






“What You Need is Data!”: Physics-Guided TimeGAN for Data Expansion in Bridge SHM

Yifu Lan^(✉) , Zhenkun Li , and Weiwei Lin 

Department of Civil Engineering, Aalto University, Espoo, Finland
yifu.lan@aalto.fi

Abstract. Traditional Structural Health Monitoring (SHM) technologies have not yet been widely adopted by infrastructure asset managers. However, with the recent advancements in information technology and artificial intelligence (AI), modern SHM has brought greater practical potential. AI-based SHM often relies on big data, but in practice, data availability is a problem, especially for some difficult-to-obtain cases. Theoretically, Generative Adversarial Networks (GANs) can augment data and significantly expand databases. On the other hand, some argue that GAN models can be challenging to train and that the amount of data required to train a GAN might be sufficient to train a diagnostic model. More importantly, the data generated by GANs may not truly capture the underlying physical characteristics. To address these, this paper proposes to use physical laws to guide GANs, with TimeGAN adopted as the base model due to its strong performance with time-series data. In this study, the proposed physics-guided TimeGAN (PyTiGAN) is used for data expansion in bridge SHM under the excitation of traffic events. The results demonstrate the effectiveness of the proposed method in expanding bridge SHM datasets from multiple dimensions.

Keywords: Structural health monitoring · Data expansion · AI · TimeGAN · Physics-guided GANs · Vehicle-bridge interaction

1 Introduction

The maintenance and management of infrastructure assets, such as bridges, are crucial for ensuring public safety and sustaining economic activities. Over the years, Structural Health Monitoring (SHM) has emerged as a vital means for assessing the condition of infrastructure and detecting early signs of deterioration [1]. However, traditional SHM technologies have not yet seen widespread adoption by asset managers. On-site surveys are typically conducted by experienced experts through visual monitoring. However, these methods have several limitations, including high labor costs, operational disruptions, etc. [2]. Though sensor-based SHM offers an alternative approach, it comes with its own challenges, such as the high cost of sensor installation and maintenance, as well as the complexity of data processing—where raw signals, such as vibrations, must be converted into human-readable information [3]. Furthermore, because the equipment is permanently attached to the infrastructure as a customized SHM system, it could be

challenging to transfer one monitoring framework to others [4]. With the rapid advancements in information technology and artificial intelligence (AI), modern SHM systems have started to demonstrate significant potential for overcoming these barriers, paving the way for more practical and scalable solutions [5–7].

One of the key advantages of modern SHM is its ability to leverage AI-powered algorithms for real-time data analysis and decision-making. AI-based SHM systems often rely on big data collected from vehicles, drones, and other sensing equipments [8]. However, in practice, data availability remains a significant challenge, particularly for scenarios that are difficult or expensive to monitor consistently. For example, data for extreme events, such as bridge responses under heavy traffic loads or unusual environmental conditions, are often sparse [9]. The scarcity of such data can limit the ability of AI algorithms to learn robust patterns, thereby reducing the reliability of diagnostic models. Addressing this issue has become a critical research focus in SHM.

In theory, Generative Adversarial Networks (GANs) offer a promising approach to augmenting datasets by generating synthetic data that mimics real-world observations. GANs have gained considerable attention across multiple domains, including image processing, speech synthesis, and medical diagnosis, for their ability to generate realistic and high-dimensional data [10]. By applying GANs to SHM, researchers can potentially expand existing datasets, enabling the development of more robust and accurate diagnostic models. However, despite their promise, GANs face several limitations [11]. First, GANs are notoriously difficult to train, requiring careful hyperparameter tuning and significant computational resources. Second, the amount of data required to train a GAN model can often rival or exceed the data needed to directly train a diagnostic model. Finally, and most critically, the synthetic data generated by GANs may fail to capture the underlying physical characteristics of the monitored system, leading to potential inaccuracies in downstream applications.

To address these challenges, the authors are interested in integrating physics-domain knowledge into GAN architectures. Specifically, incorporating physical laws into the data generation process can guide the synthetic data to better mimic the real-world responses of infrastructure systems. In this context, this paper proposes a novel physics-guided GAN (PyTiGAN) framework for SHM, with a focus on time-series data. The TimeGAN model, known for its strong performance in handling temporal dependencies, is adopted as the base architecture [12]. By embedding physical principles into the training process, the proposed PyTiGAN enhances the quality and reliability of the generated data. The proposed framework will be validated within a bridge SHM system under the excitation of traffic events. This enriched dataset can enable the development of more accurate diagnostic models, ultimately supporting more informed decision-making for infrastructure maintenance and management.

