"Why Should I Trust You?": Exploring Interpretability in Machine Learning Approaches for Indirect SHM

Yifu LAN, Zhenkun LI, and Weiwei LIN
Aalto University, Espoo, Finland
yifu.lan@aalto.fi

Abstract. Currently, machine learning (ML) methods are widely adopted in structural health monitoring (SHM), yet they are still mostly black boxes. On the other hand, given the significant responsibility associated with SHM, understanding the rationale behind it is critically important. In some cases, even experienced experts have difficulties finding evidence related to structural integrity within complex structural signals. Thus, solely relying on these black-box SHM systems carries inherent risks. Trustworthiness is the key for decision-makers when planning to act on predictions or deciding whether to deploy a new model. This understanding can also offer insights about the models, transforming untrustworthy models or predictions into reliable ones. The indirect SHM method using passing vehicles, an emerging technique in the past two decades, offers a rapid and cost-effective solution for bridge monitoring. Its signal components are affected by factors such as vehicle dynamics and road roughness, making them more complex than those in the direct method. Although ML methods have shown promising results in this domain, their results require further explanation. In this work, an interpretation tool is proposed to interpret the result prediction of ML methods in indirect SHM. The trustworthiness of models is demonstrated through simulation databases: deciding whether a prediction should be trusted, choosing between models, and determining why a classifier should not be trusted.

Keywords: structural health monitoring, machine learning, convolutional neural network, explainable models, drive-by methods

1. Introduction

In recent years, with the rapid development of artificial intelligence, ML-based SHM methods have sharply become mainstream. They typically rely on large datasets and models such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN) [1]. Current research mainly focuses on deep learning models, whose high accuracy and efficiency have made them highly sought after, despite their lack of physical interpretabilities; most models are black boxes [2]. They have inevitably become integrated into urban infrastructure monitoring, forming an indispensable part of smart cities.

Given the significant responsibilities associated with SHM, understanding the fundamental rationale behind it is crucial. In some cases, even experienced experts have difficulties finding evidence related to structural integrity within complex structural signals. Thus, solely relying on these black-box SHM systems carries inherent risks, especially in methods that have relatively large uncertainty. For instance, the indirect SHM method using passing vehicles is an emerging technology aimed at providing a rapid, cost-effective solution for bridge monitoring.

The drive-by method, which necessitates just a few sensors on vehicles, captures bridge characteristics in the vehicle's response via vehicle-bridge interaction (VBI), providing a low-cost solution for monitoring fleets of bridges [3]. However, its practical implementation often introduces more uncertainty than direct methods. These uncertainties primarily arise from road roughness and vehicle dynamics, which may mask bridge characteristic components in vehicle vibrations, thereby affecting SHM results [4–6]. Over the past two decades, many efforts have been made to minimize these uncertainties, which are key in traditional modal parameter-based methods [7].

Recently, a surge of research into the application of ML in the drive-by method has emerged, claiming promising results [8–14]. However, due to the significant uncertainties associated with the drive-by method, decision makers may not trust it as much as the direct method. So, are these results truly reliable, and how should asset managers decide whether to trust them? These drive the authors to propose explanatory approaches to ML-based SHM. Some methods in computer science like SHAP (SHapley Additive exPlanation) often require substantial computational power [15]. Sometimes, engineers or researchers need a quick method to preliminarily judge whether to choose a model.

This paper proposes an efficient ML model explanatory algorithm named the noise perturbation-based feature importance calculation method. The algorithm provides insights about the model by adding perturbation to features to determine their contribution to the results. The algorithm is demonstrated on a vehicle-based SHM case using a 1D-Convolutional Neural Network (CNN), which is considered one of the most effective neural networks for vibrational signals [16]. A dataset is established through numerical simulation, and a designed CNN model is trained and validated on it. The proposed algorithm will first compare results with SHAP, then be used to interpret the ML model, deciding whether to trust its predictions and choose between models.

