecasting and regression analysis. MAPE measures the size of the error in percentage terms, providing an easy-to-interpret scale of how large the prediction errors are relative to the actual values.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{\hat{y}_i}| \tag{5.4}$$

R-squared (R^2), or the coefficient of determination, is a statistical measure used to evaluate the goodness-of-fit of a regression model. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. R-squared values range from 0 to 1, where higher values suggest a better fit between the model and the data.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{5.5}$$

Here, SS_{res} represents the residual sum of squares (the sum of squared errors between predicted and actual values) and SS_{tot} represents the total sum of squares (the sum of squared differences between actual values and their mean).

Table 2: Performance indices for the optimized LSTM network

Performance Index	Value
MedAE:	0.12594
MAE:	0.009
MSE:	0.018
MAPE:	0.9382
R-squared (R^2):	0.99525

The MedAE (Median Absolute Error) of 0.12594 indicates the typical magnitude of errors, while the MAE (Mean Absolute Error) of 0.009 suggests a high level of accuracy with

small prediction errors. The MSE (Mean Squared Error) of 0.018 penalizes larger errors more heavily, and the MAPE (Mean Absolute Percentage Error) of 0.9382 indicates that, on average, predictions are off by about 94%. The high R-squared value of 0.99525 suggests that the model explains approximately 99.5% of the variability in the data, indicating an excellent fit overall.

5.1.6. Comparative Analysis of LSTM and RNN

To compare the performance of the LSTM model, the same analysis was done using the RNN, and it has been found that the RNN performed significantly poorer, with a validation loss of 0.6268, compared to the LSTM's markedly lower validation loss of 0.0038 as illustrated in Figure 23 and Figure 26. This discrepancy in performance can be logically attributed to the inherent architectural advantages of LSTMs over traditional RNNs. Specifically, LSTMs are designed to overcome the vanishing gradient problem that RNNs are prone to, enabling them to learn from long-range dependencies within the data more effectively. Moreover, the RNN's tendency to overfit to the noise within the training set, as suggested by the fluctuations in training loss, could result in a failure to generalize to new, unseen data, as reflected in its higher validation loss. Lastly, the hyperparameter settings for the RNN may not have been optimal, or its simpler structure could have lacked the necessary capacity to model the complexity of the task at hand, further contributing to its inferior performance when compared to the LSTM model.

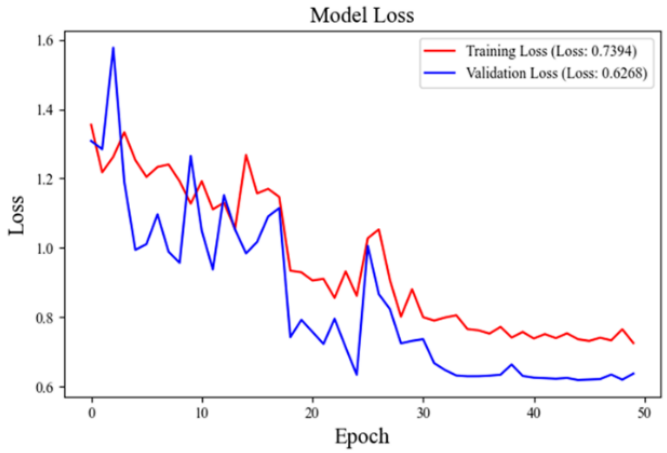


Figure 26: Epoch vs validation loss for RNN using three sensors for simply supported beam

5.2. LSTM for Devon Bridge under a Double Axles Moving Load

Upon determining the optimal hyperparameters for the simply supported beam, our investigation extended to the empirical validation of these parameters for the prediction of velocity and load characteristics pertaining to the acceleration sensors deployed in the Devon bridge model. This validation process entailed the meticulous adjustment of bridge parameters, including the configuration of acceleration and displacement sensors, as well as the mitigation of noise in the acceleration sensors. The ensuing outcomes are delineated herein