2 Proposed Algorithm

2.1 Overview of TimeGAN

TimeGAN is a generative model specifically designed for time-series data, combining adversarial training with supervised learning to enhance temporal dependencies in generated sequences [12]. Unlike traditional GANs that may struggle to maintain coherence

in sequential data, TimeGAN introduces an embedding function that maps both real and synthetic data into a shared latent space, ensuring consistency in temporal structures. The TimeGAN framework consists of four key components:

1. **Embedding Network:** Transforms raw time-series data into a latent space representation to capture essential temporal dependencies.
2. **Recovery Network:** Reconstructs time-series data from the latent space, ensuring fidelity to the original input.
3. **Generator & Adversarial Networks:** The generator creates synthetic sequences based on sampled latent vectors, while the adversarial discriminator differentiates real from synthetic data.
4. **Supervised Loss:** Enforces consistency between latent representations of real and generated data, aiding in training stability and improving synthetic data quality.

In TimeGAN, the most critical formula is the combined objective function, which integrates adversarial training, supervised learning, and embedding consistency. This ensures the generated time-series data preserves temporal dependencies while maintaining realism. The key formula is:

$$\min_G \max_D L_{adv} + \lambda L_{sup} + \gamma L_{rec} \quad (1)$$

where: L_{adv} is the adversarial loss, ensuring the generated data is indistinguishable from real data. L_{sup} is the supervised loss, enforcing consistency in the latent space between real and generated sequences. L_{rec} is the reconstruction loss, ensuring the generated sequences can be mapped back to meaningful real-world data. λ and γ are weighting factors that balance these terms.

The training process follows a two-stage approach:

1. **Pretraining with Supervised Learning** – The embedding and recovery networks are trained first, ensuring that the model learns accurate latent representations of time-series data.
2. **Adversarial Training** – The generator and discriminator networks are then trained iteratively to produce realistic time-series sequences that align with real data distributions.

By this way, TimeGAN effectively generates high-quality synthetic data while preserving critical temporal structures. However, the generated sequences may not always align with real-world physical behaviors, particularly in SHM applications.

2.2 PyTiGAN

To address it, a physics-guided approach is proposed, incorporating domain-specific physical knowledge into the data generation process. This ensures that synthetic data not only preserves statistical patterns but also complies with the fundamental physical laws. In this study, the bridge SHM scenario based on vehicle-induced excitation involves a vehicle crossing a simply supported beam bridge. It should be ensured that the generated

bridge response adheres to the governing equations of motion for a simply supported beam under vehicular excitations (Euler-Bernoulli beam model) [13]:

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

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

where $[M_v]$, $[C_v]$, and $[K_v]$ are the mass, damping, and stiffness matrices of the vehicle, respectively; $[M_b]$, $[C_b]$ and $[K_b]$ represent the mass, damping, and stiffness matrices of the bridge model, respectively. In the equations, $\{u_v\}$ is the displacement vector of the vehicle and $\{u_b\}$ is the nodal displacement of the bridge system. $\{F_{cv}\}$ and $\{F_{cb}\}$ stand for the time-varying interaction forces on the vehicle and the bridge, respectively.

We aim to obtain the physical solution. For a 2-DOF vehicle model, its subsystem matrices and response vector are as follows:

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

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

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

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

where m_v and m_t are the body and axle masses; c_s and c_t stand for the suspension and tire damping; k_s and k_t denote the suspension and tire stiffnesses; y_v and y_t are the vertical displacements of the vehicle body and the axle.

The bridge is considered as a simply supported Euler-Bernoulli beam, with two DOFs at each node (vertical translation and rotation). It has a length of L , a uniform flexural rigidity of EI , and a mass per unit length of m . Mass-stiffness proportional Rayleigh damping is used to simulate the bridge's damping. These are illustrated in Fig. 1.

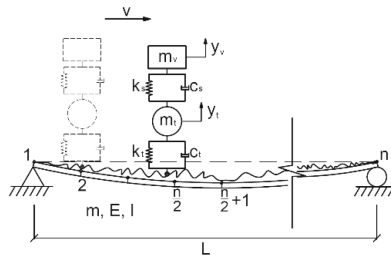


Fig. 1. Vehicle-bridge interaction model.

