13th International Conference on
Structural Health Monitoring of Intelligent Infrastructure
DOI: 10.3217/978-3-99161-057-1-146

On a data compression technique for acceleration signals from a railway bridge

Pranav Yadav[1, 0009-0003-9229-4797], Vaibhav Gupta[1, 0009-0002-6651-4154], U. Saravanan[1, 0000-0001-8565-0632]
[1]Department of Civil Engineering, Indian Institute of Technology Madras, Adyar, Chennai 600036 Tamil Nadu, India
email: pranavyadav23103@gmail.com, vaibgupta24@gmail.com, saran@iitm.ac.in

ABSTRACT: Structural health monitoring (SHM) is essential for ensuring bridge safety and longevity. Under dynamic loads, such as train traffic, acceleration data from sensors offers valuable insights into the condition of the structure. Vehicle bridge interaction models required to predict the acceleration time histories involve numerous parameters for rail traffic. Also, model-based methods have a trade-off between high-fidelity, computationally intensive, and less accurate models. To overcome these limitations, this study introduces a deep learning (DL) algorithm to identify changes in the bridge. However, large datasets resulting from high-frequency sampling and long observation periods pose computational challenges as the train passes over the bridge. To address this, down sampling is employed, reducing data complexity while preserving essential features of the signal. The approach is demonstrated using acceleration data recorded at a node point of a railway truss bridge during train passage. An Autoencoder is employed, compressing high-dimensional data into a low-dimensional latent space, and a deep neural network (DNN) is applied to the latent space, incorporating a measurement loss function to estimate the system parameters. This framework ensures computational efficiency and data integrity, enabling precise system parameter estimation and showcasing its effectiveness in real-life bridge SHM.

KEY WORDS: Data compression; Autoencoder; Deep neural network; Deep learning; Railway steel truss bridge; Structural health monitoring.

1 INTRODUCTION

Structural health monitoring (SHM) has advanced significantly with the development of sensing technologies and data collection capabilities [1],[2]. It plays a crucial role in ensuring the safety, durability, and reliability of critical structures, particularly railway bridges that experience continuous dynamic loading [3],[4]. Accurately estimating structural parameters such as cross-sectional area, damping coefficients, and stiffness properties is fundamental for detecting structural degradation, damage progression, and potential failures [5],[6]. Traditional methods, including finite element model (FEM)-based approaches, rely on high-fidelity models and experimental calibration, which are computationally expensive and susceptible to modeling inaccuracies [7],[8],[9]. These challenges necessitate data-driven approaches that leverage machine learning (ML) techniques for efficient and accurate structural assessment [10],[11].

Neural networks, a part of ML techniques, have emerged as powerful tools in SHM, enabling automated inspection processes and addressing the growing complexity of intelligent monitoring systems. Their primary advantages include automatic feature extraction, effectiveness in handling noisy datasets, and accurate modeling of nonlinear relationships [12]. Nevertheless, conventional artificial neural networks (ANNs) frequently encounter issues such as convergence to local minima, susceptibility to overfitting, and limited ability for deeper feature extraction due to shallow network architectures [13]. Recent advancements in deep learning (DL) have introduced data-driven SHM approaches, wherein structural parameters are inferred directly from measured vibration signals [14],[15] to overcome the above mentioned challenges. Such DL frameworks effectively capture complex nonlinear relationships inherent in multi-sensor datasets, making them particularly suitable for classification and regression tasks [16]. Nevertheless, standalone DL models share several limitations commonly observed with traditional ANNs, including high computational efficiency, overfitting, and convergence issues, highlighting the need for hybrid DL methodologies [17]. Over the past decade, several hybrid DL approaches have been developed for bridge damage detection and condition assessment. One promising approach is the Autoencoder, which is known for its ability to perform dimensionality reduction and data compression effectively.

To highlight a few instances, an unsupervised Autoencoder approach proposed in [18] achieved real-time bridge damage detection directly from raw acceleration data, although it was limited to single-sensor applications without the capability for damage localization or quantification. In another study [19], a hybrid methodology combining statistical modeling, neural networks, and deep support vector domain description demonstrated effective real-time damage detection with minimal false alarms; however, this method lacked localization capability, exhibited reduced performance with multi-sensor datasets, and has not been tested on full-scale structures. [20] introduced an Autoencoder-based method emphasizing the relationships among natural frequencies and mode shapes. Further extending this concept, [21] proposed a deep sparse Autoencoder specifically used for structural damage detection using these vibration characteristics. [22] developed a two-level hybrid learning framework, employing unsupervised learning for preliminary damage detection followed by supervised validation, demonstrating its efficacy through

DOI: 10.3217/978-3-99161-057-1-146

numerical simulations of concrete beams and experimental validation using laboratory frames. Additionally, [23] introduced an unsupervised deep neural network (DNN) method combining a deep Autoencoder with a one-class support vector machine (SVM), eliminating the reliance on extensive labeled datasets by utilizing only intact structural data.

While prior research has explored hybrid DL models involving Autoencoders for data compression, most studies have primarily addressed laboratory-scale experiments or relied on supervised learning methods. In contrast, this study introduces an integrated framework combining unsupervised Autoencoder-based compression and a supervised DNN model to effectively handle on-field acceleration signals coming in the form of latent space representation from the Autoencoder. The proposed Autoencoder-DNN framework is computationally efficient and provides an effective compression strategy that preserves the important structural characteristics of real-world acceleration data by extracting relevant features, facilitating accurate and reliable estimation of structural parameters in railway steel truss bridge members.

The methodology presented herein constitutes the preliminary phase of a comprehensive, multi-stage investigation to extend parameter estimation from an individual joint (node point considered in this study) to the entire railway steel truss bridge structure. By demonstrating the effectiveness and robustness of the developed approach through this single-node case study, the present work lays a solid foundation for data-driven SHM of complex infrastructure with relevance to railway bridge systems.

The structure of this paper is as follows: **Section 2** details the bridge type and instrumentation scheme. **Section 3** explains data collection and segmentation. **Section 4** presents the proposed methodology, including the Autoencoder and the DNN model. **Section 5** discusses the results, focusing on model accuracy, anomaly detection, and computational efficiency. Finally, **Section 6** concludes the key findings and explores potential directions for future research.