5.2.1. Influence of the Devon Bridge's Parameters

Influence of the Number of Acceleration Sensors

Figure 27 illustrates the validation loss comparison across different acceleration sensor configurations during the training of a Devon Bridge with a double axle load in SAP2000, over the course of 50 epochs. The training with data from one sensor results in a validation loss of 0.2004, indicating a less accurate model. The inclusion of a second sensor substantially reduces

the validation loss to 0.0462, signifying an improvement in model accuracy. Upon adding a third sensor, the validation loss decreases further to 0.0341, the lowest of the three configurations, thereby demonstrating the best model performance. This pattern suggests that the number of sensors is positively correlated with the model's predictive accuracy; more sensors provide a richer dataset, which enhances the model's learning, improves its ability to generalize, and minimizes validation loss.

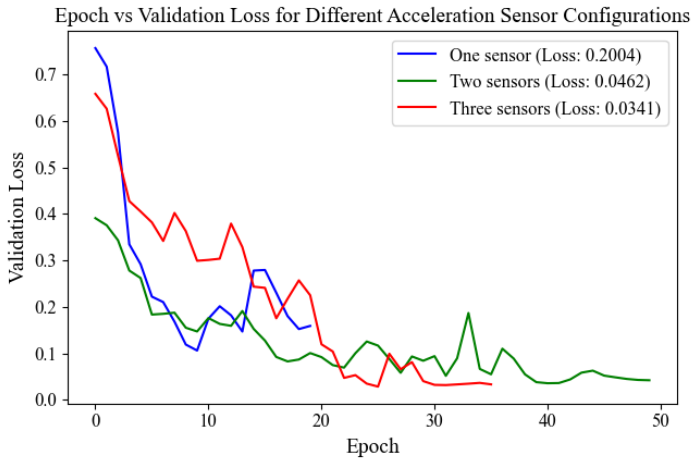


Figure 27: Validation loss comparison across different displacement sensor configurations during training for Devon bridge

Influence of the Number of Displacement Sensors

Figure 28 presents the "Epoch vs Validation Loss for Different Displacement Sensor Configurations" during the training process. This graph details how the validation loss changes over 30 epochs for three separate sensor setups. The model trained with data from one sensor ends with a high validation loss of 0.6149, suggesting it's the least precise. Adding a second sensor reduces the validation loss to 0.1765, showing notable improvement. Introducing a third sensor further lowers the loss to 0.0066, marking the greatest accuracy and demonstrating the most effective model performance out of the configurations tested. This information highlights a

clear correlation between the increase in the number of sensors and the enhancement in the model's predictive capability, affirming the idea that a greater number of sensors can provide more comprehensive data, which is instrumental in refining the model's learning process and in turn, reducing the validation loss.

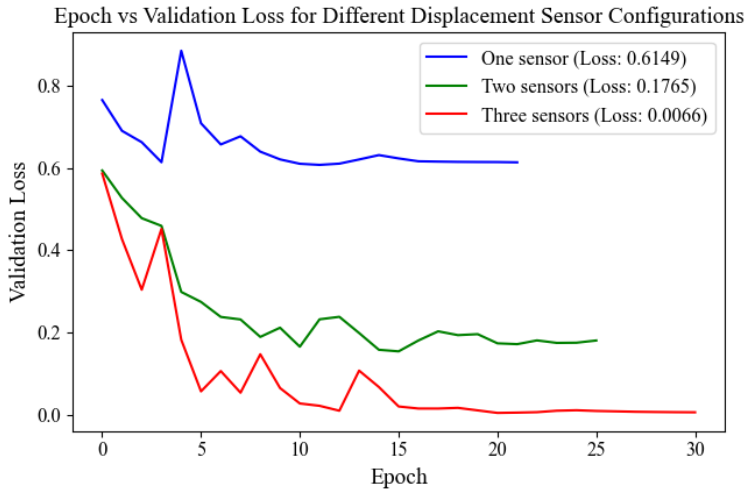


Figure 28: Epoch vs validation loss for different displacement sensor configurations during training for Devon bridge