2. Explanatory algorithm

1.1 SHAP approach

First, we introduce the SHAP method, a classical approach for explaining ML models, which is used in this paper to compare with the proposed rapid explanatory algorithm. The SHAP methodology builds upon the cooperative game theory concept of Shapley values to assign an importance value to each feature, reflecting its contribution to the prediction made by a ML model [15]. This approach ensures a fair distribution of the 'payout' among features, akin to how players are rewarded in a coalition game. By leveraging SHAP, researchers and practitioners can gain transparent insights into predictive models, facilitating better understanding and trust in ML-driven decisions.

SHAP values are derived by computing the average change in the prediction outcome when a feature value is introduced into a model, compared to predictions without the feature. The fundamental formula is shown in Equation (1). In the equation, F is the set of all features, S is a subset of features excluding feature i, and f is the prediction model. This ensures each feature's contribution is fairly distributed according to its marginal effect

on the model's output. It is known from the formula that the factorial operation involved in calculating SHAP values requires significant computational resources, which is not conducive to scenarios that demand quick evaluation or selection of models.

$$SHAP_i = \sum_{S \subseteq F\{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \quad (1)$$

1.2 Proposed explanatory algorithm

The algorithm calculates the importance value of each feature through perturbation. **Algorithm 1** details the proposed method: The algorithm commences with an initialization of an empty set $\boldsymbol{Imp}$. It iterates over each sample $\boldsymbol{x_j}$ within the set $\boldsymbol{X}$. For every sample, the algorithm initializes an empty set $\boldsymbol{Imp_j}$. Within this loop, for each feature e_k in $\boldsymbol{x_j}$, the algorithm adjusts e_k by adding Gaussian noise $\mathcal{N}(0, \mu^2)$ to yield a perturbed feature $\boldsymbol{x_j}'[e_k]$. It then computes a new prediction $\boldsymbol{y_j}'$. The algorithm extends $\boldsymbol{Imp_j}$ with the absolute difference between $\boldsymbol{y_j}$ and $\boldsymbol{y_j}'$. Upon completion of feature iterations within $\boldsymbol{x_j}$, the algorithm merges $\boldsymbol{Imp_j}$ into the main set $\boldsymbol{Imp}$. Once all samples are processed, it calculates the mean, labeled $\boldsymbol{Imp^{avg}}$; the algorithm ends by returning $\boldsymbol{Imp^{avg}}$. The computational efficiency of this algorithm is higher than SHAP and many other explanatory methods.

Algorithm 1 Noise perturbation-based feature importance calculation

Require: Model f and the set of samples $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3} \ldots]$
Require: Model prediction $\boldsymbol{Y} = f(\boldsymbol{X}) = [\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3} \ldots]$
Require: Noise level μ (0.001)
$\boldsymbol{Imp} \leftarrow []$
For sample $\boldsymbol{x_j}$ in $\boldsymbol{X}$:
    $\boldsymbol{Imp_j} \leftarrow \{\}$
    For feature e_k in $\boldsymbol{x_j}$:
        $\boldsymbol{x_j}'[e_k] \leftarrow \boldsymbol{x_j}[e_k] + \mathcal{N}(0, \mu^2)$
        $\boldsymbol{y_j}' \leftarrow f(\boldsymbol{x_j}')$
        $\boldsymbol{Imp_j} \leftarrow \boldsymbol{Imp_j} \cup (|\boldsymbol{y_j} - \boldsymbol{y_j}'|)$
    $\boldsymbol{Imp} \leftarrow \boldsymbol{Imp} \cup \boldsymbol{Imp_j}$
$\boldsymbol{Imp^{avg}} \leftarrow mean(\boldsymbol{Imp})$
Return $\boldsymbol{Imp^{avg}}$