2 BRIDGE DETAILS AND INSTRUMENTATION SCHEME

The Pamban Bridge is a railway steel truss bridge linking Rameswaram on Pamban Island to mainland India, as shown in Figure 1. Commissioned in 1914, it was India's first sea bridge. Although most spans are conventional I-plate girders on concrete piers, the bridge features a notable double-leaf bascule section that pivots to let ships and barges pass. This movable portion, designed by Scherzer, is counterbalanced and pivots around a horizontal axis, with the superstructure rolling atop the track girder. Each leaf consists of a rigid jaw-and-tongue system to transfer shear without moments and is further subdivided into north and south trusses [24].

Since the bridge endures harsh marine conditions, corrosion is a significant concern, making it essential to evaluate any loss of cross-sectional area for structural assessments. A total of 40 uniaxial accelerometers have been installed at various bottom nodes in biaxial mode on the bridge's Mandapam and Pamban truss segments in both the north and south directions, as shown in Figure 2.

Figure 1. Orientation of the Pamban bridge.

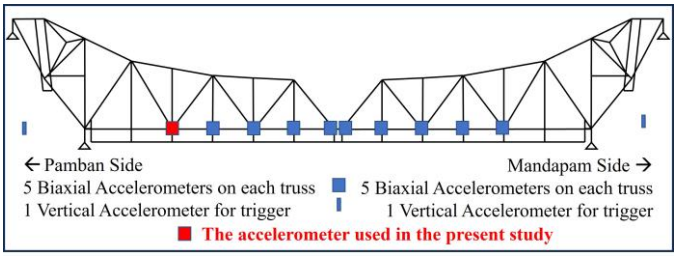

Figure 2. Location of accelerometers on the bridge.

3 DATA COLLECTION AND SEGMENTATION

The acceleration data used in this study was collected from a bottom node marked with a red box in Figure 2, where two uniaxial accelerometers were installed in biaxial mode. Measurements were taken in both the x and y directions, with the x direction corresponding to the direction of train movement and the y direction representing the direction of gravity. When detecting a train pass event, the DAQ system continuously recorded acceleration data for 480 seconds, which was sufficient to cover the train passage with an ample margin before and after the event.

The sampling frequency of the acceleration was set at 600 Hz, resulting in 288,000 data points for each recording session. After analyzing acceleration signals from 150 train passages for each direction, it was observed that the train-induced vibrations predominantly occupied the initial segment of the recorded signal. At the same time, the remaining portion primarily consisted of ambient noise. Based on this observation, a consistent segmentation approach was adopted, wherein the first 1,60,000 data points were extracted from each recording for further analysis. This segmentation ensured the retention of train-induced dynamic responses while excluding prolonged noise periods, thereby reducing the number of data points in the subsequent processing stages.

4 METHODOLOGY

This study adopts a multi-step methodology comprising data pre-processing and data compression to improve the performance, scalability and computational efficiency of the Autoencoder-DNN framework for parameter estimation. The overall flowchart is presented in Figure 3, which outlines the

DOI: 10.3217/978-3-99161-057-1-146

key steps involved. Each step is further detailed in the following sections, explaining the implemented approach.

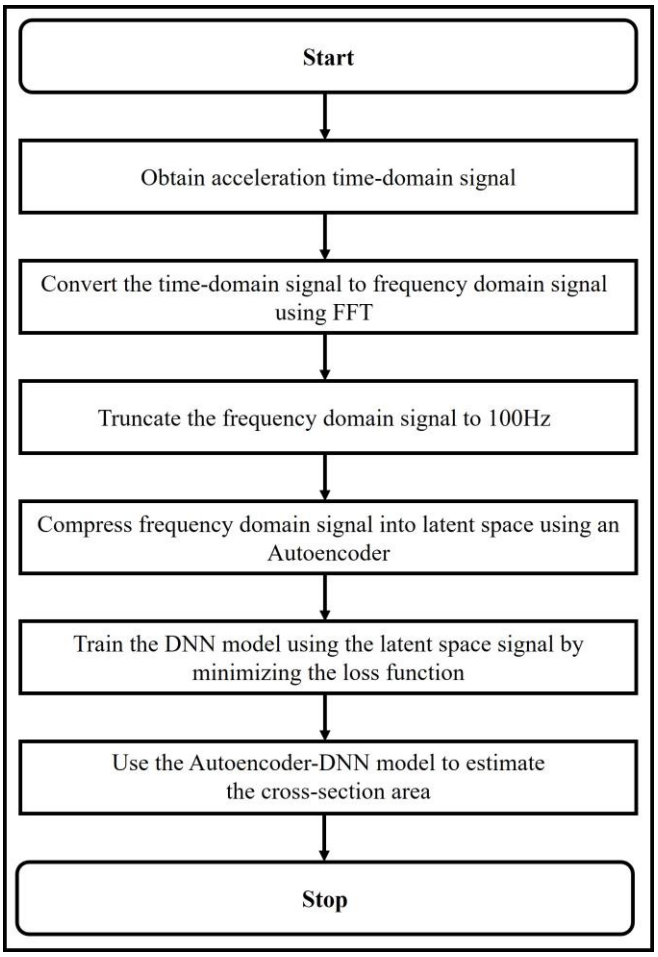

Figure 3. Flowchart depicting the methodology.

4.1 Data Preprocessing

After obtaining the train pass signals, the time-domain data was converted into the frequency domain using the fast Fourier transform (FFT). This transformation was essential to facilitate efficient data processing, as the large dataset in the time domain posed computational challenges for subsequent analyses, notably when applying the DNN model. The FFT utilizes the symmetric property of real-valued signals, reducing the adequate number of data points by half while retaining the qualities of the original signal, which optimizes storage and computational requirements. The amplitude spectrum of the transformed signal was carefully examined, revealing that frequency components beyond 100 Hz have acceleration amplitudes 1/4th less than the maximum amplitude. Therefore, a frequency threshold of 100 Hz was adopted to truncate the signal, effectively eliminating higher-frequency content while retaining the dominant spectral content. This transformation approach introducing symmetry and frequency truncation has resulted in a reduced dataset of approximately 26,666 data points. Figure 4 illustrates the process, where the time-domain acceleration signal is converted to the frequency domain and truncated up to 100 Hz. This significant reduction enhanced computational efficiency and preserved the essential dynamic characteristics of the train-induced vibrations. However, despite the reduction, the large dimensionality of the dataset still posed computational challenges, particularly concerning memory requirements and the convergence efficiency of the DNN model.