Influence of Noise in Acceleration Sensors

Figure 29 provides a rigorous evaluation of model performance degradation due to varying noise levels. The validation loss at zero noise is minimal, indicating high model accuracy. As noise increments are introduced, there is a noticeable and consistent increase in validation loss, which underscores a direct and detrimental impact on model precision. The slight decrease in loss at the highest noise level suggests a non-linear response of the model to noise, possibly indicating areas where the model may be overfitting to noise-free data or underfitting when noise is present. This could hypothesize that the model may have a threshold of noise level up to which it can maintain learning efficacy before the accuracy begins to decline. Such a pattern necessitates further investigation into the model's resilience to noise and could inform the development of more robust noise-handling mechanisms in ML.

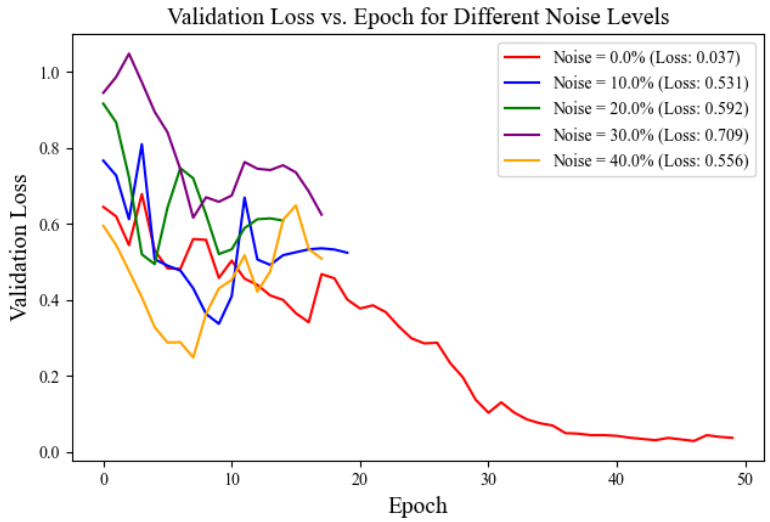


Figure 29: Effect of data noise levels on model validation loss over training epochs for Devon bridge

CHAPTER VI

CONCLUSIONS

6.1. Summary and Conclusions

This comprehensive thesis undertakes a detailed investigation into the deployment of cost-effective sensors for predicting train load and speed by harnessing an extensive dataset from structural health monitoring sensors on railroad bridges. The study utilizes LSTM networks to analyze time series data, meticulously optimizing hyperparameters such as the number of LSTM units, batch size, and learning rate, with the goal of enhancing predictive accuracy.

The methodology is thorough, involving the extraction and cleaning of data from simulated beam models in both OpenSEESPy and SAP2000. This data forms the foundation upon which the LSTM models are trained, with the aim of ascertaining the most effective sensor configurations and hyperparameter settings that result in the lowest validation loss and the most robust model performance. The following are the significant findings from the thesis:

1. The research makes significant strides in assessing the impact of sensor quantity on prediction accuracy, discovering a direct correlation between an increased number of sensors and improved model accuracy. This is evidenced by a remarkable decrease in validation loss from 0.6149 with one sensor to a mere 0.0066 with three sensors for a simply supported beam, and a reduction to 0.0341

for a Devon Bridge model, underlining the superior predictive performance achievable with enhanced sensor data.