3. VBI model and simulation

3.1 VBI model

In this study, the vehicle is modeled using the 2-DOF (Degrees of Freedom) quarter-car model. This simplification is widely adopted in the literature, for instance, by Liu et al. [17] and Lan et al. [18]. The bridge is modeled as a simply supported Euler–Bernoulli beam; each node of its finite element model consisted of two DOFs: vertical translation and rotation. The bridge model is structured into n elements, $n + 1$ nodes, and $2n$ DOFs, not accounting for the vertical constraints at the ends. It spans a length L, with a uniform flexural rigidity EI, and a mass per unit length m. Bridge damping is approximated through mass-stiffness proportional Rayleigh damping. The employed VBI model is illustrated in **Fig. 1**. The system's dynamics are governed by the coupling equations:

$$[M_v]\{\ddot{y}_v\} + [C_v]\{\dot{y}_v\} + [K_v]\{y_v\} = \{F_{cv}\} \quad (2)$$

$$[M_b]\{\ddot{y}_b\} + [C_b]\{\dot{y}_b\} + [K_b]\{y_b\} = \{F_{cb}\} \quad (3)$$

where the matrices $[M_v]$, $[C_v]$, and $[K_v]$ represent the vehicle's mass, damping, and stiffness, whereas $[M_b]$, $[C_b]$, and $[K_b]$ stand for the bridge's equivalent matrices. The displacement vectors of the vehicle and bridge are denoted by $\{y_v\}$ and $\{y_b\}$, respectively, with $\{F_{cv}\}$ and $\{F_{cb}\}$ symbolizing the dynamic interaction forces between them.

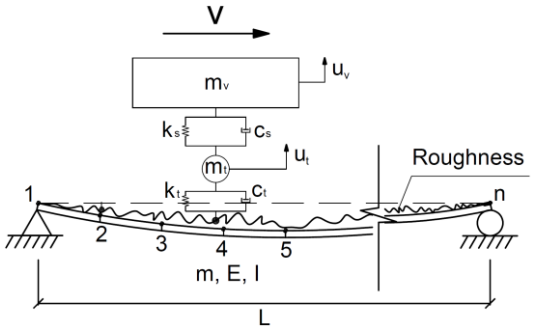


Fig. 1. VBI model.

The subsystem matrices and response vector for the vehicle model are as follows, where the body and axle masses are denoted by m_v and m_t, the suspension and tire damping by c_s and c_t, and the suspension and tire stiffness by k_s and k_t, respectively. The vertical displacements of the vehicle body and axle are denoted u_v and u_t, respectively. Dynamic responses of the vehicle through the VBI process are determined using the Newmark-Beta method, with the method's parameters β and γ chosen as 0.25 and 0.5,

$$[M_v] = \begin{bmatrix} m_v & \\ & m_t \end{bmatrix} \quad (4)$$

$$[C_v] = \begin{bmatrix} c_s & -c_s \\ -c_s & c_s + c_t \end{bmatrix} \quad (5)$$

$$[K_v] = \begin{bmatrix} k_s & -k_s \\ -k_s & k_s + k_t \end{bmatrix} \quad (6)$$

$$\{y_v\} = [u_v \quad u_t]^T \quad (7)$$

In this study, road roughness is generated according to ISO 8608 [19], with a roughness coefficient $G_d(n_{s,0}) = 16 \times 10^{-6}\, m^3$ (Class A). Notably, this research applied a 10% noise level, diverging from the commonly adopted high noise level of 5% [20]. This is to demonstrate the high accuracy of ML methods in vehicle-based SHM applications.

3.2 Simulation and dataset

In this study, the bridge parameters are as follows: mass per unit length $m = 2400\, kg/m$, bending stiffness $EI = 5.5 \times 10^9\, N \cdot m^2$, and length $L = 25\, m$. It is divided into 10 elements ($n = 10$). Generally, bridge damage can be considered as stiffness loss, representing bridge damage like cracks and delamination. The vehicle parameters are: $m_v = 1.28 \times 10^4\, kg$, $m_t = 1.0 \times 10^3\, kg$, $c_s = 1.0 \times 10^3\, N \cdot s/m$, $c_t = 0$, $k_s = 4.0 \times 10^5\, N/m$, $k_t = 3 \times 10^5\, N/m$, and $v = 8\, m/s$. The sampling rate is 1000 Hz (or a time step of 0.001 s).