4.2 Data Compression Using Autoencoder

To mitigate computational challenges, a data compression technique based on an Autoencoder is implemented. The Autoencoder used in this study consists of two key components: an encoder, which compresses the input data into a lower-dimensional latent space, and a decoder, which reconstructs the data from this compressed representation. Figure 5 illustrates the step-by-step application of the Autoencoder, detailing the transformation process at each stage.

Since acceleration signals exhibit distinct dynamic characteristics in the x and y directions, separate Autoencoders are implemented for each direction. The Autoencoders applied in both directions maintain an identical architecture, layer structure, and activation functions, as shown in Figure 5, but they are trained independently to capture the unique frequency and amplitude variations inherent to each direction. This ensures that the learned representations accurately reflect the direction-specific vibration behavior while preserving consistency in the latent space size.

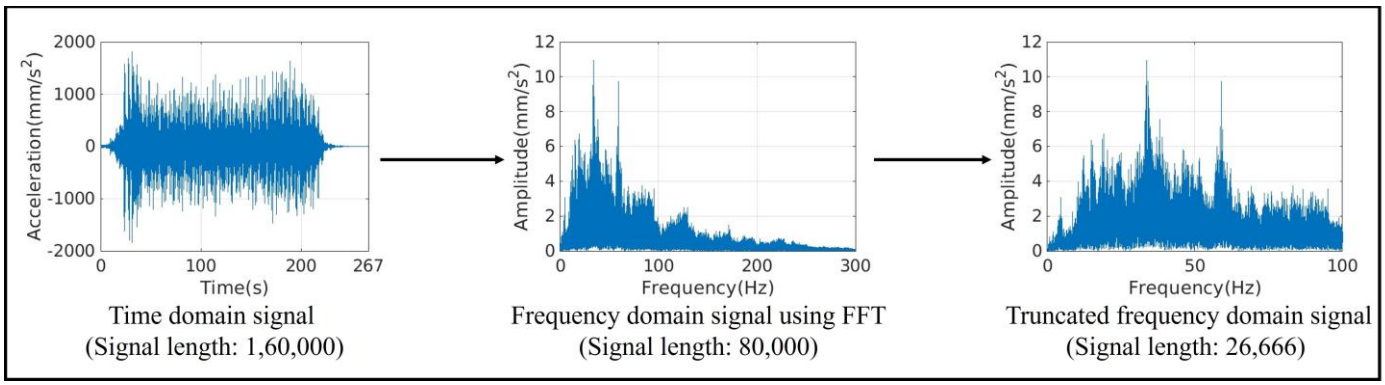

Figure 4. Data preprocessing workflow.

DOI: 10.3217/978-3-99161-057-1-146

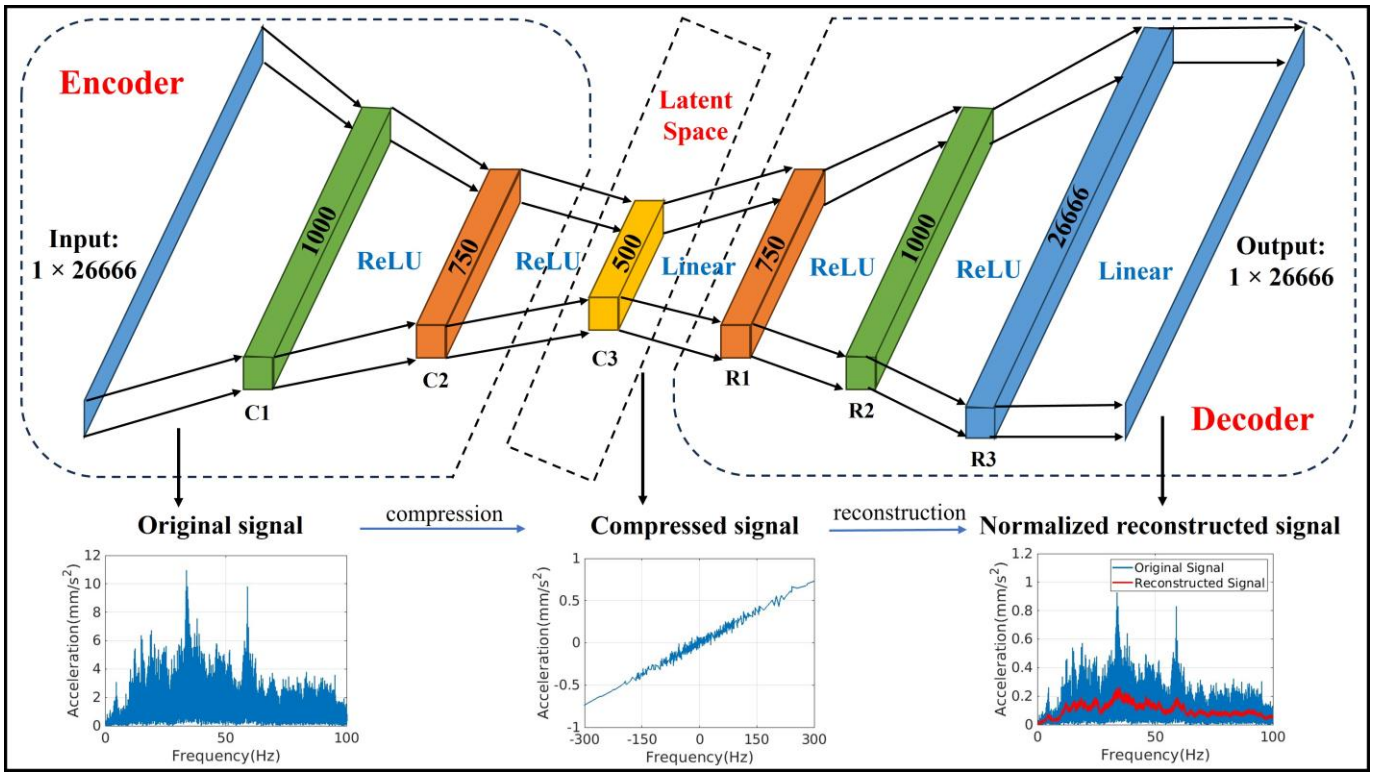

Figure 5. The architecture of the Autoencoder.