2. The study examines how well the model performs in the presence of noise. It found that as the noise levels increased, the error in the model's predictions generally rose. However, at very high noise levels, the error unexpectedly decreased slightly, suggesting there might be a threshold where the model begins to handle noise better. The model's accuracy is confirmed by several key metrics: it has a MedAE of 0.12594, a MAE of 0.009, a MSE of 0.018, and an R-squared value of 0.99525. These numbers show that the model is highly reliable and accurate in its predictions.
3. The conclusive findings of this work assert that strategic hyperparameter optimization, combined with the deliberate scaling of sensor numbers, significantly elevates the precision of predictive models in the context of structural health monitoring. Additionally, the insights into the model's noise resistance are particularly valuable for real-world applications, where data disturbances are commonplace. The results advocate for the use of LSTM networks in the proactive maintenance and safety management of vital railway infrastructures, emphasizing the imperative for continued research to bolster the models' defenses against diverse noise interferences, thus ensuring their operational reliability in varying conditions.

6.2. Future Recommendations

Future work will enhance the scope and accuracy of the structural health monitoring models by addressing several key areas:

1. **Multi-railcar Dynamics:** The models will be extended to simulate the complex dynamics of multiple railcars, which is crucial for analyzing the varying load distributions that more accurately reflect real-world train compositions. This enhancement will allow for the differentiation between train types based on their unique signatures in the data.

2. **Damage Modeling:** Training the models to recognize the altered acceleration response due to damage in structural components, such as bridge beams or rails, is essential. This involves developing algorithms capable of detecting and quantifying structural damage from the patterns in sensor data, significantly improving early warning systems.

3. **Environmental Variability:** Incorporating environmental factors such as wind loads, temperature fluctuations, and seismic events into the model will create a more robust system. Adjusting for these variables ensures that the monitoring system remains accurate under a variety of environmental stressors, reflecting the complexity of real-world conditions.

REFERENCES

Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. 2021. "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions." *Journal of Big Data* 8:1–74.

Anon. n.d. "Packt+ | Advance Your Knowledge in Tech." *Packt*. Retrieved April 30, 2024 (https://www.packtpub.com/en-at/product/neural-networks-with-keras-cookbook-11-9781789346640/chapter/building-a-recurrent-neural-network/introduction).

Barke, Derek, and Wing Kong Chiu. 2005. "Structural Health Monitoring in the Railway Industry: A Review." *Structural Health Monitoring* 4(1):81–93.

Beghini, Marco, Tommaso Grossi, Michael B. Prime, and Ciro Santus. 2023. "Ill-Posedness and the Bias-Variance Tradeoff in Residual Stress Measurement Inverse Solutions." *Experimental Mechanics* 63(3):495–516.

Bikmukhametov, Timur. 2020. "How to Reshape Data and Do Regression for Time Series Using LSTM." *Medium*. Retrieved April 30, 2024 (https://towardsdatascience.com/how-to-reshape-data-and-do-regression-for-time-series-using-lstm-133dad96cd00).

Cao, X., Y. Sugiyama, and Y. Mitsui. 1998. "Application of Artificial Neural Networks to Load Identification." *Computers & Structures* 69(1):63–78.

Chan, Tommy HT, S. S. Law, T. H. Yung, and X. R. Yuan. 1999. "An Interpretive Method for Moving Force Identification." *Journal of Sound and Vibration* 219(3):503–24.

CHE, Lin, Jie Liu, and Chao Jiang. 2013. "# Br# Radial Basis Shape Function Method for Identification of Dynamic Load in Time Domain." *China Mechanical Engineering* 24(3):285.

Chicco, Davide, Matthijs J. Warrens, and Giuseppe Jurman. 2021. "The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation." *Peerj Computer Science* 7:e623.

Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. 2000. "Learning to Forget: Continual Prediction with LSTM." *Neural Computation* 12(10):2451–71.

Graves, Alex, and Jürgen Schmidhuber. 2005a. “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures.” *Neural Networks* 18(5–6):602–10.

Graves, Alex, and Jürgen Schmidhuber. 2005b. “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures.” *Neural Networks* 18(5–6):602–10.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9(8):1735–80.