The process can be solved by employing the Newmark-Beta method to acquire the bridge's dynamic responses ($\beta = 0.25$, $\gamma = 0.5$). Based on the experimental beam

and vehicle model used in our studies, the parameters can be selected as follows: $m = 1250 \text{ kg/m}$, $EI = 2.6 \times 10^{12} \text{ N} \cdot \text{m}^2$, $L = 45 \text{ m}$, $m_v = 1.6 \times 10^4 \text{ kg}$, $m_t = 7 \times 10^3 \text{ kg}$, $c_s = 1.0 \times 10^4 \text{ N} \cdot \text{s/m}$, $c_t = 0$, $k_s = 4 \times 10^5 \text{ N/m}$, $k_t = 1 \times 10^4 \text{ N/m}$, and $v = 9 \text{ m/s}$. The road surface roughness can be simulated according to ISO 8608 [14]. Class-A level roughness is selected to represent the road condition of the experiment in this paper. For more details about VBI, please refer to the author’s previous work [15–17].

The physical solution obtained from the VBI process, denoted as Phy , will be incorporated into the training process. Assuming a processor consists of a function $\Gamma(\cdot)$, it maps the input physical solution Phy to a processed representation Phy' (see Eq. 8). It aims to compress the target time series into a structured latent representation.

$$Phy' = \Gamma(Phy) \quad (8)$$

A new loss term is constructed as the mean squared error (MSE) between the synthetic embedding E_z , generated by the generator’s embedder from simulated noise z , and the processed target time series Phy' , obtained through the processor.

Mathematically, it can be expressed as:

$$L_{Phy} = \frac{1}{NTF} \sum_{n=1}^N \sum_{t=1}^T \sum_{f=1}^F \left(E_z^{(n,t,f)} - Phy'^{(n,t,f)} \right)^2 \quad (9)$$

where N is the number of samples, T is the number of timesteps, and F is the number of features. This loss term ensures that the generated embeddings align closely with the known nature of the target time series and is incorporated into the generator loss function with a weight factor α . Phy' influences the loss function and optimization, encouraging the generator to align its embeddings E_z with the expected target patterns, ultimately guiding the synthetic time series toward more physically realistic representations.

3 Experiments

To evaluate the effectiveness of PyTiGAN, we will compare it with TimeGAN using a dataset collected from laboratory experiments. Additionally, a linear SVM model will be trained to classify healthy and damaged datasets; they can assess the effectiveness of the data in some aspects.

To ensure a comprehensive evaluation, we assess the quality of generated data based on three key criteria: (1) Diversity – the generated samples should exhibit a broad distribution that effectively captures the variability of real data; (2) fidelity – the synthetic data should be indistinguishable from real data in terms of statistical and temporal properties. (3) usefulness – when used for SHM tasks, synthetic data should perform comparably to real data.

3.1 Experimental Setup

The experimental setup involves a HEA400 beam model representing the bridge, with key physical properties including an elastic modulus of $E = 199 \text{ GPa}$; density $\rho = 7.85 \times 10^3 \text{ kg/m}^3$; length $L = 4.4 \text{ m}$; section area $A = 15898 \text{ mm}^2$, and moment of

inertia $I = 8.564 \times 10^7 \text{ mm}^4$. The bridge's sectional dimensions are shown in Fig. 2, and the layout of the bridge model, which includes an acceleration ramp, deceleration ramp, and a wire system for vehicle guidance, is also illustrated. For the purposes of the experiment, artificial damage is introduced by adding mass at the mid-span of the bridge, an accelerometer is mounted on the mid-span to collect vibration data as the vehicle crosses the bridge. The data is captured using a PC-driven data acquisition system with a sampling rate of 1 kHz. The bridge model arranged, damage pattern, and sensor installation in the laboratory are shown in Fig. 3a, b, and c.

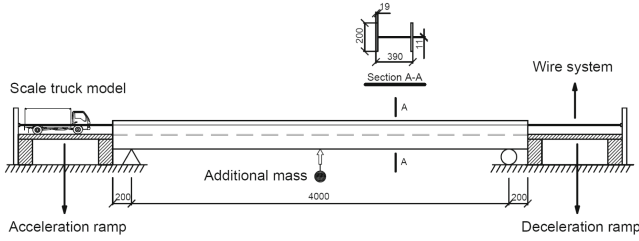
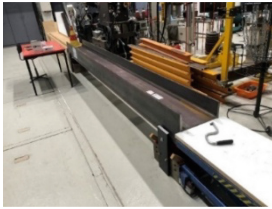


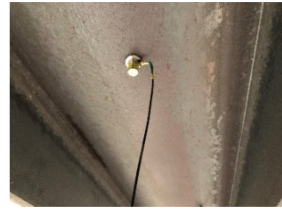
Fig. 2. Bridge model (unit: mm).