Five damage cases (DC 0–DC 4) are considered, with stiffness loss of 0% (healthy), 2%, 4%, 6%, and 8% for the fifth element ($n = 5$). They represent minor structural damage. For the different DCs, each consists of 200 samples, forming a database. They are then randomly divided into training and testing sets at a 9:1 ratio.

3. ML-based SHM and interpretation

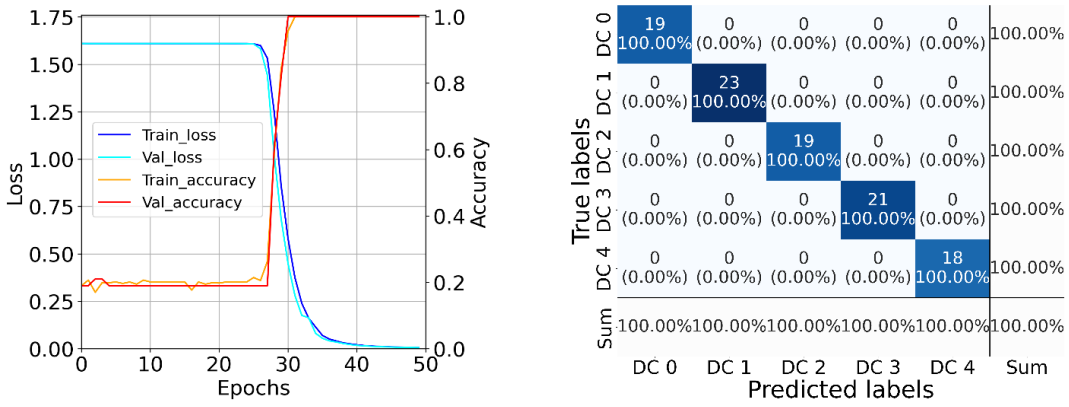
3.1 CNN model and performance

In this study, the CNN model can be trained in a supervised manner since the dataset is labeled. The model's input is the frequency domain data corresponding to 5 seconds of acceleration data measured on the vehicle's axle. This might be more intuitive for humans and beneficial to the subsequent model explanation. After some trials by the authors, **Table 1** shows the 1D-CNN architecture that performed well on the dataset of this study. A key feature of this architecture is the adoption of LeakyReLU as the activation function for its convolutional layers, replacing the traditional ReLU. It aims to mitigate the problems of dead neurons and vanishing gradients [21].

Table 1. CNN configuration

Layer	Output shape	Parameter	Activation
Conv1d	2500 × 64	Kernel number: 64; Kernel size:10; Stride: 1	LeakyReLU
Max pooling	1250 × 64	Kernel: 2; Stride: 2	None
Conv1d	1250 × 128	Kernel number: 128; Kernel size:10; Stride: 1	LeakyReLU
Max pooling	625 × 128	Kernel: 2; Stride: 2	None
Conv1d	625 × 256	Kernel number: 256; Kernel size:10; Stride: 1	LeakyReLU
Max pooling	312 × 256	Kernel: 2; Stride: 2	None
Conv1d	312 × 512	Kernel number: 512; Kernel size:10; Stride: 1	LeakyReLU
Max pooling	156 × 512	Kernel: 2; Stride: 2	None
Flatten	79872	None	None
Dense	100	None	LeakyReLU
Dense	5	None	Softmax