Hodge, Victoria J., Simon O’Keefe, Michael Weeks, and Anthony Moulds. 2014. “Wireless Sensor Networks for Condition Monitoring in the Railway Industry: A Survey.” *IEEE Transactions on Intelligent Transportation Systems* 16(3):1088–1106.

Hwang, J. S., T. J. Lee, H. J. Park, and S. H. Park. 2016. “Dynamic Force Identification of a Structure Using State Variable in the Frequency Domain.” *J. Wind. Eng* 20:195–202.

Jacquelin, E., A. Bennani, and P. Hamelin. 2003. “Force Reconstruction: Analysis and Regularization of a Deconvolution Problem.” *Journal of Sound and Vibration* 265(1):81–107.

Jang, T. S. 2013. “A Method for Simultaneous Identification of the Full Nonlinear Damping and the Phase Shift and Amplitude of the External Harmonic Excitation in a Forced Nonlinear Oscillator.” *Computers & Structures* 120:77–85.

Jang, T. S., Hyoungsu Baek, Hang S. Choi, and Sun-Gu Lee. 2011. “A New Method for Measuring Nonharmonic Periodic Excitation Forces in Nonlinear Damped Systems.” *Mechanical Systems and Signal Processing* 25(6):2219–28.

Jiang, Han, Yajie Zou, Shen Zhang, Jinjun Tang, and Yinhai Wang. 2016. “Short-Term Speed Prediction Using Remote Microwave Sensor Data: Machine Learning versus Statistical Model.” *Mathematical Problems in Engineering* 2016.

Jiang, Jinhui, Mohammed Seaid, M. Shadi Mohamed, and Hongqiu Li. 2020. “Inverse Algorithm for Real-Time Road Roughness Estimation for Autonomous Vehicles.” *Archive of Applied Mechanics* 90:1333–48.

Jiang, R. J., F. T. K. Au, and Y. K. Cheung. 2003a. “Identification of Masses Moving on Multi-Span Beams Based on a Genetic Algorithm.” *Computers & Structures* 81(22–23):2137–48.

Jiang, R. J., F. T. K. Au, and Y. K. Cheung. 2003b. “Identification of Masses Moving on Multi-Span Beams Based on a Genetic Algorithm.” *Computers & Structures* 81(22–23):2137–48.

Kyprioti, Aikaterini P., and Alexandros A. Taflanidis. 2021. “Kriging Metamodeling for Seismic Response Distribution Estimation.” *Earthquake Engineering & Structural Dynamics* 50(13):3550–76.

Law, S. S., Tommy HT Chan, and Q. H. Zeng. 1999. “Moving Force Identification—a Frequency and Time Domains Analysis.”

Li, Cong, Yaonan Zhang, and Guohui Zhao. 2019. “Deep Learning with Long Short-Term Memory Networks for Air Temperature Predictions.” Pp. 243–49 in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. IEEE.

Li, Hongqiu, Jinhui Jiang, and M. Shadi Mohamed. 2021. “Online Dynamic Load Identification Based on Extended Kalman Filter for Structures with Varying Parameters.” *Symmetry* 13(8):1372.

Lipton, Zachary C., John Berkowitz, and Charles Elkan. 2015. “A Critical Review of Recurrent Neural Networks for Sequence Learning.” *arXiv Preprint arXiv:1506.00019*.

Liu, Jie, Xianghua Meng, Chao Jiang, Xu Han, and Dequan Zhang. 2016. “Time-domain Galerkin Method for Dynamic Load Identification.” *International Journal for Numerical Methods in Engineering* 105(8):620–40.

Liu, Jun, Amir Shahroudy, Dong Xu, and Gang Wang. 2016. “Spatio-Temporal Lstm with Trust Gates for 3d Human Action Recognition.” Pp. 816–33 in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer.