(a) Steel beam



(b) Additional mass



(c) Sensor

Fig. 3. Beam model setup.

The vehicle model used is a Tamiya Mercedes-Benz 1850 L (1/14 scale, 568 mm \times 202 mm) designed to accurately simulate a full-sized truck, except for its weight (see Fig. 4a). The self-weight of the vehicle is 4.05 kg, with an additional 5-kg mass inside the body, giving it a total weight of 9.05 kg. The vehicle's speed ranges from 0.825 m/s to 1.025 m/s across runs, with an average of 0.93 m/s (see Fig. 4b). The vehicle is repeatedly maneuvered across the bridge to collect data. In total, one healthy state and four damaged cases are used (see Table 1). The healthy state has 200 vehicle runs, while each damaged state also includes 200 runs.

3.2 Results

The PyTiGAN is configured with the following parameters: The model architecture comprises 4 layers, a hidden dimension of 32, and a time step size of 40. λ and γ are selected to 0.01 and 5, respectively. The learning rate is 0.0001, optimized using a batch size of 64. The Physics loss weight α is set to 0.5. The length ratio of training to

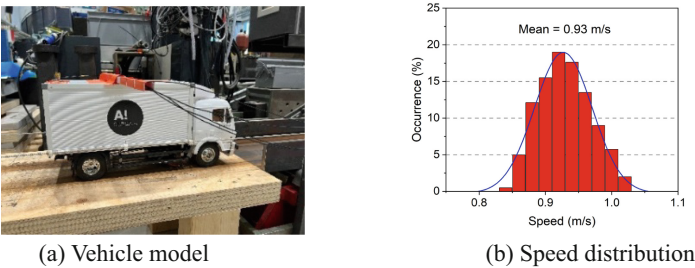


Fig. 4. Vehicle model setup.

Table 1. Damaged cases.

Case No	Location	Weight	Runs	Case No	Location	Weight	Runs
1	Mid-span	4 kg (0.8%)	200	2	Mid-span	6 kg (1.2%)	200
3	Mid-span	8 kg (1.6%)	200	4	Mid-span	10 kg (2%)	200

testing time series is 4:1. In general, these parameters need to be set very carefully. The amount and quality of data, as well as the choice of parameters can significantly affect the performance of GANs, but this is somewhat beyond the scope of this article.

The training history in Fig. 5 shows a consistent decline in all loss components, indicating successful model convergence. The autoencoder loss rapidly decreases, demonstrating effective latent representation learning. The generator loss drops significantly in the early stages, stabilizing later; the discriminator loss follows a similar pattern, reflecting the adversarial balance between generator and discriminator. The physics loss steadily decreases throughout the training, highlighting the model’s success in generating data that adheres to physical constraints.

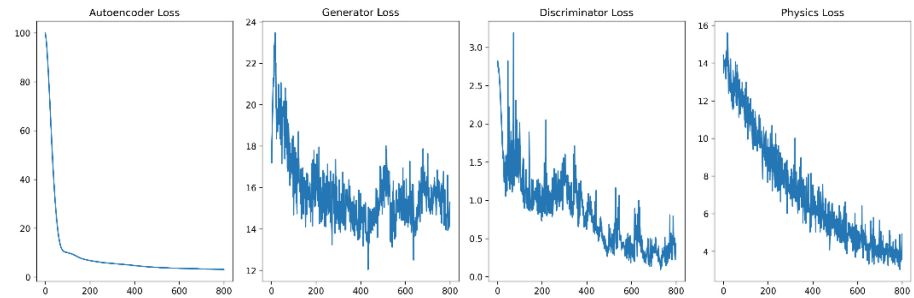


Fig. 5. Training history.

The results in Fig. 6 compare the actual data with the outputs of TimeGAN and PyTiGAN. It can be found that the PyTiGAN generated data follows the actual patterns

more closely, successfully capturing the characteristics associated with the VBI process. In contrast, TimeGAN produces less structured and more inconsistent results, failing to replicate the characteristic temporal patterns. This comparison highlights PyTiGAN’s performance in preserving the dynamic behavior of the system.

