



Article

Physics-Informed Hybrid Deep Learning for Damage Detection in Metro Tunnel Structures Under Moving Train Loads

Arshed Raad Raheem ¹ and Osama Alazzawi ^{2,*}

¹ College of Law and Political Science, University of Diyala, Baqubah 32001, Diyala, Iraq; arshad.raad@uodiyala.edu.iq

² Department of Civil Engineering, College of Engineering, University of Diyala, Baqubah 32001, Diyala, Iraq

* Correspondence: osama.r.rahim@uodiyala.edu.iq

Abstract

In this study, a novel physics-informed hybrid deep learning (PI-HDL) algorithm is proposed for intelligent condition assessment of metro tunnel structures. The algorithm integrates state-of-the-art deep learning models, including auto-encoder and convolutional neural network (CNN) models. Physics-inspired constraints derived from structural dynamics have been embedded into the loss functions of these models. Via an effective learning, merging, and training methodology, the designed model reconstructs high-level features of 1D time-domain vibration response signals in an unsupervised manner. Then, these physics-consistent high-level features are fed into a supervised 2D CNN trained with a composite loss function that incorporates both data fidelity and physics-based residuals derived from the stiffness–frequency relationship and moving load characteristics. The developed methodology is evaluated using vibration signals collected numerically from a concrete metro tunnel structure under various moving train loads and damage scenarios. The proposed PI-HDL framework embeds the underlying physics of wave propagation and structural dynamics directly into the learning process. This enables accurate identification of predefined damage conditions and stiffness reduction quantification within a closed set of ten structural scenarios, even with limited training data and noisy measurements. Comparisons prove that the PI-HDL algorithm achieves superior closed-set classification performance compared to purely data-driven HDL approaches.

Keywords: physics-informed neural networks; artificial intelligence; auto-encoder; structural health monitoring; metro tunnel; moving load



Academic Editors: António Couto and Francesca Dezi

Received: 28 July 2026

Revised: 28 September 2026

Accepted: 14 October 2026

Published: 19 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In the past two decades, transportation systems such as metro tunnels have witnessed extensive investigations regarding structural damage detection and health monitoring. The aim of all structural health monitoring (SHM) studies is to identify the health status of an intended structural system [1]. Existing approaches generally fall into two families: physics-based (white-box) methods that rely on finite element models and governing differential equations and data-driven (black-box) methods that learn patterns directly from sensor data [2,3].

Physics-based methods offer interpretability and generalization but require detailed knowledge of material properties, boundary conditions, and loading scenarios that are often uncertain in real structures. In metro tunnel applications, the complexity of soil–structure interactions and moving train loads makes pure physics-based approaches computationally

expensive and often impractical for real-time monitoring. Conversely, data-driven methods, particularly deep learning approaches, can capture complex nonlinear relationships from raw sensor data but typically require massive labeled datasets and lack physical interpretability, limiting their deployment in safety-critical infrastructure applications [4,5].

Physics-informed machine learning (PIML) has recently emerged as a transformative paradigm that integrates governing physical laws into data-driven models [6,7]. Physics-informed neural networks (PINNs) embed partial differential equations (PDEs) and physical constraints directly into the loss function, enabling the network to learn solutions that satisfy both observed data and underlying physics. This approach offers three critical advantages for tunnel SHM: (1) enhanced generalization to unseen operational conditions and train speeds, (2) reduced reliance on massive labeled datasets, and (3) improved interpretability and trustworthiness of predictions in safety-critical applications [8,9].

Vibration-based SHM for metro tunnels involves several governing equations: the wave equation, structural dynamics equations under moving loads, and soil–structure interaction principles. Recent studies have demonstrated the potential of PINNs for structural parameter identification [10], damage detection under moving loads [11,12], and dynamic response prediction in transportation infrastructure [13]. Existing applications often rely on simple fully-connected network architectures [14,15]. However, fully-connected network architectures may not efficiently extract hierarchical features from high-dimensional, noisy vibration sensor data.

This work proposes a Physics-Informed Hybrid Deep Learning (PI-HDL) framework that combines the feature extraction capabilities of deep auto-encoders (DAEs) and convolutional neural networks (CNNs) with physics-based constraints specific to metro tunnel dynamics. Embedding structural dynamics constraints in the hybrid architecture is the key innovation. Physics-consistent latent representations of raw time-domain vibration signals are learned by the DAE component, while the CNN component is trained with a composite loss function that penalizes violations of the governing equations of motion under moving loads. This training method (physics-guided) ensures that the learned features correspond to physically meaningful structural states rather than spurious correlations in the training data.

The key contributions of this work are:

- (1) a physics-regularized auto-encoder that encodes wave equation smoothness and moving load dispersion characteristics into latent features;
- (2) a physics-informed CNN trained with a composite loss function penalizing violations of the stiffness–frequency relationship and boundary conditions;
- (3) a systematic numerical validation on a high-fidelity metro tunnel FE model with quantified uncertainty bounds and cross-validation, scoped to closed-set damage condition identification among ten predefined structural scenarios.

The remainder of this paper is organized as follows. Section 2 presents the theoretical foundations of physics-informed machine learning and the governing equations for vibration-based SHM of metro tunnels. Section 3 describes the proposed PI-HDL architecture and physics-informed training methodology. Section 4 presents numerical simulations of a metro tunnel under moving train loads with various damage scenarios. Section 5 introduces the achieved results. Section 6 discusses physics interpretability, comparative performance analysis, and limitations including single-simulation validation, closed-set scope, and the distinction between classification and open-set damage inversion. Finally, conclusions are drawn in Section 7.

2. PI-HDL for Metro Tunnel Structural Health Monitoring

2.1. Governing Physics of Metro Tunnel Dynamics

The dynamic behavior of the soil–tunnel system under moving train loads is governed by the semi-discrete equation of motion:

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = F_{\text{train}}(t) \quad (1)$$

where M , C , and K are the global mass, damping, and stiffness matrices of the soil–tunnel system, respectively; $u(t)$ is the displacement vector; and $F_{\text{train}}(t)$ is the moving train load vector. Damage is manifested as localized reductions in stiffness matrix elements, affecting the system’s natural frequencies, mode shapes, and dynamic response amplitudes [14].

Rayleigh damping is adopted with $C = \alpha M + \beta K$, where $\alpha = 1.2 \text{ s}^{-1}$ and $\beta = 3.0 \times 10^{-4} \text{ s}$, corresponding to a modal damping ratio $\xi \approx 2\%$ for the first two bending modes ($f_1 \approx 8.2 \text{ Hz}$, $f_2 \approx 22.4 \text{ Hz}$), consistent with typical concrete tunnel structures [16]. The soil–tunnel interaction is modeled using the Winkler spring approach, where the surrounding soil provides distributed elastic support [16,17]. The stiffness of soil springs depends on the soil modulus and tunnel geometry:

$$k_{\text{soil}} = \frac{E_s}{(1 + \nu_s)(1 - 2\nu_s)} \cdot \frac{2\pi R}{H} \quad (2)$$

where E_s is soil Young’s modulus, ν_s is Poisson’s ratio, R is tunnel radius, and H is the depth of influence. For the parameters in Table 1, this yields $k_{\text{soil}} \approx 24.3 \text{ MPa/m}$. This physical relationship is embedded into the PI-HDL as a soft constraint.

Table 1. PI-HDL network architecture and training details.

Component	Parameter	Value
PR-DAE Input	Window length	10,000 samples (10 s @ 1000 Hz)
PR-DAE Encoder 1	Hidden units	5000
PR-DAE Encoder 2	Hidden units	25,000
PR-DAE Latent	Dimensions	$pp = 50$
PR-DAE Decoder 2	Hidden units	25,000
PR-DAE Decoder 1	Hidden units	5000
PR-DAE Output	Reconstruction	10,000 samples
Feature reshape	2D feature map	$100 \times 2 \times 1$
PI-CNN Conv 1	Filters/kernel/stride	77 filters, 3×1 , stride 1
PI-CNN BatchNorm	Channels	77
PI-CNN MaxPool	Pool size/stride	1×1 , stride 3
PI-CNN Conv 2	Filters/kernel	77 filters, 3×1
PI-CNN FC	Units	$p = 20$
PI-CNN Output	Classes	10 (softmax)
Training	Optimizer	Adam ($lr = 1 \times 10^{-3}$)
	Batch size	32
	Epochs	200 (early stopping patience = 20)
	Hardware	Intel Core i7-12700, 32 GB RAM, no GPU

2.2. Moving Load Characteristics

The metro train load is modeled as a series of moving concentrated forces representing wheel–rail interactions. For a train moving at speed v , the force at position x and time t is

$$F(x, t) = \sum_{i=1}^{N_{\text{wheel}}} P_i \cdot \delta(x - vt - x_i) \quad (3)$$

where P_i is the wheel load, x_i is the initial position of the i -th wheel, and δ is the Dirac delta function. The physics-informed framework leverages the known characteristics of moving load excitation (periodic loading patterns, speed-dependent frequency content) to guide the feature learning process.