Liu, Yang. 2020. “Multi-Scale Spatio-Temporal Feature Extraction and Depth Estimation from Sequences by Ordinal Classification.” *Sensors* 20(7):1979.

Liu, Yaru, Lei Wang, and Kaixuan Gu. 2021. “A Support Vector Regression (SVR)-Based Method for Dynamic Load Identification Using Heterogeneous Responses under Interval Uncertainties.” *Applied Soft Computing* 110:107599.

Malla, Ramesh B., David Jacobs, Suvash Dhakal, and Surendra Baniya. 2017. *Dynamic Impact Factors on Existing Long Span Truss Railroad Bridges*.

Mariani, Stefano, Thompson Nguyen, Xuan Zhu, and Francesco Lanza di Scalea. 2017. “Field Test Performance of Noncontact Ultrasonic Rail Inspection System.” *Journal of Transportation Engineering, Part A: Systems* 143(5):04017007.

Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. 2017. “Regularizing and Optimizing LSTM Language Models.” *arXiv Preprint arXiv:1708.02182*.

Mousavi, Mohsen, Damien Holloway, and J. C. Olivier. 2019. “Using a Moving Load to Simultaneously Detect Location and Severity of Damage in a Simply Supported Beam.” *Journal of Vibration and Control* 25(15):2108–23.

Ni, Hao, Jiangyan Yi, Zhengqi Wen, and Jianhua Tao. 2016. “Recurrent Neural Network Based Language Model Adaptation for Accent Mandarin Speech.” Pp. 607–17 in *Pattern Recognition: 7th Chinese Conference, CCPR 2016, Chengdu, China, November 5-7, 2016, Proceedings, Part II 7*. Springer.

O’Connor, Colin, and Tommy Hung Tin Chan. 1988. “Dynamic Wheel Loads from Bridge Strains.” *Journal of Structural Engineering* 114(8):1703–23.

Ordóñez, Francisco Javier, and Daniel Roggen. 2016. “Deep Convolutional and Lstm Recurrent Neural Networks for Multimodal Wearable Activity Recognition.” *Sensors* 16(1):115.

Pokhrel, Mahesh. 2023. “A Data-Driven Method for Damage Detection in an Open Deck Steel Truss Railroad Bridge Under the Moving Train Load.”

Reimers, Nils, and Iryna Gurevych. 2017a. “Optimal Hyperparameters for Deep Lstm-Networks for Sequence Labeling Tasks.” *arXiv Preprint arXiv:1707.06799*.

Reimers, Nils, and Iryna Gurevych. 2017b. “Optimal Hyperparameters for Deep Lstm-Networks for Sequence Labeling Tasks.” *arXiv Preprint arXiv:1707.06799*.

Sanatel, Çağatay, and Gülay Öke Günel. 2023. “Long Short Term Memory Based Self Tuning Regulator Design for Nonlinear Systems.” *Neural Processing Letters* 55(3):3045–79.

Su, Lei, Hua-Ping Wan, You Dong, Dan M. Frangopol, and Xian-Zhang Ling. 2021a. “Efficient Uncertainty Quantification of Wharf Structures under Seismic Scenarios Using Gaussian Process Surrogate Model.” *Journal of Earthquake Engineering* 25(1):117–38.

Su, Lei, Hua-Ping Wan, You Dong, Dan M. Frangopol, and Xian-Zhang Ling. 2021b. “Efficient Uncertainty Quantification of Wharf Structures under Seismic Scenarios Using Gaussian Process Surrogate Model.” *Journal of Earthquake Engineering* 25(1):117–38.

Taheri Andani, Masood, Abdullah Mohammed, Ashish Jain, and Mehdi Ahmadian. 2018. “Application of LIDAR Technology for Rail Surface Monitoring and Quality Indexing.” *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 232(5):1398–1406.

Thomas, D. L., J. M. Wilson, and R. R. Wilson. 1973. “Timoshenko Beam Finite Elements.” *Journal of Sound and Vibration* 31(3):315–30.