Some might argue that the observed differences are primarily due to suboptimal parameter tuning, which is indeed a crucial aspect of GAN training. However, these results demonstrate that PyTiGAN can guide the generated responses closer to actual physical behavior to some extent, even without meticulously optimized parameters. A more detailed comparison under optimized settings is left for future research (due to space constraints in this paper).

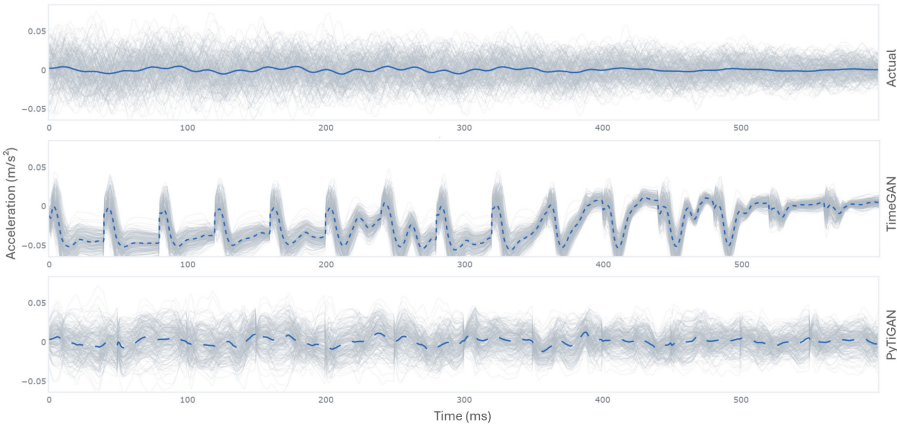


Fig. 6. Result comparison.

Figure 7 presents the distribution of original and synthetic data using PCA [18] and t-SNE [19] visualizations. In the visualizations, the synthetic data (orange) has similar patterns to the original data (blue) but does not completely overlap (especially in s-SNE). This indicates that the generated samples successfully capture the overall distribution of the training set while retaining sufficient diversity, which is essential for the model’s generalization capability.

A linear SVM with a regularization parameter of $C = 1.0$ is employed to perform binary classification between healthy and damaged states. The model’s performance across the four damage cases is presented in Table 2. Conservatively, we deliberately limit the amount of synthetic data to not exceed the original data. PyTiGAN achieves the highest accuracy in all cases, with improvements ranging from 1.0% to 2.7% over the baseline. While GAN and TimeGAN exhibit decreased accuracy, this can be attributed to insufficient parameter adjustment, as discussed above. GANs are typically challenging to train, but in any case, the introduction of PyTiGAN improves the generalizability of GANs from this perspective.

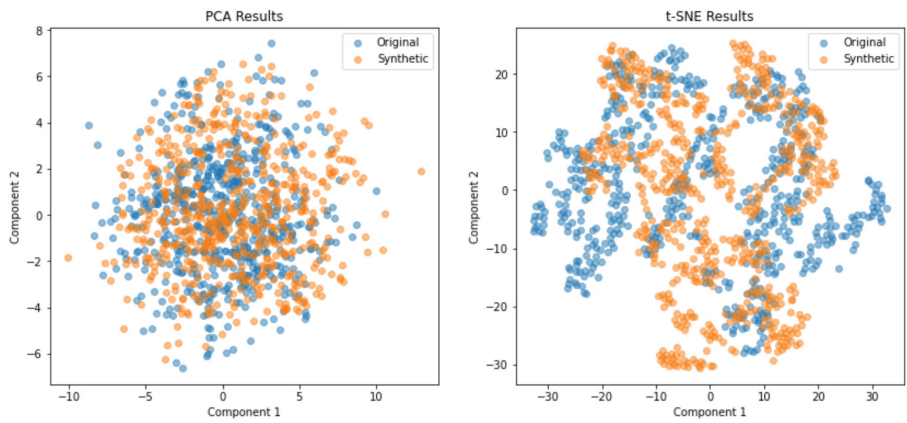


Fig. 7. PCA and t-SNE visualizations.

Table 2. Model performance.