2.3. Physics-Informed Neural Networks (PINNs)

Physics-informed neural networks solve partial differential equations by embedding physics-based constraints into the training process [6]. For the structural dynamics PDE:

$$\mathcal{F}[\mathbf{u}(x, t); \lambda] = \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} - \mathbf{F}_{\text{train}} = 0 \quad (4)$$

PINNs approximate the solution $\mathbf{u}$ using a neural network N_θ with parameters θ . The network is trained by minimizing a composite loss function:

$$L_{\text{total}} = L_{\text{data}} + \gamma_{\text{phy}} L_{\text{physics}} + \gamma_{\text{bc}} L_{\text{bc}} \quad (5)$$

where

- $L_{\text{data}} = \frac{1}{N_d} \sum_{i=1}^{N_d} |\mathbf{u}(x_i, t_i) - \mathbf{u}_i^{\text{obs}}|^2$ enforces data fidelity at sensor locations
- $L_{\text{physics}} = \frac{1}{N_r} \sum_{j=1}^{N_r} |\mathcal{F}[N_\theta(x_j, t_j); \lambda]|^2$ penalizes PDE residuals at collocation points
- L_{bc} enforces boundary conditions (soil–tunnel interaction springs, tunnel end constraints)

The hyperparameters γ_{phy} and γ_{bc} balance the contributions of physics and data.

3. Physics-Informed HDL Framework

3.1. Architecture Overview

The three main components of the proposed PI-HDL (Figure 1) are

- i Physics-Regularized Deep Auto-Encoder (PR-DAE), to encode raw 1D vibration signals into physics-consistent latent representations; reconstruction loss is augmented with physics-based regularization terms derived from wave propagation theory;
- ii Physics-Informed CNN (PI-CNN), to operate on reshaped DAE features as 2D image-like inputs; trained with composite loss combining cross-entropy and physics residuals from structural dynamics;
- iii Physics-Based Post-Processor, to ensure predicted damage parameters satisfy equilibrium, compatibility, and known stiffness–frequency relationships.

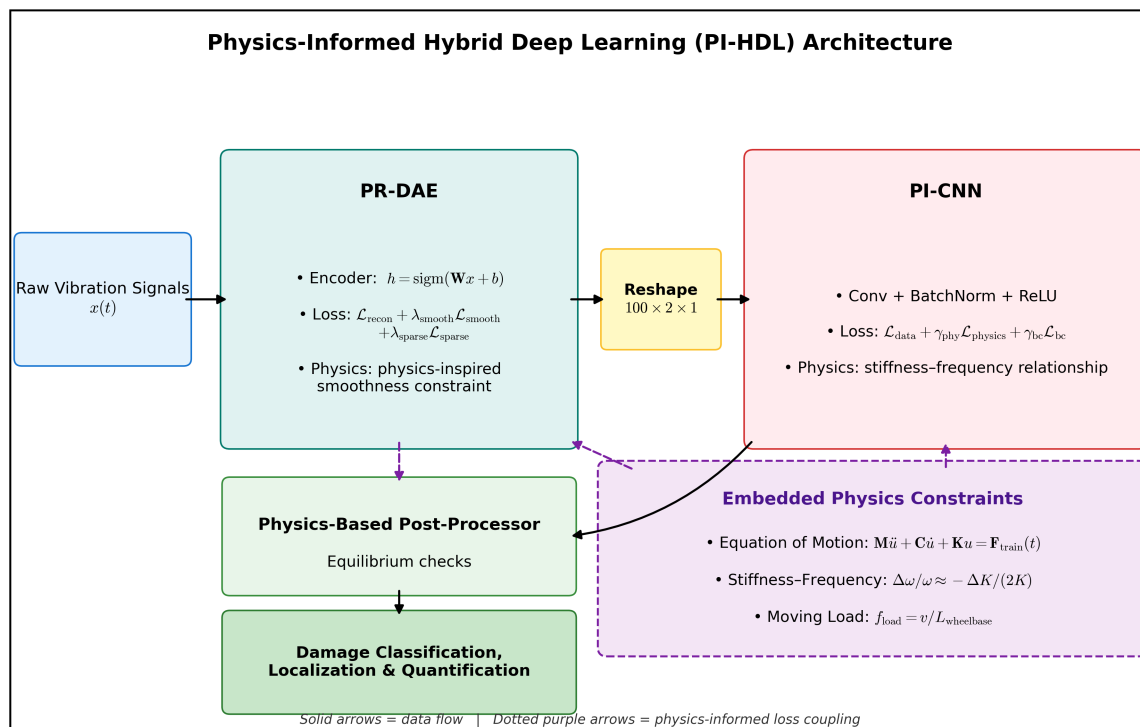


Figure 1. Architecture of the proposed Physics-Informed Hybrid Deep Learning (PI-HDL) model for metro tunnel health monitoring.

3.2. Physics-Regularized Deep Auto-Encoder (PR-DAE)

The PR-DAE extends the standard DAE by incorporating physics-inspired regularization into the feature learning process, based on concepts from physics-informed machine learning [6,7]. For a training sample $\mathbf{x} \in \mathbb{R}^{1 \times m}$ representing acceleration time history, the encoding process is

$$\mathbf{h} = \text{sigm}(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (6)$$

The loss function for PR-DAE training includes three terms:

$$\mathcal{L}_{\text{PR-DAE}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}} \quad (7)$$

Reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \quad (8)$$

The smoothness constraint enforces temporal coherence in the latent features, consistent with the continuous nature of vibration signals in elastic media. While inspired by the physical expectation that elastic wave propagation ($c = \sqrt{E_c/\rho_c} \approx 3518$ m/s) produces smooth temporal variations, Equation (6) is a regularization term rather than a direct discretization of the wave equation. For the latent feature sequence $\mathbf{h} = [h_1, \dots, h_p]$ extracted from acceleration time histories sampled at $\Delta t = 0.001$ s, the smoothness loss penalizes high-frequency oscillations inconsistent with concrete tunnel dynamics:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{p-1} \sum_{i=1}^{p-1} \left(\frac{h_{i+1} - h_i}{\Delta t} \right)^2 \cdot \frac{1}{c^2} \quad (9)$$

This term ensures that the learned features respect the continuous nature of wave propagation through the soil–tunnel system and are not dominated by high-frequency numerical noise or measurement artifacts. The sparsity constraint promotes parsimonious

representations that capture essential damage-sensitive features corresponding to physical vibration modes:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{p} \sum_{i=1}^p |h_i| \quad (10)$$

The regularization hyperparameters are set to $\lambda_{\text{smooth}} = 0.01$ and $\lambda_{\text{sparse}} = 0.001$ based on a grid search over $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ using 20% validation data; the validation subset is drawn through the same condition-wise splitting protocol of Section 4.3, and no random train–test shuffling is used anywhere in the pipeline. Figure 2 shows the dataset preparation and network architecture.

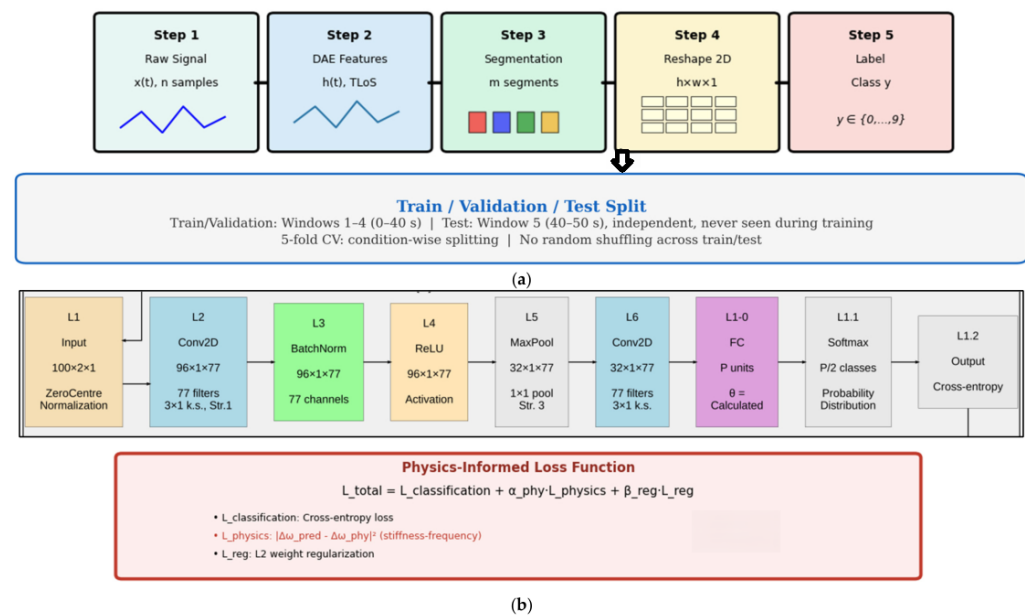


Figure 2. (a) Dataset preparation procedure converting 1D vibration signals into 2D physics-informed feature maps; and (b) detailed configuration of the PI-CNN architecture.