4.2.1 Encoder network

It consists of three layers labeled C1 to C3. The ReLU activation function is applied in layers C1 and C2, while the linear activation function is used in layer C3. As described in the previous section, the network takes preprocessed input data of size 1 × 26666 for one train pass. However, since the autoencoder is trained on data from 112 (75% of 150 train passes) train passes, the input data size becomes 1 × 26666 × 112 for each direction. The data is progressively compressed through each layer, reducing its dimensionality from 1 × 26666 × 112 in the input layer to 1 × 1000 × 112 in C1**,** 1 × 750 × 112 in C2, and finally to 1 × 500 × 112 in C3, which represents the latent space.

4.2.2 Decoder network

The decoder network consists of three layers, labeled R1 to R3, designed to restore the compressed latent space to its original dimension. The ReLU activation function is applied in layers R1 and R2, while the linear activation function is used in the final layer R3. The input to the reconstruction network is the latent space representation of size 1 × 500 × 112. The data is gradually reconstructed through each layer, expanding from 1 × 750 × 112 in R1, then to 1 × 1000 × 112 in R2, and ultimately restored to its original size of 1 × 26666 × 112 in R3 at the output layer.

4.2.3 Dropout Regularization

To prevent overfitting and enhance the generalization capability of the Autoencoder model, dropout regularization is applied to each layer except the latent space layer. A 10% dropout rate is used, randomly deactivating 10% of the neurons to zero during each training iteration.

4.2.4 Loss function

The Autoencoder is trained using the mean squared error (MSE) loss function, which measures the reconstruction error between the original input data and the reconstructed output. The MSE loss is defined as:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \quad (1)$$

where x_i represents the original input data, $\hat{x}_i$ represents the reconstructed output, and N is the total number of train passes. The loss function is minimized during training, ensuring that the reconstructed signal closely approximates the characteristics of the input signal. This optimization enables the Autoencoder to retain essential features of the input data, making the compressed representation more computationally efficient.

4.2.5 Optimization

The Adam optimizer employed to train the Autoencoder is widely recognized for its effectiveness in various applications. It is well-suited for non-stationary objectives and problems with noisy or sparse gradients [25]. The optimization is performed over 30 epochs, minimizing the MSE loss to improve reconstruction accuracy while retaining the essential features of the input data.

4.3 Integration with DNN Model

Integrating the compressed latent space representations of the frequency-domain acceleration data with the DNN model helps in parameter estimation. Figure 6 illustrates the step-by-step application of the DNN model.

DOI: 10.3217/978-3-99161-057-1-146

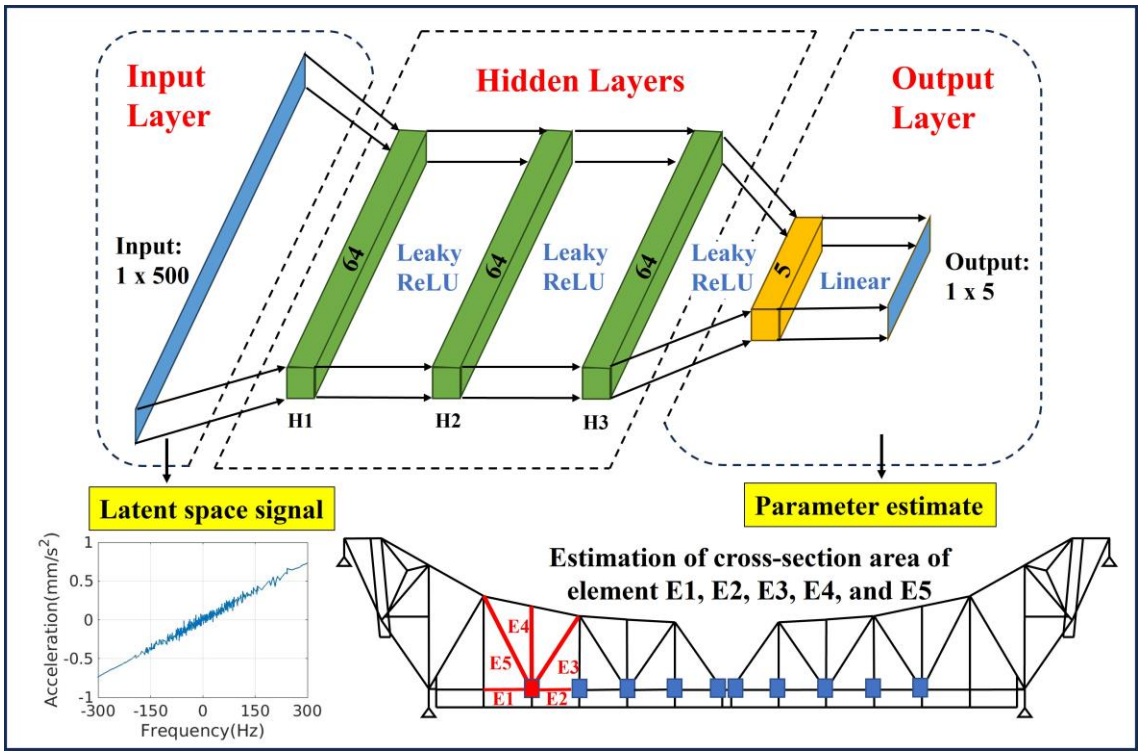

Figure 6. The architecture of the DNN model.

4.3.1 Input to DNN model

The compressed latent space representations of the frequency-domain acceleration data in both the x and y directions are integrated within the DNN model to estimate the cross-sectional area of the truss members. Since acceleration responses exhibit distinct dynamic characteristics in different directions, the compressed x and y components are concatenated to form a combined 500 × 2 input vector for each train pass.

This combined representation allows the DNN model to capture the full structural behavior influenced by both directional responses. The combined 500 × 2 × 112 input vector for all train passes is passed into the DNN model for training, enabling the model to learn the system's dynamics across multiple loading conditions.

4.3.2 DNN architecture

As mentioned earlier, the DNN model receives compressed latent space representations of the acceleration data from the Autoencoder, presented in a concatenated format that includes data from both directions. After conducting hyperparameter tuning using the grid search method, Table 1 displays the final values of the parameters used in the DNN model.

Table 1. DNN model parameters and their values.