Method	Case 1 (0.8%)	Case 2 (1.2%)	Case 3 (1.6%)	Case 4 (2.0%)
Original	76.3%	81.3%	85.0%	90.0%
GAN	73.0%	79.0%	82.0%	85.0%
TimeGAN	75.0%	80.0%	83.0%	87.0%
PyTiGAN	79.0%	83.0%	86.0%	92.0%

4 Conclusion

In this paper, a physics-guided TimeGAN (PyTiGAN) for time-series data generation in SHM applications is introduced. By integrating physical constraints into the TimeGAN architecture, the proposed model can generate statistically consistent and physically meaningful data, as validated through experiments on the VBI data. Based on the results, the following conclusions can be drawn:

- (1) The addition of physics loss terms effectively guides TimeGAN to generate physically consistent data without requiring stringent parameter tuning or complex training.
- (2) Compared to TimeGAN, PyTiGAN generates synthetic data that aligns visually with the original time-domain data, as demonstrated through PCA and t-SNE visualizations, while maintaining sufficient diversity.
- (3) The inclusion of PyTiGAN-generated data enhances the damage detection accuracy of machine learning models, emphasizing its usefulness in SHM applications.

Future research will discuss the impact of hyper-parameters on PyTiGAN’s performance and explore its effectiveness across diverse datasets.

Acknowledgement. This research is sponsored by the Jane and Aatos Erkko Foundation in Finland (Decision number: 210018).

References

1. Li H, Ou J (2011) Structural health monitoring: from sensing technology stepping to health diagnosis. *Procedia Eng.* 14:753–760
2. Dong CZ, Catbas FN (2021) A review of computer vision-based structural health monitoring at local and global levels. *Struct. Health Monit.* 20(2):692–743
3. Abdulkarem M, Samsudin K, Rokhani FZA, Rasid MF (2020) Wireless sensor network for structural health monitoring: a contemporary review of technologies, challenges, and future direction. *Struct. Health Monit.* 19(3):693–735. <https://doi.org/10.1177/1475921719854528>
4. Malekjafarian A, McGettrick PJ, OBrien EJ (2015) A Review of Indirect Bridge Monitoring Using Passing Vehicles. *Shock Vib.* 2015:1–16. <https://doi.org/10.1155/2015/286139>
5. Lan Y, Zhang Y, Lin W (2023) Diagnosis algorithms for indirect bridge health monitoring via an optimized AdaBoost-linear SVM. *Eng. Struct.* 275:115239
6. Lan Y, Li Z, Lin W (2023) A time-domain signal processing algorithm for data-driven drive-by inspection methods: an experimental study. *Materials* 16(7):2624
7. Li Z, Lan Y, Lin W (2024) Footbridge damage detection using smartphone-recorded responses of micromobility and convolutional neural networks. *Autom. Constr.* 166:105587
8. Sun L, Shang Z, Xia Y, Bhowmick S, Nagarajaiah S (2020) Review of bridge structural health monitoring aided by big data and artificial intelligence: from condition assessment to damage detection. *J. Struct. Eng.* 146(5):04020073
9. Quqa S et al (2025) Regional-scale bridge health monitoring: survey of current methods and roadmap for future opportunities under changing climate. *Struct. Health Monit.* 24(4):2309–2337. <https://doi.org/10.1177/14759217241310525>
10. Gui, J., Sun, Z., Wen, Y., Tao, D., Ye, J.: A review on generative adversarial networks: algorithms, theory, and applications (2020)
11. Wang Z, She Q, Ward TE (2021) Generative adversarial networks in computer vision: a survey and taxonomy. *ACM Comput. Surv.* 54(2):1–38
12. Yoon, J., Jarrett, D., Schaar, M.: Time-series generative adversarial networks. *Adv. Neural Inf. Process. Syst.* (2019)
13. Lan Y, Li Z, Lin W (2024) Physics-guided diagnosis framework for bridge health monitoring using raw vehicle accelerations. *Mech. Syst. Signal Process.* 206:110899
14. Múčka P (2018) Simulated road profiles according to ISO 8608 in vibration analysis. *J. Test. Eval.* 46:20160265
15. Lan Y, Li Z, Koski K, Fülöp L, Tirkkonen T, Lin W (2023) Bridge frequency identification in city bus monitoring: a coherence-PPI algorithm. *Eng. Struct.* 296:116913
16. Li Z, Lan Y, Lin W (2023) Indirect damage detection for bridges using sensing and temporarily parked vehicles. *Eng. Struct.* 291:116459
17. Li Z, Lan Y, Lin W (2024) Indirect frequency identification of footbridges with pedestrians using the contact-point response of shared scooters. *J. Bridg. Eng.* 29(6):04024036
18. Greenacre M, Groenen PJF, Hastie T, D’Enza AI, Markos A, Tuzhilina E (2022) Principal component analysis. *Nat. Rev. Methods Primers* 2(1):1–21
19. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9(86):2579–2605