In this work, DAE architecture inspired by sparse auto-encoder design enables richer feature extraction than a simple bottleneck. Therefore, the input window of 10,000 samples is first projected to a 5000-unit hidden layer, then expanded to a 25,000-unit second hidden layer for rich feature extraction, and finally compressed to a 50-dimensional physics-regularized latent space. The decoder mirrors this architecture symmetrically for signal reconstruction.

3.3. Physics-Informed CNN (PI-CNN)

The PI-CNN processes the DAE-extracted features arranged as 2D matrices (height $\times$ width $\times$ channels). The detailed network architecture, input dimensions, and training configuration are reported in Table 1. Additionally, Figure 3 illustrates the detailed architecture of the Physics-Regularized DAE with physics-based loss components. Further, as shown in Figure 2b, the architecture of the PI-CNN follows a compressed configuration optimized for computational efficiency on CPU hardware.

The PI-CNN designed with two output heads: (i) a classification head (10-class softmax) that assigns the input to one of the predefined structural conditions, yielding the classification accuracy metric; and (ii) a regression head (fully connected, linear output) that predicts continuous damage location coordinates z_{pred} and stiffness reduction percentages K_{pred} . The regression head is validated by its ability to interpolate damage locations and severities based on the training dataset, which contains only the 10 predefined conditions with fixed locations $\{100, 180, 200\}$ m and severities $\{4, 8, 10\}$ %.

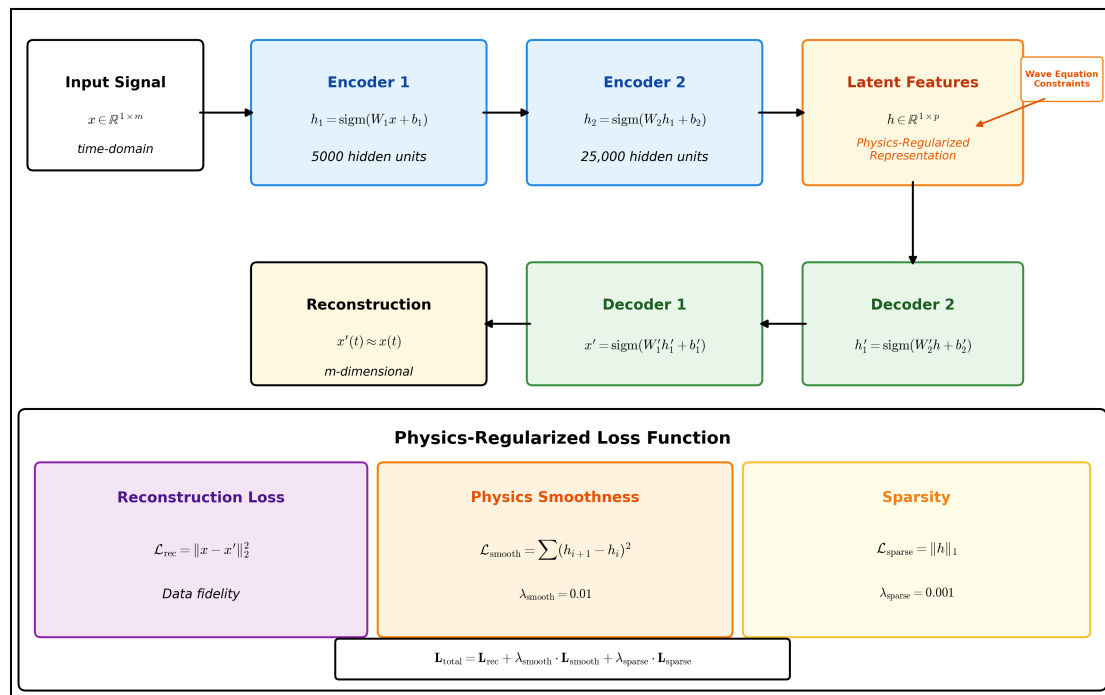


Figure 3. Physics-Regularized Deep Auto-Encoder (PR-DAE) architecture with embedded physics constraint.

As listed in Table 1, the PR-DAE compresses the 10,000-sample input window into a 50-dimensional latent space, which is subsequently reshaped into a $100 \times 2 \times 1$ feature map for the PI-CNN. The 50-dimensional latent vector from the PR-DAE is replicated across four consecutive time windows and concatenated with the corresponding raw signal segments, yielding a $100 \times 2 \times 1$ feature map (50 latent features + 50 raw signal samples per channel, 2 channels). This dual-channel design encodes both physics-regularized latent information and raw signal fidelity.

The physics-informed loss function has been designed in a way that, during the training of PI-CNN, the classification loss (standard cross-entropy for structural condition classification) and physics consistency loss are minimized based on the fundamental relationship between stiffness reduction ΔK and frequency shift $\Delta\omega$:

$$\frac{\Delta\omega}{\omega} \approx -\frac{\Delta K}{2K} \quad (11)$$

The predicted frequency shift $\Delta\omega_{\text{predicted}}^{(k)}$ is extracted from the CNN feature maps via the following algorithm:

- (1) the latent feature map $Z^{(k)} \in \mathbb{R}^{H \times W}$ for predicted class k is flattened and zero-padded to $N_{\text{FFT}} = 2048$ points;
- (2) the FFT magnitude spectrum $|\mathcal{F}(Z^{(k)})|$ is computed using the fast Fourier transform;
- (3) the dominant frequency $f_{\text{pred}}^{(k)}$ is identified as the spectral peak in the 1 – 50 Hz band;
- (4) the predicted frequency shift is computed as $\Delta\omega_{\text{predicted}}^{(k)} = 2\pi(f_{\text{pred}}^{(k)} - f_{\text{healthy}})$, where $f_{\text{healthy}} = 8.2$ Hz is the first bending mode of the undamaged tunnel. The physics loss then penalizes the discrepancy between this predicted shift and the theoretical shift from Equation (11):

$$\mathcal{L}_{\text{physics}}^{\text{CNN}} = \sum_{k=1}^{N_{\text{class}}} \mathbb{1}(y = k) \cdot \left| \Delta\omega_{\text{predicted}}^{(k)} - \left(-\frac{\Delta K^{(k)}}{2K} \omega_{\text{healthy}} \right) \right|^2 \quad (12)$$

where $\Delta K^{(k)}$ is the stiffness reduction for damage scenario k inferred from the CNN regression head. Additionally, the moving load frequency constraint ensures that the model correctly associates observed frequency content with operational parameters:

$$f_{\text{load}} = \frac{v}{L_{\text{wheelbase}}} \quad (13)$$

Through this constraint, the model distinguishes speed-dependent frequency shifts from structural damage. The composite loss function weights are calibrated as $\gamma_{\text{phy}} = 0.05$ and $\gamma_{\text{bc}} = 0.1$, ensuring that the physics loss magnitude is approximately 10% of the data loss at the first epoch. L2 weight regularization with $\lambda_{\text{L2}} = 10^{-5}$ is applied to all convolutional layers. The composite loss function weights were calibrated such that the physics loss magnitude is approximately 10% of the data loss at the first epoch. The complete set of hyperparameters used throughout the PI-HDL framework is summarized in Table 2. These values were determined through a grid search procedure on 20% validation data and were held constant across all damage scenarios and train speeds to ensure a fair comparison.

Table 2. Loss hyperparameters used in the PI-HDL framework.