Tyagi, Kanishka, Chinmay Rane, and Michael Manry. 2022. “Regression Analysis.” Pp. 53–63 in *Artificial intelligence and machine learning for EDGE computing*. Elsevier.

Van Houdt, Greg, Carlos Mosquera, and Gonzalo Nápoles. 2020. “A Review on the Long Short-Term Memory Model.” *Artificial Intelligence Review* 53(8):5929–55.

Wan, Hua-Ping, and Yi-Qing Ni. 2018. “Bayesian Modeling Approach for Forecast of Structural Stress Response Using Structural Health Monitoring Data.” *Journal of Structural Engineering* 144(9):04018130.

Wang, Cheng, Delei Chen, Jianwei Chen, Xiongming Lai, and Ting He. 2021. “Deep Regression Adaptation Networks with Model-Based Transfer Learning for Dynamic Load Identification in the Frequency Domain.” *Engineering Applications of Artificial Intelligence* 102:104244.

Wang, Peng, Guo Lai Yang, and Hui Xiao. 2013. “Dynamic Load Identification Theoretical Summary and the Application on Mining Machinery.” *Applied Mechanics and Materials* 330:811–14.

Wang, Xin, Xiang Liu, and Zheyong Bian. 2022. “A Machine Learning Based Methodology for Broken Rail Prediction on Freight Railroads: A Case Study in the United States.” *Construction and Building Materials* 346:128353.

Wu, Yuting, Mei Yuan, Shaopeng Dong, Li Lin, and Yingqi Liu. 2018a. “Remaining Useful Life Estimation of Engineered Systems Using Vanilla LSTM Neural Networks.” *Neurocomputing* 275:167–79.

Wu, Yuting, Mei Yuan, Shaopeng Dong, Li Lin, and Yingqi Liu. 2018b. “Remaining Useful Life Estimation of Engineered Systems Using Vanilla LSTM Neural Networks.” *Neurocomputing* 275:167–79.

Wu, Yuting, Mei Yuan, Shaopeng Dong, Li Lin, and Yingqi Liu. 2018c. “Remaining Useful Life Estimation of Engineered Systems Using Vanilla LSTM Neural Networks.” *Neurocomputing* 275:167–79.

Xu, Jingmang, Ping Wang, Boyang An, Xiaochuan Ma, and Rong Chen. 2018. “Damage Detection of Ballastless Railway Tracks by the Impact-Echo Method.” Pp. 106–14 in *Proceedings of the Institution of Civil Engineers-Transport*. Vol. 171. Thomas Telford Ltd.

Yang, Hongji, Jinhui Jiang, Guoping Chen, M. Shadi Mohamed, and Fan Lu. 2021a. “A Recurrent Neural Network-Based Method for Dynamic Load Identification of Beam Structures.” *Materials* 14(24):7846.

Yang, Hongji, Jinhui Jiang, Guoping Chen, M. Shadi Mohamed, and Fan Lu. 2021b. “A Recurrent Neural Network-Based Method for Dynamic Load Identification of Beam Structures.” *Materials* 14(24):7846.

Yang, Hongji, Jinhui Jiang, Guoping Chen, M. Shadi Mohamed, and Fan Lu. 2021c. “A Recurrent Neural Network-Based Method for Dynamic Load Identification of Beam Structures.” *Materials* 14(24):7846.

Yang, Hui, Weiming Yan, and Haoxiang He. 2016a. “Parameters Identification of Moving Load Using ANN and Dynamic Strain.” *Shock and Vibration* 2016.

Yang, Hui, Weiming Yan, and Haoxiang He. 2016b. “Parameters Identification of Moving Load Using ANN and Dynamic Strain.” *Shock and Vibration* 2016.

Yang, Yeong-Bin, J. D. Yau, Zhongda Yao, and Y. S. Wu. 2004. *Vehicle-Bridge Interaction Dynamics: With Applications to High-Speed Railways*. World Scientific.