Parameter	Value
Hidden Layers	3
Hidden Neurons	64
Output Neurons	5
Hidden Activation	Leaky ReLU
Dropout Layer	0
Learning Rate	0.001
Optimizer	ADAM
Epochs	20000

The final output layer estimates the cross-sectional area of the truss members connected, which is highlighted by a red color line in Figure 6. The network effectively learns the mapping from the compressed input to the target output while integrating the measurement loss function to improve accuracy.

4.3.3 Measurement loss function

The DNN model incorporates a measurement loss to improve the accuracy of the estimated area. This loss minimizes the difference between the true cross-sectional area (provided in Table 2) and the predicted area obtained from the DNN model using the MSE loss function defined as:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{5}\left(\left(x_{true}^{j}\right)^{i} - \left(x_{predicted}^{j}\right)^{i}\right)^{2} \qquad (2)$$

where x_{true}^{j} is the true value of the j^{th} parameter of the system to be estimated – the cross-sectional area and $x_{predicted}^{j}$ is the corresponding predicted value from the DNN model.

Table 2. The table shows the true values of the cross-sectional area.

Parameter	Value (mm²)
Element 1 (E1)	24400
Element 2 (E2)	12178
Element 3 (E3)	15800
Element 4 (E4)	20100
Element 5 (E5)	8700

DOI: 10.3217/978-3-99161-057-1-146

4.4 *Hypothesis for Estimating Cross-Sectional Area from a Single Node Response*

The study focuses on a single node among those where bi-axial accelerations are recorded. This node is structurally connected to five truss elements, each assumed to have known lengths (L1, L2, L3, L4, and L5) and elastic modulus, while the cross-sectional areas (A1, A2, A3, A4, and A5) are unknown and to be estimated. The arrangement of the members is as shown in Figure 7. The assumption that all five truss elements are fixed at the other end is not true in the actual truss structure. However, this assumption allows checking of the algorithm in a smaller setting.

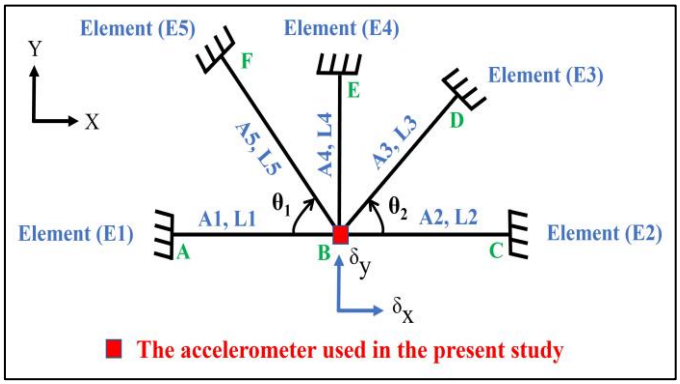

Figure 7. Schematic of the monitored truss node with five connected members. The bi-axial accelerometer is positioned at the node to record dynamic responses during train passages.

The mass matrix is formulated based on the shape functions that approximate the displacement. The consistent element mass matrix formulated in global coordinates for member AB is given by:

$$[M_{AB}] = \frac{\rho AL}{6}\begin{bmatrix} 2c^2 & 2cs & c^2 & cs \\ 2cs & 2s^2 & cs & s^2 \\ c^2 & cs & 2c^2 & 2cs \\ cs & s^2 & 2cs & 2s^2 \end{bmatrix} \quad (1)$$

where c and s denote cos(θ) and sin(θ), respectively, θ is the counter- clockwise angle measured with respect to the positive x-axis, ρ is the density of the material, A is the cross-sectional area, and L is the length of the element.

Meanwhile, the stiffness matrix is derived under the assumption of linear elastic behaviour. The stiffness matrix used in global coordinates for member AB is given by:

$$[K_{AB}] = \frac{EA}{L}\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \quad (2)$$

where E is Young's modulus, A is the cross-sectional area, and L is the element length.

Similarly, the global mass and stiffness matrices for all five connected truss elements are formulated and subsequently condensed by applying suitable boundary conditions according to the idealization results in a 2×2 system of equations relating the translational degrees of freedom at the monitored node to the forces acting on the node due to the train. During a train passage, this node undergoes dynamic displacement, and the bi-axial accelerometer captures its response along the train movement (x-direction) and gravity direction (y-direction). Given that the displacement (u_x and u_y) and acceleration (a_x and a_y) responses are known, and the excitation is indirectly inferred from the train passage characteristics (f_x and f_y), the cross-sectional areas, which influence the mass and stiffness of each member, leave a unique imprint on the node's dynamic response as reflected in:

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}\begin{Bmatrix} u_x \\ u_y \end{Bmatrix} + \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\begin{Bmatrix} a_x \\ a_y \end{Bmatrix} = \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} \quad (3)$$

where $K_{11} = \frac{EA_1}{L_1} + \frac{EA_2}{L_2} + \frac{EA_3c_2^2}{L_3} + \frac{EA_5c_1^2}{L_5}$; $K_{12} = K_{21} = \frac{EA_3c_2s_2}{L_3} - \frac{EA_5c_1s_1}{L_5}$; $K_{22} = \frac{EA_3s_2^2}{L_3} + \frac{EA_4}{L_4} + \frac{EA_5s_1^2}{L_5}$ and $M_{11} = \frac{\rho A_1L_1}{3} + \frac{\rho A_2L_2}{3} + \frac{\rho A_3L_3c_2^2}{3} + \frac{\rho A_5L_5c_1^2}{3}$; $M_{12} = M_{21} = \frac{\rho A_3L_3c_2s_2}{3} - \frac{\rho A_5L_5c_2s_2}{3}$; $M_{22} = \frac{\rho A_3L_3s_2^2}{3} + \frac{\rho A_4L_4}{3} + \frac{\rho A_5L_5s_1^2}{3}$, are respective components of the effective stiffness and mass matrices. Here, c_1 and s_1 denote $\cos(\theta_1)$ and $\sin(\theta_1)$, c_2 and s_2 denote $\cos(\theta_2)$ and $\sin(\theta_2)$, respectively, and θ is the counter-clockwise angle measured with respect to the positive x-axis.

Equation (3) represents a possible relation between the acceleration and the cross-sectional area of connected elements. The proposed framework finds this relationship between the measured acceleration and cross-sectional areas in a data-driven manner.