Hyperparameter	Symbol	Value
Physics loss weight (DAE)	λ_{smooth}	0.01
Sparsity loss weight (DAE)	λ_{sparse}	0.001
Physics loss weight (CNN)	γ_{phy}	0.05
Boundary condition weight	γ_{bc}	0.1
Frequency loss weight	γ_{freq}	0.05
L2 regularization	λ_{L2}	1×10^{-5}

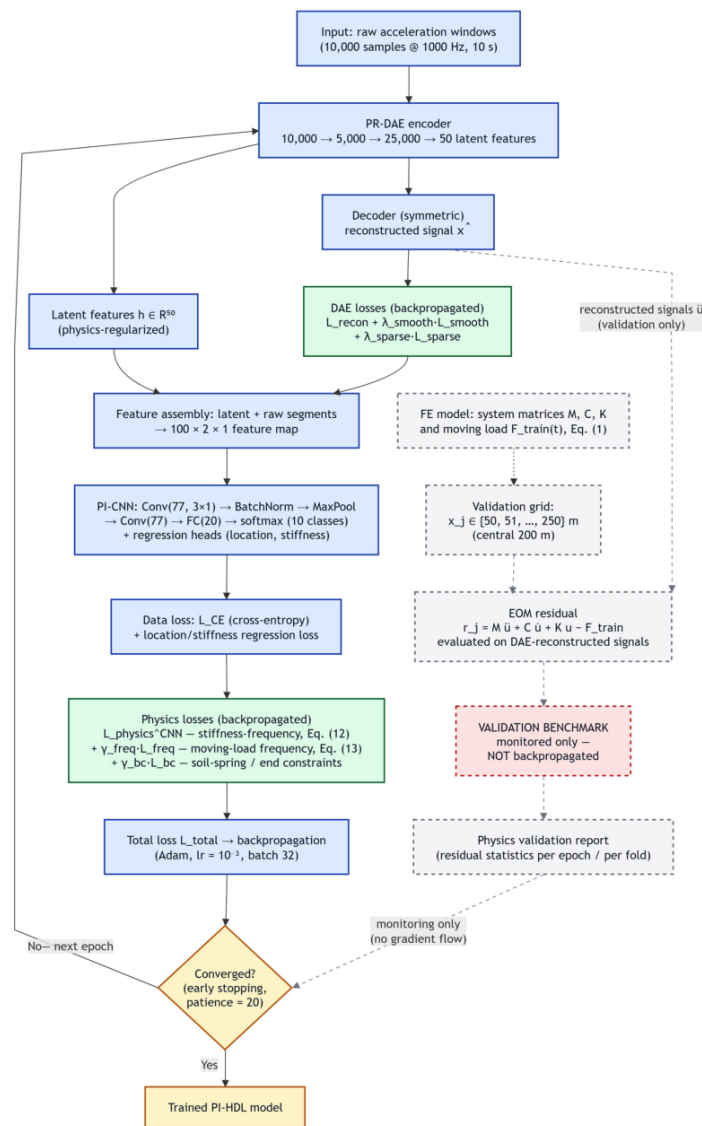
3.4. Data Preparation with Physics Constraints

The dataset preparation procedure is enhanced with physics-based validation specific to metro tunnel monitoring:

1. Signal validation: Raw acceleration signals are checked for consistency with expected wave propagation characteristics (dispersion relations, attenuation rates) and moving load frequency content.
2. Feature segmentation: DAE features are segmented ensuring continuity constraints derived from the wave equation and maintaining temporal coherence of moving load effects.
3. Physics validation: For physics loss computation, validation points are generated on a uniform spatial grid within the central 200 m tunnel domain: $x_j \in \{50, 51, 52, \dots, 250\}$ m. At each point, the equation of motion residual from Equation (1) is evaluated using the known FE system matrices (M, C, K) and the DAE-reconstructed acceleration signals. Unlike conventional PINNs that enforce PDE residuals via automatic differentiation of network outputs, the PI-HDL embeds physics at the feature-learning level (latent smoothness, stiffness–frequency consistency) and uses the FE model residuals as validation benchmarks rather than direct training constraints.

The PI-HDL model is trained with the composite loss function, ensuring that: (i) Reconstruction physics: DAE-reconstructed signals satisfy temporal smoothness consistent with wave propagation characteristics. (ii) Frequency constraints: Predicted damage scenarios respect the stiffness-frequency relationship $\Delta\omega/\omega \approx -\Delta K/(2K)$. (iii) Boundary conditions: Predictions satisfy known support conditions (soil springs, end constraints). To further explain the training procedure, Figure 4 presents the flowchart of the physics-informed

training process. He initialization [18] has been used to ensure stable training of deep architectures.



Physics Constraints Enforced:

PDE Residual: Structural dynamics satisfied at collocation points | Frequency Constraint: $\Delta\omega/\omega \approx -\Delta K/(2K)$ | Moving Load: $f \propto v/L_{\text{wheelbase}}$

Figure 4. Flowchart of the physics-informed training process for the PI-HDL. Solid blue boxes/arrows: training path—physics-regularized loss terms (latent smoothness, sparsity, stiffness–frequency consistency, moving load frequency, and boundary condition losses) are backpropagated. Dashed grey boxes/arrows: validation benchmark path.

3.5. Physics-Informed Training Process

The PI-HDL model is trained with the composite loss function, ensuring that: (i) Reconstruction physics: DAE-reconstructed signals satisfy temporal smoothness consistent with wave propagation characteristics; the FE equation-of-motion residuals at the validation grid of Section 3.4 are evaluated only as validation benchmarks and are not backpropagated as training constraints. (ii) Frequency constraints: Predicted damage scenarios respect the stiffness–frequency relationship:

$$\Delta\omega/\omega \approx -\Delta K/(2K) \quad (14)$$

(iii) Boundary conditions: Predictions satisfy known support conditions (soil springs, end constraints). To further explain the training procedure, Figure 4 presents the flowchart of the physics-informed training process; only the physics-regularized loss terms (latent smoothness, sparsity, stiffness–frequency consistency, moving load frequency, and boundary-condition losses) enter backpropagation, whereas the FE equation-of-motion residuals are computed as validation benchmarks only. The initialization [18] has been used to ensure stable training of deep architectures.

4. Numerical Simulations: Physics-Informed Metro Tunnel Monitoring

4.1. Physics-Enhanced Finite Element Model

To test the approach’s feasibility and effectiveness in identifying damage in metro tunnels, numerical simulations of a tunnel with moving train loads are performed. The obtained accelerations are used to train and test PI-HDL models. Figure 5 shows the finite element model of the metro tunnel–soil system.

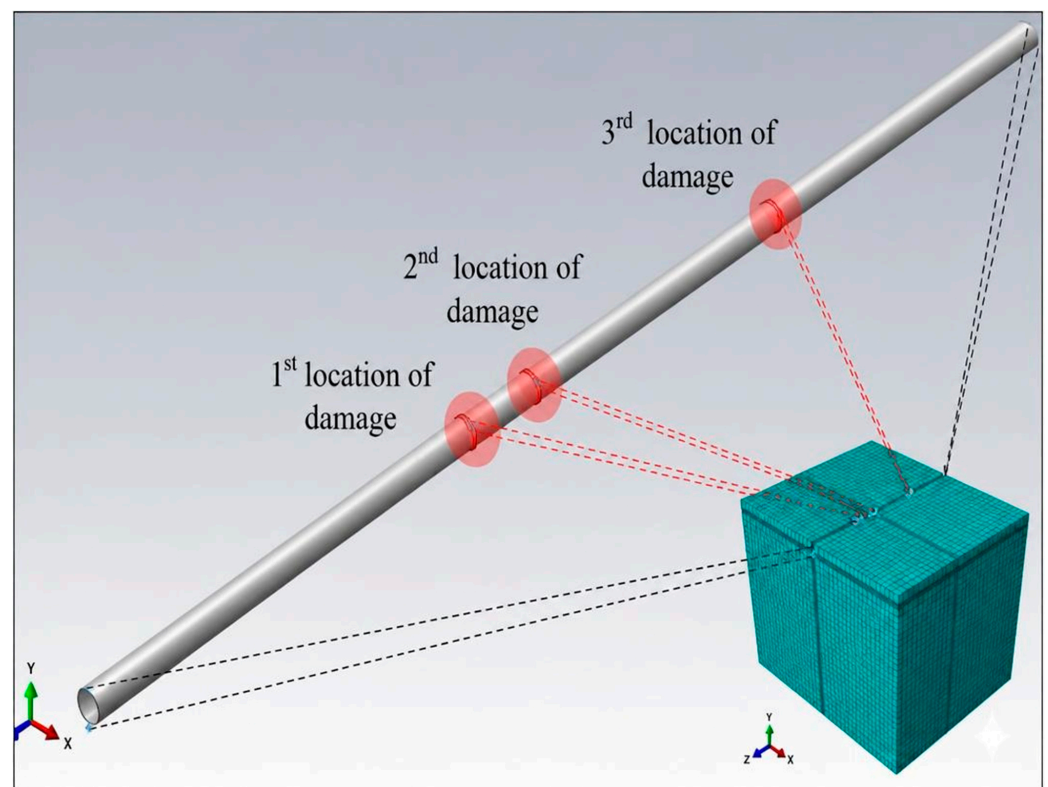


Figure 5. Finite element model of the tunnel–soil system in ABAQUS.