Model training was performed in a Python 3.11 environment. In addition to the above architecture, the other hyperparameters are: the batch size is set to 32, Adam is used as the optimizer, the learning rate is 1×10^{-4}, and the loss function used is "cross-entropy loss", and the training epochs is 50. The employed workstation at Aalto University is equipped with Intel Core i9-11900 CPUs and 32 GB of RAM. The loss and accuracy during the model training and testing phases are shown in **Fig. 2a**. Remarkably, in terms of minor damage detection, CNN achieved an exceptional test accuracy of 100%. The accuracy stabilizes around 30-epoch, and the trend indicates that there are no significant overfitting issues. These results are based on numerical simulations. However, drawing from the authors' previous works, ML methods have been shown to achieve accuracy above 85% for detecting structural changes as subtle as 2% in the experiments [10,11,13]. These demonstrate the advantages of high accuracy in applying ML methods to SHM.

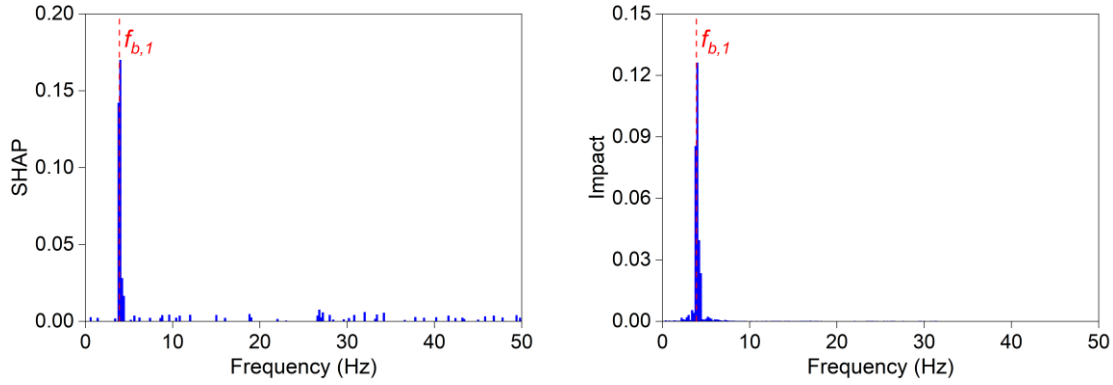


(a) (b)

Fig. 2. Selections of PFs: (a) training history, (b) confusion matrix.

3.2 Interpretation on the model

First, we will use SHAP to calculate the importance of each (frequency) feature. On the above CNN operating platform, based on 100 samples, the SHAP execution time is 4801.9 seconds. **Fig. 3a** displays the feature contributions based on SHAP. The model mainly focuses on frequencies from 3.8Hz to 4.0Hz, where the fundamental frequency of the bridge in this study is 3.8Hz. This is consistent with people's intuitive of using bridge modal characteristics to judge bridge status, in other words, the model can be trusted. Meanwhile, the feature contributions calculated using the proposed algorithm are shown in **Fig. 3b**. The algorithm results are similar to the SHAP results, i.e., focusing on features near the fundamental frequency. However, the time taken using the proposed algorithm is only 972.8 seconds, which is 20.3% of SHAP's time. Its efficiency is significantly higher than SHAP's, which is an advantage. It can quickly provide a basic understanding of the model.



(a) (b)

Fig. 3. Feature contribution: (a) SHAP, (b) proposed method.

3.3 Different models

Table 1 and **Table 2** display the configurations of Models 2 and 3 used for comparison. They are simplifications of the above CNN model to different degrees, and after training through 50 epochs, their accuracies are 46% and 21%, respectively. From the results of Model-2 shown in **Fig. 4a**, it can be observed that this model primarily focuses on features near the fundamental frequency, which is reasonable, but their importance (impact) is significantly smaller than that of the previous model (1.9×10^{-5} vs 1.3×10^{-1}). This results in a model that is much less sensitive to damage, especially minor damage, than the models described above. The detection accuracy is only 46%, which is better than random guessing (20%) but obviously unsatisfactory. The results from **Fig. 4b** indicate that, although features near the fundamental frequency are given slightly high attention, Model-3's focus is evidently dispersed, resulting in an accuracy of only 21%, close to random guessing. Hence, these models cannot be chosen or trusted. In fact, some models, despite their high accuracy, focus on features contrary to human intuition, raising further questions about their reliability. Due to space limitations, this paper does not provide detailed examples. This work shows that the interpretability of ML applications should be a focus of attention. It's noteworthy that in the simulations of this paper, the CNN can accurately capture features. In engineering practice, situations will be much more complex, and feature extraction will also be more complicated. Explanatory methods are particularly important in these cases to ensure they capture reasonable features, demanding further exploration.