Ye, X. W., Y. H. Su, and J. P. Han. 2014. "Structural Health Monitoring of Civil Infrastructure Using Optical Fiber Sensing Technology: A Comprehensive Review." *The Scientific World Journal* 2014.

Zhang, He, and Yuhui Zhou. 2023. "AI-Based Modeling and Data-Driven Identification of Moving Load on Continuous Beams." *Fundamental Research* 3(5):796–803.

Zhang, J., and W. Li. 1986. "Six-Force-Factor Identification of Helicopters." *Acta Aeronaut. Astronaut. Sin* 1986:2.

Zhang, Yi-Ming, Hao Wang, Jian-Xiao Mao, Zi-Dong Xu, and Yu-Feng Zhang. 2021a. "Probabilistic Framework with Bayesian Optimization for Predicting Typhoon-Induced Dynamic Responses of a Long-Span Bridge." *Journal of Structural Engineering* 147(1):04020297.

Zhang, Yi-Ming, Hao Wang, Jian-Xiao Mao, Zi-Dong Xu, and Yu-Feng Zhang. 2021b. "Probabilistic Framework with Bayesian Optimization for Predicting Typhoon-Induced Dynamic Responses of a Long-Span Bridge." *Journal of Structural Engineering* 147(1):04020297.

Zhang, Yi-Ming, Hao Wang, Jian-Xiao Mao, Zi-Dong Xu, and Yu-Feng Zhang. 2021c. "Probabilistic Framework with Bayesian Optimization for Predicting Typhoon-Induced Dynamic Responses of a Long-Span Bridge." *Journal of Structural Engineering* 147(1):04020297.

Zhang, Yi-Ming, Hao Wang, Jian-Xiao Mao, Zi-Dong Xu, and Yu-Feng Zhang. 2021d. "Probabilistic Framework with Bayesian Optimization for Predicting Typhoon-Induced Dynamic Responses of a Long-Span Bridge." *Journal of Structural Engineering* 147(1):04020297.

Zhong-Xian, Li, CHEN Feng, and WANG Bo. 2008a. "A BP Neural Network-Based Stage Identification Method for Moving Loads on Bridges." **工程力学** 25(9):85–092.

Zhong-Xian, Li, CHEN Feng, and WANG Bo. 2008b. "A BP Neural Network-Based Stage Identification Method for Moving Loads on Bridges." **工程力学** 25(9):85–092.

Zhou, J. M., Longlei Dong, Wei Guan, and Jian Yan. 2019a. "Impact Load Identification of Nonlinear Structures Using Deep Recurrent Neural Network." *Mechanical Systems and Signal Processing* 133:106292.

Zhou, J. M., Longlei Dong, Wei Guan, and Jian Yan. 2019b. "Impact Load Identification of Nonlinear Structures Using Deep Recurrent Neural Network." *Mechanical Systems and Signal Processing* 133:106292.

Zhu, X. Q., and S. S. Law. 1999. “Moving Forces Identification on a Multi-Span Continuous Bridge.” *Journal of Sound and Vibration* 228(2):377–96.

Zhu, X. Q., and S. S. Law. 2002. “Moving Loads Identification through Regularization.” *Journal of Engineering Mechanics* 128(9):989–1000.

APPENDIX

The relevant codes for this thesis can be found at the following link:

https://github.com/masnun11/Masnun-MS-Thesis.git

VITA

MD MASNUN RAHMAN

Email: masnun11@gmail.com

EDUCATION

Md Masnun Rahman completed his Master of Science in Civil Engineering at The University of Texas Rio Grande Valley in Edinburg, Texas, in May 2024. Prior to that, he earned his Bachelor of Science in Civil and Environmental Engineering from Shahjalal University of Science and Technology in Bangladesh, which he completed in December 2018.