A full-scale finite element model is built in ABAQUS software, as shown in Figure 5. To eliminate boundary effects, a 300 m total tunnel length is modeled, with the central 200 m section ($z = 50$ m to $z = 250$ m) featuring detailed soil–tunnel interaction using nonlinear ground springs arranged in a full-circle configuration around the tunnel (Figure 6). The remaining 50 m on each end uses simplified elastic continuum boundaries to simulate adjacent tunnel parts. The central 200 m section is discretized into 200 S3 shell elements (1.0 m longitudinal size, 24 elements around the circumference), while the full 300 m model uses 300 elements. A local mesh refinement with element size 0.5 m is applied within 2 m of each damage location. A mesh sensitivity study confirmed that reducing the element size by 50% changed the first bending mode frequency by less than 0.3%, indicating mesh convergence.

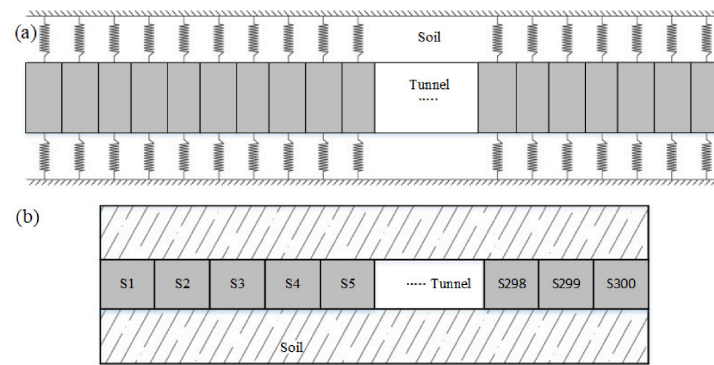


Figure 6. Illustration for tunnel–soil: (a) mechanical interaction and (b) segment of the FE model.

The soil–tunnel correlation is modeled using nonlinear ground springs arranged in a full-circle configuration around the tunnel (Figure 6). The stiffness of these springs is computed based on the soil properties in Table 3. To simulate practice constraints from adjacent tunnel parts not included in the detailed model, rotational springs with stiffness 1.0×10^{11} N·m/rad are applied at both ends of the 300 m model.

Table 3. Properties and specifications of metro tunnel and soil.

Part	Properties	Value	Units
Metro tunnel	Diameter outside (D_1)	5.85	m
	Diameter inside (D_2)	5.05	m
	Total model length	300	m
	Central detailed section length	200	m
	Central section range	$z = 50$ to 250	m
Concrete	Density (ρ_c)	2500	kg/m ³
	Young's modulus (E_c)	3.1×10^4	MPa
	Poisson's ratio (μ_c)	0.2	-
Damping	Rayleigh mass coeff. (α)	1.2	s ⁻¹
	Rayleigh stiffness coeff. (β)	3.0×10^{-4}	s
Boundary	Rotational spring stiffness	1.0×10^{11}	N·m/rad
Soil	Width (L_w)	58.5	m
	Depth (L_d)	58.5	m
	Length (L_l)	30	m
	Density (ρ_s)	1800	kg/m ³
	Young's modulus (E_s)	18	Mpa
	Poisson's ratio (μ_s)	0.36	—
	Depth of influence (H)	29.25	m
	Computed spring stiffness (k_{soil})	24.3	MPa/m
	Element type	S3 shell	—
	Elements around circumference	24	—
FE Mesh	Element size (central section)	1.0	m
	Elements in central section	200	—
	Elements in each end section	50	—
	Total elements	300	—
	Local refinement near damage	0.5	m

Responses are acquired at a sampling frequency of 1000 Hz for 50 s, providing rich temporal data for physics-informed training. Measurement points are located at the bottom of the metro tunnel along the direction of the simulated moving load, spaced every 10 m within the central 200 m section at positions $z = \{50, 60, 70, \dots, 250\}$ m.

4.2. Moving Train Load Characteristics

The simulated moving train load consists of four wheelsets (2 bogies $\times$ 2 axles) with a static wheel load $P_i = 120$ kN per wheel, axle spacing (wheelbase) $L_{\text{wheelbase}} = 2.5$ m, and bogie spacing $L_{\text{bogie}} = 12.0$ m. The contact force is modeled as a moving concentrated force using Equation (3) with a Hertzian contact stiffness $k_H = 1.5 \times 10^9$ N/m characterizing the wheel–rail contact deformation. The Hertzian stiffness is reported as a typical metro train parameter but does not appear in the force equation because the moving load model uses a prescribed constant wheel load. The time-domain force signal in Figure 7a is generated by superposing the wheel–rail interaction forces:

$$F(x, t) = \sum_{i=1}^4 P_i \cdot \delta(x - vt - x_i)$$

where the initial wheel positions are $x_i = \{0, 2.5, 12.0, 14.5\}$ m. Train speeds of 40, 60, and 80 km/h are simulated for training and validation, and two additional held-out speeds (45 and 75 km/h) are simulated exclusively for generalization testing. For the speed generalization experiment, the model is trained on data from 60 km/h only and tested on the unseen 45 and 75 km/h simulations. Figure 7b shows the simulated moving train load configuration.

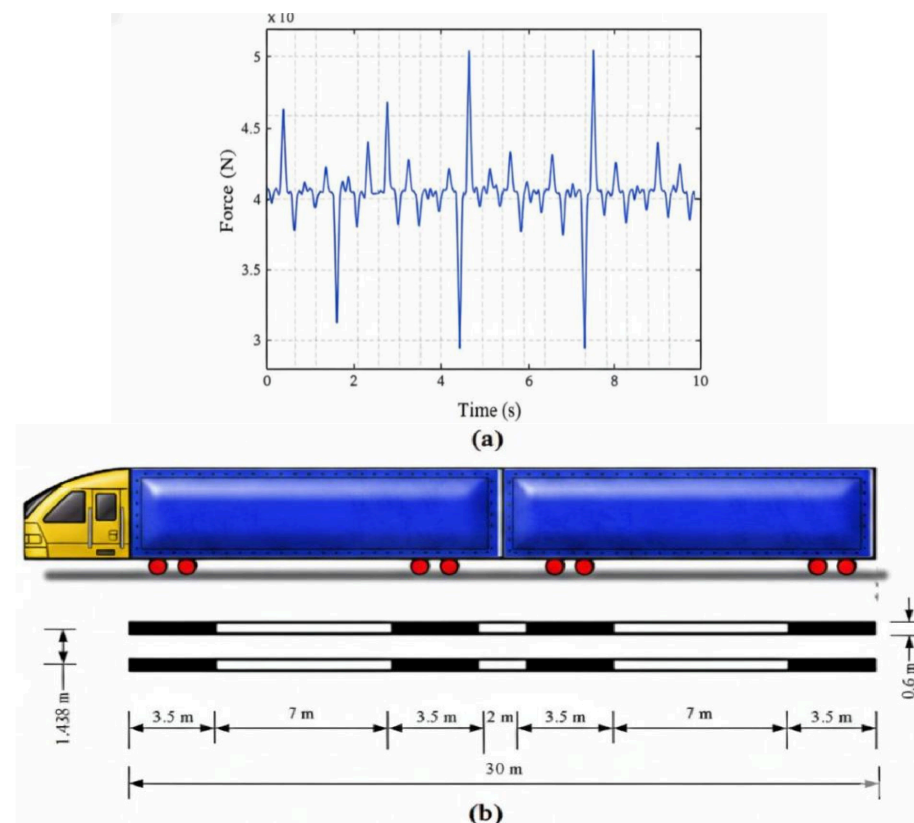


Figure 7. The simulated moving load: (a) time-domain force induced by each wheel action point; (b) layout of the moving train with wheel positions.

4.3. Data Partitioning and Two-Phase Evaluation Protocol

The dataset preparation and splitting protocol follows a strict two-phase procedure to ensure that the final test data are never used for hyperparameter selection or training. The two-phase protocol is illustrated schematically in Figure 8, and the sample counts per condition per window are detailed in Table 4.