Table 2. Model-2

Layer	Output shape	Parameter	Activation
Conv1d	2500 × 64	Kernel number: 64; Kernel size:10; Stride: 1	ReLU
Max pooling	1250 × 64	Kernel: 2; Stride: 2	None
Conv1d	1250 × 128	Kernel number: 128; Kernel size:10; Stride: 1	ReLU
Max pooling	625 × 128	Kernel: 2; Stride: 2	None
Flatten	80000	None	None
Dense	100	None	ReLU
Dense	5	None	Softmax

Table 3. Model-3

Layer	Output shape	Parameter	Activation
Conv1d	2500 × 64	Kernel number: 64; Kernel size:10; Stride: 1	ReLU
Max pooling	1250 × 64	Kernel: 2; Stride: 2	None
Flatten	80000	None	None
Dense	3	None	ReLU
Dense	5	None	Softmax

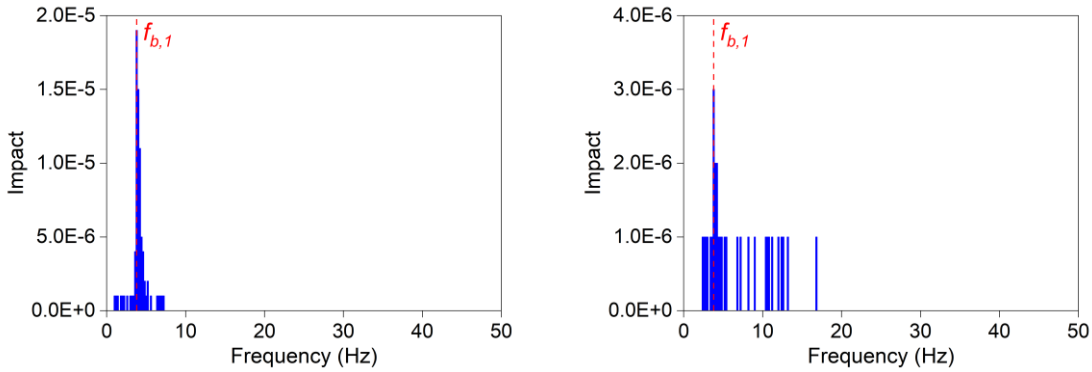


Fig. 4. Feature contribution calculated using the proposed algorithm: (a) model-2, (b) model-3.

4. Conclusion

This paper argues that trust is crucial for the application of ML methods in indirect SHM. A noise perturbation-based feature importance calculation method is proposed to quickly provide insights into the model. It was demonstrated using a CNN on a simulated drive-by SHM database. Based on the results, the following conclusions can be drawn:

1. The proposed algorithm can provide results similar to SHAP with only 20.3% of the computational time required by SHAP; it can quickly provide a basic understanding of the model.
2. The CNN model focuses on features near the bridge's fundamental frequency and detects bridge damage with 100% accuracy, making it a trustworthy model.
3. Some models give insufficient attention to features near the bridge's fundamental frequency or have dispersed attention, leading to poor performance; these are not trustworthy models.

They provide valuable information to decision-makers. Future research will further explore the interpretability of other models and SHM frameworks.

Acknowledgement

This research is sponsored by the Jane and Aatos Erkko Foundation in Finland (Grant No. 210018).