5 RESULTS AND DISCUSSIONS

This section presents the performance evaluation of the Autoencoder-DNN framework, focusing on data compression, reconstruction errors, system parameter estimation, computational efficiency, and anomaly detection. The model is trained and tested using 150 train pass datasets for each direction, with a 75%–25% train-test split, ensuring the model generalizes well to unseen data.

5.1 *Autoencoder Performance and Reconstruction Loss Behavior*

The Autoencoder uses a validation set of a split ratio of 75%-25% instead of the test set during training to monitor the reconstruction loss and ensure that the model generalizes effectively to unseen data. This is important, as the loss terms for the autoencoder should remain consistent across the training and validation sets to facilitate accurate compression and reconstruction.

Figure 8 illustrates the training and validation loss curves, showing a smooth and stable convergence of order 10^{-3}, which indicates that the latent space representation effectively captures the dominant structural features of the train-induced vibrations. The final loss values confirm that the compression process does not introduce significant deviations, making the latent representation reliable for subsequent DNN-based analysis.

DOI: 10.3217/978-3-99161-057-1-146

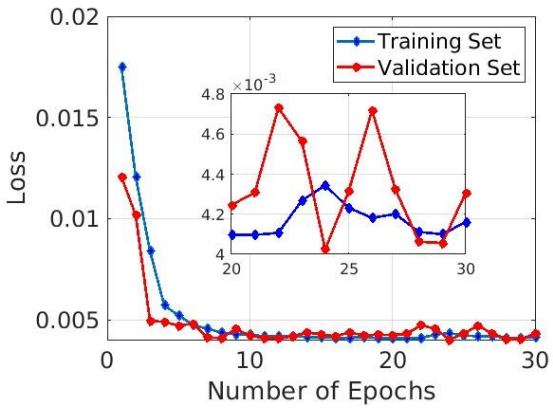

Figure 8. Comparison of loss in the training set and in the validation set.

Additionally, the method takes approximately 240 seconds to compress the acceleration data into the latent space for both directions, highlighting its effectiveness in data compression and computational speed.

5.2 DNN Model Parameter Estimation

Once the Autoencoder compresses the frequency domain acceleration data into the latent space, the DNN model is trained using 75% of the dataset and subsequently tested on the remaining 25% of unseen data. The primary objective of the DNN model is to estimate the cross-sectional area of the truss members by minimizing the measurement loss function. Figure 9 illustrates that during the training process, the MSE loss consistently decreases as the number of epochs increases, leading to stable convergence of the estimated parameters.

Once stable convergence is achieved, while observing the parameter values in the training dataset, it is noticed that the cross-sectional area of the truss members converges to their true values. To interpret this result physically, for an idealized truss structure without structural damage, the cross-sectional area values should remain constant across all train passes. Figure 10 shows the area estimation of element (E2) for five train passes. It can be observed that the cross-sectional area is converging to its true value (reported in Table 2). Similarly, for all the train passes and for all the truss members considered, the results across the entire training dataset confirm that the estimated area values closely align with the expected true values.

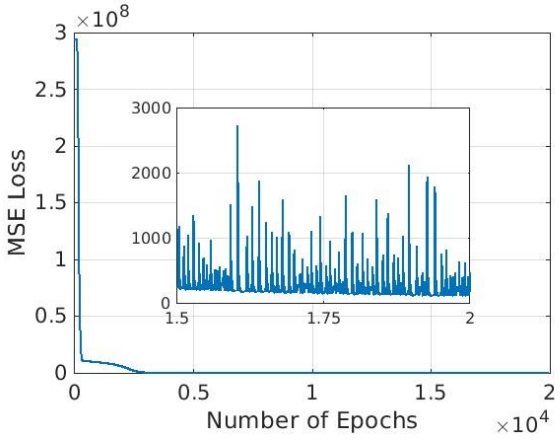

Figure 9. Convergence of measurement loss at each epoch.

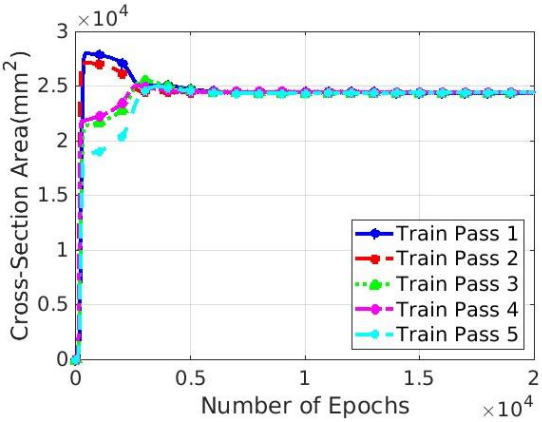

Figure 10. The cross-sectional area of element (E2) using the training data.

5.2.1 Performance Evaluation of the DNN Model on the Test Data

Once the DNN model is trained, its performance is evaluated using the test data. Figure 11 illustrates the values of the cross-sectional area obtained from the DNN model for all the test datasets. The coefficient of variation (CoV) is computed for the estimated cross-sectional area for all the truss members and tabulated in Table 3. It is observed that the CoV values for all the truss members are less and identical. This uniformity in CoV shows that the DNN model exhibits consistent relative variability in its predictions across different members. Such behavior indicates stable model performance under test conditions, with no bias or irregularity in estimating cross-sectional area. However, this observation also highlights the need for further analysis to ensure that the model is sufficiently sensitive to localized structural variations, and hence, the model is tested through false data simulating sensor fault, noise, and anomalies, as explained in the next section.

Table 3. The table shows the coefficient of variation of the predicted cross-sectional area under test data.

Parameter	Coefficient of Variation (%)
Element 1	0.18
Element 2	0.18
Element 3	0.18
Element 4	0.18
Element 5	0.18

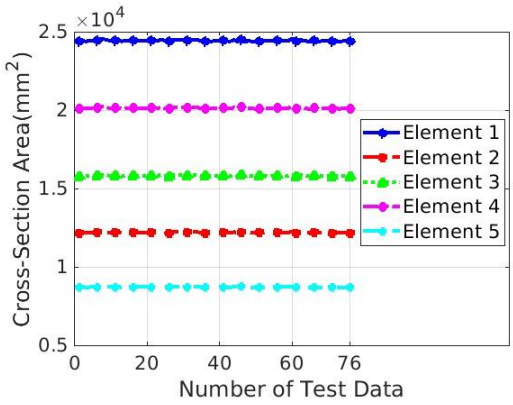

Figure 11. The cross-sectional area of the truss members for the test data.