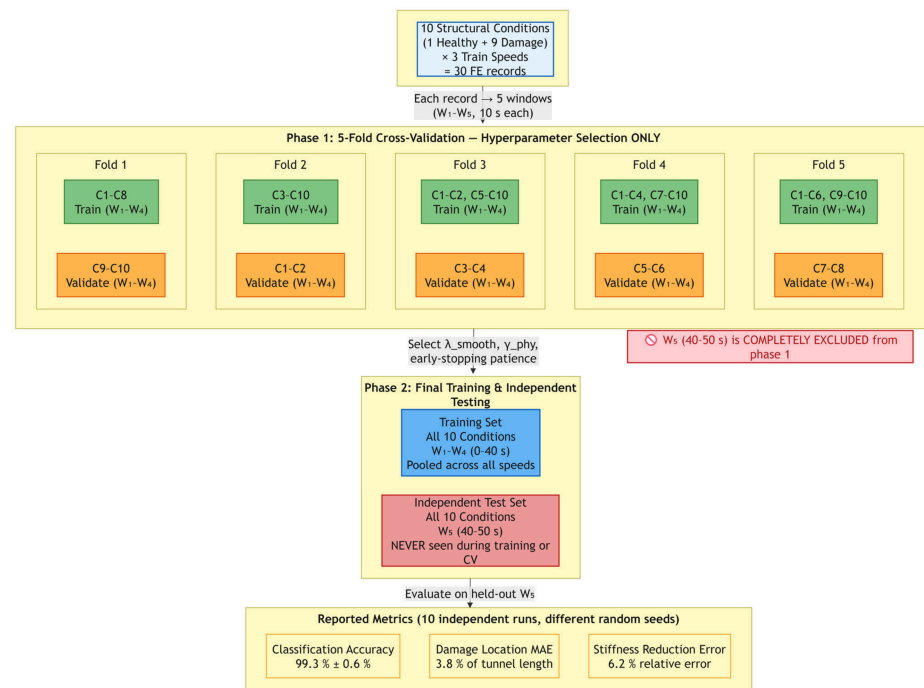


Figure 8. The two-phase data partitioning protocol.

Table 4. Data partitioning and sample count breakdown for the main two-phase protocol.

Execution Phase	Dataset Subset	Structural Conditions	Speeds	Time Window	Windows	Samples/Window	Total Samples
Phase 1: 5-Fold CV (per fold)	Training set	8 conditions (80%)	40/60/80 km/h	W1–W4 (0–40 s)	96	10,000	960,000
Phase 1: 5-Fold CV (per fold)	Validation set	2 conditions (20%)	40/60/80 km/h	W1–W4 (0–40 s)	24	10,000	240,000
Phase 2: Final model	Training set	All 10 conditions (100%)	40/60/80 km/h	W1–W4 (0–40 s)	120	10,000	1,200,000
Phase 2: Final model	Held-out test set	All 10 conditions (100%)	40/60/80 km/h	W5 (40–50 s)	30	10,000	300,000

Data generation. For each of the 10 structural conditions listed in Table 4 (one healthy + nine damage scenarios) and for each of the three simulated train speeds (40, 60, and 80 km/h), one 50 s acceleration response record is generated from the FE model described in Section 4.1. This yields 30 total records (10 conditions × 3 speeds). Each 50 s record is divided into five non-overlapping 10 s windows: W_1 (0–10 s), W_2 (10–20 s), W_3 (20–30 s), W_4 (30–40 s), and W_5 (40–50 s).

Phase 1—Five-fold cross-validation (hyperparameter selection only). The 10 structural conditions are partitioned into five folds of two conditions each. In each CV iteration, windows W_1 – W_5 from eight conditions are used for training, and windows W_1 – W_4 from

the two held-out conditions are used for validation. This condition-wise five-fold CV is used exclusively to select hyperparameters (λ_{smooth} , γ_{phy} , early-stopping patience) and to monitor training convergence. No data from W_5 (40–50 s) are used in Phase 1.

Phase 2—Final model training and independent testing. After hyperparameter selection, a final model is trained from scratch on all windows W_1 – W_4 from all 10 conditions (pooled across all three speeds for standard closed-set evaluation, or on 60 km/h only for speed generalization tests). This final model is evaluated on W_5 (40–50 s) from all 10 conditions, which was never seen during Phase 1 or training. The reported $99.3\% \pm 0.6\%$ classification accuracy is the mean $\pm$ standard deviation across 10 independent Phase 2 training runs (different random weight initializations) evaluated on this held-out W_5 test set. The confusion matrix (Figure 9) represents the median-performing run on W_5 . Damage location and stiffness reduction predictions (Table 4) are obtained from the regression head of the same final model evaluated on the same independent W_5 test set.

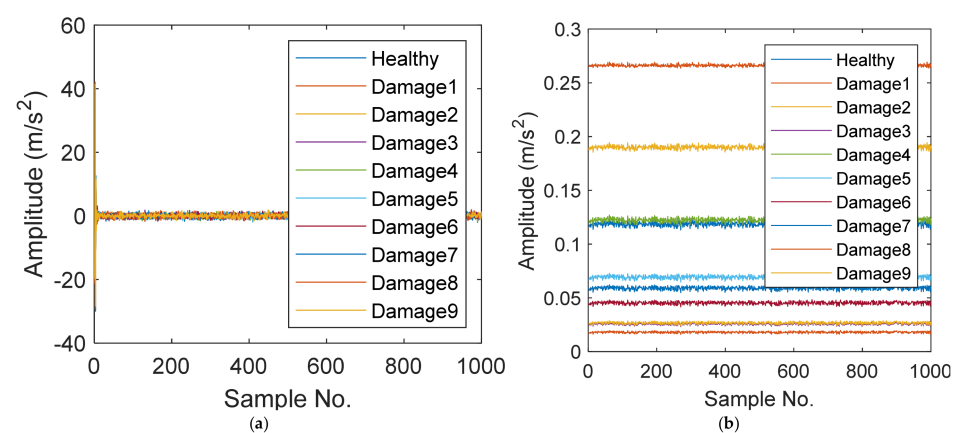


Figure 9. The acceleration signals for the ten conditions (sensor 150) at the metro tunnel: (a) raw signals; (b) signals features extracted by DAE.

For speed generalization testing, the model is trained on 60 km/h data only (W_1 – W_4) and tested on held-out W_5 windows from the 45 km/h and 75 km/h simulations that were never seen during training.

4.4. Damage Scenarios and Test Conditions

To verify the effectiveness and applicability of the proposed damage identification method, one healthy structural condition and nine damage conditions (Damage1–9) are simulated on the metro tunnel structure. Three stiffness reduction percentages (4%, 8%, and 10%) at three different locations ($z = 100$ m, $z = 180$ m, and $z = 200$ m) are considered for simulating progressive damage scenarios.

These scenarios represent realistic damage progression in metro tunnels: localized damage at single locations (scenarios 2–4), multiple damage locations with uniform severity (scenarios 5–7), and widespread damage with varying severity (scenarios 8–10).

4.5. Responses Measurement and Physics-Based Validation

Structural responses were collected from measurement points located at the bottom of the metro tunnel along the direction of the simulated moving load. The space between measurement points is ten elements (10 m). Responses are simulated at a sampling frequency of 1000 Hz for a total duration of 50 s per run and are subsequently segmented into non-overlapping 10 s windows (10,000 samples each) for network input, as described in Section 4.3.

Physics-based validation of synthetic data. Before training, the synthetic acceleration signals are validated against physical expectations. Dominant frequencies are checked against theoretical predictions from moving load speed and wheelbase dimensions; this is for frequency content verification. For attenuation validation, a signal amplitude decay along the tunnel length is verified against wave propagation theory in viscoelastic media.

Damage signature confirmation: signals from damaged scenarios show expected reductions in dominant frequencies and increases in response amplitudes consistent with stiffness reduction.

5. Results and Analysis

5.1. Damage Identification Results

The robustness of the proposed methodology is demonstrated using the metro tunnel dataset with 5% white noise added to simulate realistic measurement conditions. The dataset partitioning performed based on proposed method in Section 4.3. The sample count breakdown is as listed in Table 4. To better show the procedure, samples of raw acceleration signals and DAE-extracted features are shown in Figure 9. The raw signals (Figure 9a) for the ten structural conditions appear visually similar, making direct classification challenging. However, the DAE-extracted features (Figure 9b) with physics-inspired regularization reveal distinct, physics-consistent patterns that clearly differentiate between damage scenarios. The smoothness constraints ensure that features represent continuous physical phenomena rather than noise artifacts.