References

[1] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, S. Nagarajaiah, Review of Bridge Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection, Journal of Structural Engineering 146 (2020) 04020073.
[2] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, 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 (2021) 107077.
[3] A. Malekjafarian, R. Corbally, W. Gong, A review of mobile sensing of bridges using moving vehicles: Progress to date, challenges and future trends, Structures 44 (2022) 1466–1489.
[4] Y. Lan, W. Lin, Y. Zhang, Bridge frequency identification using vibration responses from sensors on a passing vehicle, in: Bridge Safety, Maintenance, Management, Life-Cycle, Resilience and Sustainability, CRC Press, 2022.
[5] Y. Lan, W. Lin, Y. Zhang, Bridge Frequency Identification Using Multiple Sensor Responses of an Ordinary Vehicle, International Journal of Structural Stability and Dynamics 23 (2023) 2350056.
[6] Y. Lan, Z. Li, K. Koski, L. Fülöp, T. Tirkkonen, W. Lin, Bridge frequency identification in city bus monitoring: A coherence-PPI algorithm, Engineering Structures 296 (2023) 116913.
[7] Z.L. Wang, J.P. Yang, K. Shi, H. Xu, F.Q. Qiu, Y.B. Yang, Recent Advances in Researches on Vehicle Scanning Method for Bridges, International Journal of Structural Stability and Dynamics (2022) 2230005.
[8] J. Liu, S. Chen, M. Bergés, J. Bielak, J.H. Garrett, J. Kovačević, H.Y. Noh, Diagnosis algorithms for indirect structural health monitoring of a bridge model via dimensionality reduction, Mechanical Systems and Signal Processing 136 (2020) 106454.
[9] M.Z. Sarwar, D. Cantero, Deep autoencoder architecture for bridge damage assessment using responses from several vehicles, Engineering Structures 246 (2021) 113064.
[10] Y. Lan, Y. Zhang, W. Lin, Diagnosis algorithms for indirect bridge health monitoring via an optimized AdaBoost-linear SVM, Engineering Structures 275 (2023) 115239.
[11] Y. Lan, Z. Li, W. Lin, A Time-Domain Signal Processing Algorithm for Data-Driven Drive-by Inspection Methods: An Experimental Study, Materials 16 (2023) 2624.
[12] Z. Li, Y. Lan, W. Lin, Investigation of Frequency-Domain Dimension Reduction for A2M-Based Bridge Damage Detection Using Accelerations of Moving Vehicles, Materials 16 (2023) 1872.
[13] Y. Lan, Z. Li, Y. Zhang, W. Lin, Small-scale damage detection of bridges using machine learning techniques and drive-by inspection methods, in: Life-Cycle of Structures and Infrastructure Systems, CRC Press, 2023.
[14] R. Corbally, A. Malekjafarian, A deep-learning framework for classifying the type, location, and severity of bridge damage using drive-by measurements, Computer-Aided Civil and Infrastructure Engineering 00 (2023) 1–20.
[15] S. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, (2017).
[16] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing 151 (2021) 107398.
[17] C. Liu, Y. Zhu, H. Ye, Bridge frequency identification based on relative displacement of axle and contact point using tire pressure monitoring, Mechanical Systems and Signal Processing 183 (2023) 109613.
[18] Y. Lan, Z. Li, W. Lin, Physics-guided diagnosis framework for bridge health monitoring using raw vehicle accelerations, Mechanical Systems and Signal Processing 206 (2024) 110899.
[19] P. Múčka, Simulated Road Profiles According to ISO 8608 in Vibration Analysis, Journal of Testing and Evaluation 46 (2018) 20160265.
[20] Z. Li, Y. Lan, W. Lin, Indirect damage detection for bridges using sensing and temporarily parked vehicles, Engineering Structures 291 (2023) 116459.
[21] B. Xu, N. Wang, T. Chen, M. Li, Empirical Evaluation of Rectified Activations in Convolutional Network, (2015).