DOI: 10.3217/978-3-99161-057-1-146

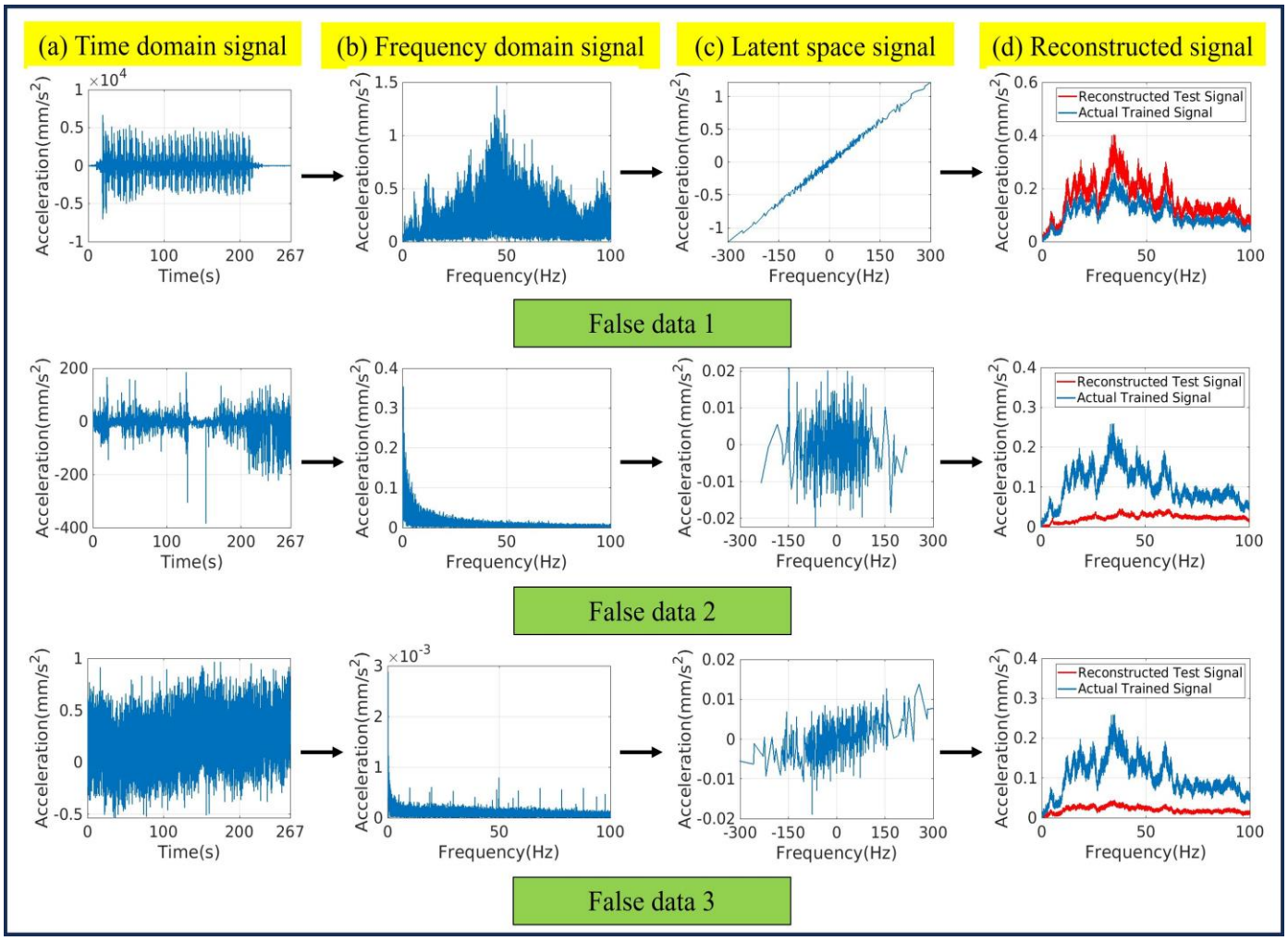

Figure 12. Illustration of the false data for the DNN model, showing (a) the time-domain acceleration signal, (b) its frequency-domain representation, (c) the latent space obtained from the Autoencoder, and (d) a comparison between the normalized reconstructed false data and original data, highlighting their differences.

5.3 Computational Efficiency

The DNN model employed in this study took approximately 100 seconds to estimate parameters involving 150 train passages in both directions, emphasizing the model's effectiveness in balancing accuracy and computational speed in parameter estimation.

5.4 Anomaly Detection Through False Data Injection

To evaluate the robustness of the Autoencoder-DNN framework and its ability to detect inconsistencies in the input data, three false datasets are used in this study. These false data are provided to check whether the proposed framework can detect potential structural anomalies, sensor noise, or environmental influences, which introduce variations in the acceleration response in the on-field conditions. To be specific, the first false data replaces the train pass signal with data collected from a different node point, the second exhibits a faulty sensor that fails to capture the correct train pass signal, and the third represents an anomaly where the sensor is unable to record any valid data—potentially caused by false triggers. Figure 12 illustrates the procedure followed for preparing these false datasets prior to their input into the DNN model. It sequentially presents (a) the acceleration signal in the time domain, (b) its corresponding frequency-domain representation, (c) the latent-space representation obtained from the Autoencoder, and (d) a comparison between the reconstructed false data and original data from the trained Autoencoder, clearly highlighting differences between the two signals. When the latent space of the false data was passed onto the DNN model, the predicted cross-sectional area for all the truss members exhibited notable deviations, as shown in Table 4. It is observed that the percentage deviation of the cross-sectional area for all the truss members exhibits the same percentage change for the different false datasets, regardless of the specific member location. For false data 1, 2, and 3, the percentage deviation is approximately 4.30%, 1.10%, and 3.30%, respectively. The statistically significant percentage change across members highlights the framework's ability to detect gross inconsistencies in the input signals, though it also indicates a need for future enhancement to improve sensitivity to localize anomalies. It may be that localization would occur when multiple nodes are used. Thus, the proposed framework provides the identification of faulty sensor readings, environmental influences, or potential structural changes. However, the classification of the signal remains a challenge.

DOI: 10.3217/978-3-99161-057-1-146

Table 4. The table shows the percentage deviation of the predicted cross-sectional area values with respect to the true cross-sectional area values for the false data.