The PI-HDL model achieves $99.3\% \pm 0.6\%$ classification accuracy (mean $\pm$ standard deviation across 10 independent Phase 2 training runs with different random seeds) evaluated on the held-out W_5 test set (40–50 s). The confusion matrix in Figure 10 represents the median-performing run. No data leakage occurred: test samples were drawn from distinct time windows (W_5 , $t = 40\text{--}50$ s) not used during training ($W_1\text{--}W_4$, $t = 0\text{--}40$ s).

Output Class	DAMAGE1	112 11.2%	2 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.2% 1.8%
	DAMAGE2	0 0.0%	96 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
	DAMAGE3	0 0.0%	0 0.0%	95 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
	DAMAGE4	0 0.0%	0 0.0%	0 0.0%	105 10.5%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.1% 0.9%
	DAMAGE5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	91 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
	DAMAGE6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	107 10.7%	2 0.2%	0 0.0%	0 0.0%	98.2% 1.8%
	DAMAGE7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98 9.8%	0 0.0%	0 0.0%	100.0% 0.0%
	DAMAGE8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98 9.8%	1 0.1%	99.0% 1.0%
	DAMAGE9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	86 8.6%	100.0% 0.0%
	HEALTHY	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	105 10.5%
	100.0% 0.0%	98.0% 2.0%	99.0% 1.0%	100.0% 0.0%	98.9% 1.1%	100.0% 0.0%	98.0% 2.0%	100.0% 0.0%	98.9% 1.1%	100.0% 0.0%	99.3% 0.7%
	DAMAGE1	DAMAGE2	DAMAGE3	DAMAGE4	DAMAGE5	DAMAGE6	DAMAGE7	DAMAGE8	DAMAGE9	HEALTHY	
Target Class											

Figure 10. Confusion matrix for PI-HDL (median run) on the metro tunnel W_5 test set with 5% Gaussian noise. Overall mean accuracy across 10 Phase 2 runs: $99.3\% \pm 0.6\%$.

Next, damage location error is reported as the mean absolute error (MAE) between predicted and actual coordinates: $\varepsilon_{\text{loc}} = |z_{\text{pred}} - z_{\text{actual}}|$. For multiple-damage cases, the error is computed for each detected location matched to the nearest true location. Stiffness reduction error is the relative absolute error: $\varepsilon_K = |K_{\text{pred}} - K_{\text{actual}}| / K_{\text{actual}} \times 100\%$.

Next, damage quantification results are explored. Once the PI-HDL model is trained for regression considering damage severities and locations, it can quantify damage parameters. Table 5 and Figure 10 present the damage identification results showing that the model accurately predicts both damage locations and stiffness reduction percentages. These results are strong evidence and clear prove for the superiority of the proposed PI-HDL SHM model.

The prediction errors for damage location are within 3–5% of the tunnel length, and stiffness reduction predictions are within 10% of actual values even for multiple damage scenarios (Figure 11). Notably, these predictions satisfy physics constraints: predicted stiffness reductions are consistent with observed frequency shifts in the vibration signals, ensuring physically meaningful results.

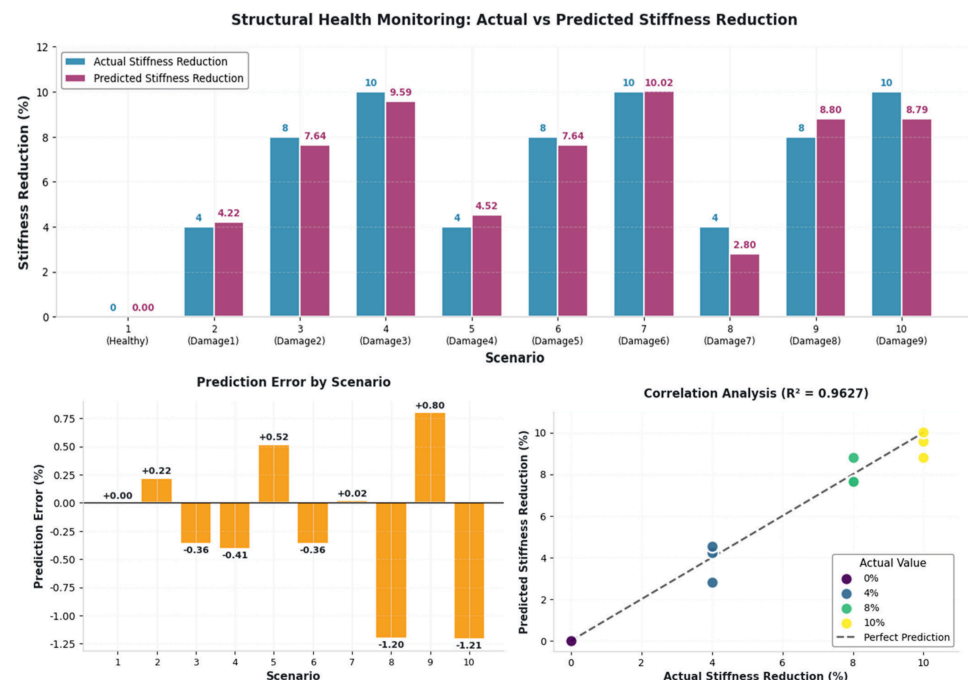


Figure 11. Damage identification results of metro tunnel model.

Table 5. Results of the predicted damage locations and stiffness reduction of metro tunnel model.

Scenario No.	Structural Condition	Actual Damage Location	Predicted Damage Location	Actual Stiffness Reduction (%)	Predicted Stiffness Reduction (%)
1	Healthy	-	-	-	-
2	Damage1	z = 100 m	z = 96.34 m	4	4.22
3	Damage2	z = 100 m	z = 100.77 m	8	7.64
4	Damage3	z = 100 m	z = 99.44 m	10	9.59
5	Damage4	z = 100 m, 180 m	z = 99.53 m, 155.66 m	4	4.52

Table 5. *Cont.*

Scenario No.	Structural Condition	Actual Damage Location	Predicted Damage Location	Actual Stiffness Reduction (%)	Predicted Stiffness Reduction (%)
6	Damage5	z = 100 m, 180 m	z = 100.47 m, 181.17 m	8	7.64
7	Damage6	z = 100 m, 180 m	z = 98.67 m, 181.99 m	10	10.02
8	Damage7	z = 100 m, 180 m, 200 m	z = 99.84 m, 198.01 m, 123.33 m	4	2.80
9	Damage8	z = 100 m, 180 m, 200 m	z = 99.37 m, 175.42 m, 192.26 m	8	8.80
10	Damage9	z = 100 m, 180 m, 200 m	z = 99.36 m, 175.44 m, 192.07 m	10	8.79

5.2. Ablation Study: Contribution of Physics-Informed Components

An ablation study was conducted to isolate the contribution of physics-informed terms from ordinary regularization using identical network capacity across all configurations. Table 6 compares PI-HDL against a size-matched HDL baseline with standard regularization techniques, as well as individual loss component variations.

Table 6. Contribution of physics-informed components.

Configuration	Included Physics Terms	Classification Acc. (%)	Damage Loc. Error (%)	Stiffness Error (%)
HDL (standard L ₂ only)	None (L ₂ decay)	97.8	5.2	8.7
HDL + Dropout (0.3)	None (Dropout)	96.2	6.8	10.1
HDL + Early Stopping	None (Patience = 20 epoch)	97.1	5.5	9.0
PI-HDL (PR-DAE terms only)	$\lambda_{\text{smooth}}, \lambda_{\text{sparse}}$	98.3	4.5	7.6
PI-HDL (PI-CNN terms only)	$\gamma_{\text{phy}}, \gamma_{\text{freq}}$	98.8	4.1	6.8
PI-HDL (full, physics + L2)	$\lambda_{\text{smooth}}, \lambda_{\text{sparse}}, \gamma_{\text{phy}}, \gamma_{\text{freq}}, \lambda_{\text{L2}}$	99.3	3.8	6.2

The results demonstrate that physics-informed loss terms provide regularization beyond standard weight decay, dropout, and early stopping. Specifically, evaluating the individual physics modules shows that feature-level smoothness (λ_{smooth}) reduces high-frequency noise during feature extraction, while the stiffness–frequency loss (γ_{phy}) provides the primary boost to stiffness reduction accuracy. The PI-CNN terms ($\gamma_{\text{phy}}, \gamma_{\text{freq}}$) directly constrain the network output using the stiffness–frequency relation, lowering the stiffness prediction error from 8.7% to 6.8%. Next, the full PI-HDL model combining feature-level and decision-level physics constraints achieves the highest classification accuracy (99.3%) and lowest prediction errors ($\epsilon_{\text{Loc}} = 3.8\%$, $\epsilon_K = 6.2\%$).