Parameter	Percentage deviation from the true cross-sectional area value (%)		
	False data 1	False data 2	False data 3
Element 1	4.30	1.10	3.30
Element 2	4.30	1.10	3.30
Element 3	4.30	1.10	3.30
Element 4	4.30	1.10	3.30
Element 5	4.30	1.10	3.30

6 CONCLUSION

This study introduced an Autoencoder-DNN framework for data-driven structural parameter estimation using train-induced vibration responses. The Autoencoder effectively compressed high-dimensional acceleration data in the frequency domain into a lower-dimensional latent space, preserving critical structural features while significantly enhancing computational efficiency. The DNN model, trained on the compressed latent representation, allows for accurate estimation of the cross-sectional area of the truss members connected to the node point considered in this study. The model exhibited stable performance on test datasets, with low prediction errors and consistent coefficients of variation, indicating reliable and uniform estimation capabilities. Furthermore, the framework effectively detected inconsistencies when introduced with false datasets simulating sensor faults, noise, and anomalies. In all cases, uniform deviations in the estimated cross-sectional area confirmed the model's robustness in identifying global anomalies in the input data.

Overall, the proposed framework offers an Autoencoder-based compression, supervised DNN model, and automated anomaly detection, making it a scalable and computationally efficient tool for large-scale truss structures in real-world applications. Future work will focus on improving the framework's sensitivity to localize damage and integrating the full-scale truss bridge model and extended datasets to further examine the framework's generalization capability.

ACKNOWLEDGMENTS

The authors thank Southern Railways for funding this work and allowing the utilization of the collected data for analysis.

REFERENCES

[1] S. Park, H. S. Park, J. H. Kim, and H. Adeli, 3D displacement measurement model for health monitoring of structures using a motion capture system, Measurement, 59, 352–362, 2015.
[2] J. P. Amezquita-Sanchez and H. Adeli, Signal processing techniques for vibration-based health monitoring of smart structures, Archives of Computational Methods in Engineering, 23, 1–15, 2016.
[3] V. Gupta, and U. Saravanan, Condition monitoring of steel truss bridge using acceleration data, In: Proceedings of the Fourteenth International Workshop on Structural Health Monitoring (IWSHM), September 12-14, 2023.
[4] V. Gupta and U. Saravanan, Comparative study on PSD and HHT techniques for condition assessment of steel truss bridge using acceleration data, Proceedings of the 10th European Workshop on Structural Health Monitoring (EWSHM 2024), June 10-13, 2024 in Potsdam, Germany, e-Journal of Nondestructive Testing Vol. 29(7), 2024.
[5] C. R. Farrar, S. W. Doebling, and D. A. Nix, Vibration–based structural damage identification, Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences, 359(1778), 131–149, 2001.
[6] E. P. Carden and P. Fanning, Vibration based condition monitoring: a review, Structural health monitoring, 3(4), 355–377, 2004.
[7] M. Azimi, A. D. Eslamlou, and G. Pekcan, Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review, Sensors, 20(10), 2778, 2020.
[8] G. Gui, H. Pan, Z. Lin, Y. Li, and Z. Yuan, Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection, KSCE Journal of Civil Engineering, 21, 523–534, 2017.
[9] F. Wedel and S. Marx, Application of machine learning methods on real bridge monitoring data, Engineering Structures, 250, 113365, 2022.
[10] C. R. Farrar and K. Worden, Structural health monitoring: a machine learning perspective, Chichester, West Sussex, UK; Hoboken, NJ: Wiley, 2013.
[11] R. de Almeida Cardoso, A. Cury, and F. Barbosa, Automated real-time damage detection strategy using raw dynamic measurements, Engineering Structures, 196, 109364, 2019.
[12] R. P. Bandara, T. H. Chan, and D. P. Thambiratnam, Frequency response function based damage identification using principal component analysis and pattern recognition technique, Engineering Structures, 66, 116–128, 2014.
[13] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications, Mechanical systems and signal processing, 147, 107077, 2021.
[14] X. Jian, H. Zhong, Y. Xia, and L. Sun, Faulty data detection and classification for bridge structural health monitoring via statistical and deep-learning approach, Structural Control and Health Monitoring, 28(11), e2824, 2021.
[15] D. Hajializadeh, Deep learning-based indirect bridge damage identification system, Structural health monitoring, 22(2), 897–912, 2023.
[16] M. Z. Sarwar and D. Cantero, Vehicle assisted bridge damage assessment using probabilistic deep learning, Measurement, 206, 112216, 2023.
[17] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, Hands-on bayesian neural networks—a tutorial for deep learning users, IEEE Computational Intelligence Magazine, 17(2), 29–48, 2022.
[18] V. Giglioni, I. Venanzi, V. Poggioni, A. Milani, and F. Ubertini, Autoencoders for unsupervised real-time bridge health assessment, Computer-Aided Civil and Infrastructure Engineering, 38(8), 959–974, 2023.
[19] S. Shi, D. Du, O. Mercan, E. Kalkan, and S. Wang, A novel unsupervised real-time damage detection method for structural health monitoring using machine learning, Structural Control and Health Monitoring, 29(10), e3042, 2022.
[20] C. S. N. Pathirage, J. Li, L. Li, H. Hao, W. Liu, and P. Ni, Structural damage identification based on autoencoder neural networks and deep learning, Engineering structures, 172, 13–28, 2018.
[21] C. S. N. Pathirage, J. Li, L. Li, H. Hao, W. Liu, and R. Wang, Development and application of a deep learning–based sparse autoencoder framework for structural damage identification, Structural Health Monitoring, 18(1), 103–122, 2019.
[22] H. Sarmadi and A. Entezami, Application of supervised learning to validation of damage detection, Archive of Applied Mechanics, 91(1), 393–410, 2021.
[23] Z. Wang and Y.-J. Cha, Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage, Structural Health Monitoring, 20(1), 406–425, 2021.
[24] V. Gupta and U. Saravanan, Bridge condition monitoring using frequency domain decomposition method. In International Operational Modal Analysis Conference. Springer, 2024.
[25] F. Ni, J. Zhang, and M. N. Noori, Deep learning for data anomaly detection and data compression of a long-span suspension bridge, Computer-Aided Civil and Infrastructure Engineering, 35(7), 685–700, 2020.