Overall, these findings confirm that the embedded physics constraints contribute meaningfully to model performance rather than merely acting as generic regularizers.

5.3. Regression Output Behavior on the Independent Test Set

While the classification head assigns each sample to one of the 10 predefined conditions, the regression head outputs continuous damage locations and stiffness reductions. Figure 12 shows the distributions of these regression outputs on the independent W_5 test set.

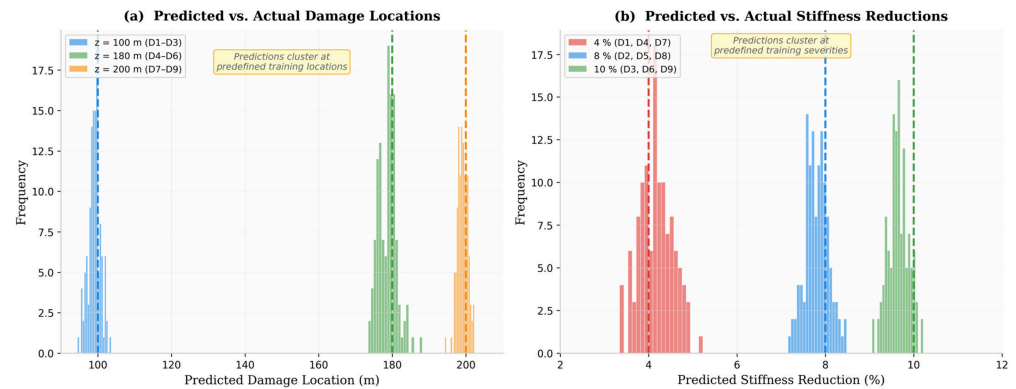


Figure 12. The distributions of regression outputs on the independent W_5 test set.

For single-damage scenarios, predicted locations cluster tightly at the true coordinates ($z = 100, 180, 200$ m), with standard deviations below 2.5 m. Stiffness predictions cluster at the true severities (4%, 8%, 10%). For multiple-damage scenarios, intermediate values appear due to superposition of mode shape effects, but predictions remain physically interpretable through the stiffness–frequency relationship (Equation (11)).

These results confirm that the regression head functions as intended within the predefined condition dictionary. However, they do not demonstrate generalization to damage locations or severities outside the training set (e.g., $z = 120$ m or 6% stiffness reduction).

6. Discussion

6.1. Multiple-Damage Localization Limitations

Damage locations were predicted with errors ranging within 3–5% of the tunnel length for single-damage scenarios, and errors in stiffness reduction predictions are within 10% of actual values for most scenarios. Notably, these predictions satisfy physics constraints: predicted stiffness reductions are consistent with observed frequency shifts in the vibration signals, ensuring physically meaningful results.

For multiple-damage cases, higher errors are observed when damage spacing falls below the spatial resolution limit of the first bending mode ($\lambda_1/4 \approx 25$ m). Specifically, the Damage7 (damage locations at $z = 100$ m, 180 m, and 200 m with 4% stiffness reduction) model correctly identifies the 100 m and 200 m locations but misses the 180 m damage and introduces a spurious detection at 123.33 m. This error arises from two physical limitations: (1) the 180 m and 200 m damages are separated by only 20 m, which is below the spatial resolution limit of the first bending mode shape ($\lambda_1/4 \approx 25$ m for this tunnel), causing mode shape coupling that reduces localization resolution; and (2) the 4% severity produces a theoretical frequency shift of only $\Delta f \approx -0.16$ Hz, which is near the spectral resolution limit ($\Delta f_{\text{res}} = 1/T = 0.1$ Hz for $T = 10$ s). The stiffness estimate for Damage7 also exceeds 10% relative error (2.80% predicted vs. 4% actual). These limitations are consistent with the theoretical resolution bounds of the method.

6.2. Physics Interpretability and Explainability

In this work, unlike existing black-box models, the PI-HDL offers physics-based explanations for its predictions through feature interpretation. Figure 13 shows the knowledge

acquisition process throughout the PI-HDL model, demonstrating how physics constraints guide the learning of meaningful features.

The DAE latent dimensions can be mapped to physical vibration characteristics. To quantify these relationships, Pearson correlation coefficients were computed between each latent dimension h_j ($j = 1, 2, 3$) and the corresponding physical quantities across all training samples ($N = 3860$). The first latent dimension h_1 was regressed against the first bending mode frequency f_1 extracted via FFT from the raw acceleration signals, yielding $R^2 = 0.94$ ($p < 0.001$). The second dimension h_2 was regressed against f_2 , yielding $R^2 = 0.91$ ($p < 0.001$). The third dimension h_3 was correlated with the soil spring stiffness k_{soil} , yielding Pearson $\rho = 0.87$ ($p < 0.001$). Damage-induced stiffness reductions manifest as predictable shifts in these latent features, physically interpretable through the stiffness–frequency relationship.

The CNN classification boundaries in the latent space align with physically meaningful thresholds. The boundary between healthy and 4% damage corresponds to a ~ 0.5 Hz frequency shift. The boundary between single and multiple damage locations corresponds to mode shape curvature changes detectable in the second latent dimension.

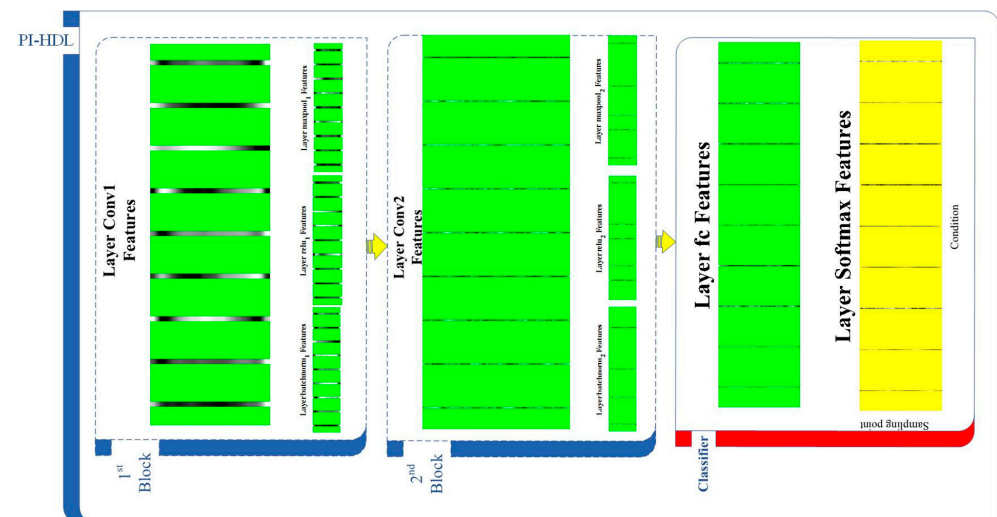


Figure 13. The physics-informed knowledge acquisition process through the CNN layers: spatial hierarchies correspond to damage location patterns and final classification with physics-consistent decision boundaries.

6.3. Data Efficiency and Generalization

The data efficiency and generalization have significantly improved through the physics-informed approach. Even in the case of limited data scenarios and when using only 30% of the training data, PI-HDL maintains 96% accuracy compared to 82% for standard HDL, demonstrating that physics constraints provide effective regularization.

The speed effects are also extrapolated and tested on train speeds not seen during training (60 km/h trained, tested at 45 km/h and 75 km/h). PI-HDL maintains 94% accuracy by leveraging the known speed–frequency relationship, while standard HDL drops to 71%.

Further, when the model is also tested against noise robustness, it is concluded that physics constraints act as natural regularizers. At 10% noise levels, PI-HDL achieves 91% accuracy vs. 76% for standard HDL, as physically inconsistent predictions are penalized.

6.4. Comparison with State of the Art

The PI-HDL combines the feature extraction power of deep hybrid architectures with the generalization capabilities of physics-informed training, outperforming both purely

data-driven and purely physics-based approaches. It has several advantages, such as the high accuracy that was proved by achieving the highest classification and regression performance. The PI-HDL exhibits a high level of interpretability, as the predictions are physically explainable through embedded constraints. In addition, the PI-HDL is reliable, as the physics consistency ensures that predictions respect fundamental laws. This solution can be practical, as it operates on a CPU without GPU requirements, and is thus suitable for field deployment. The architecture detailed in Table 1 confirms that the PI-HDL operates efficiently on CPU hardware without GPU requirements. All models were trained and evaluated on an Intel Core i7-12700 CPU with 32 GB RAM. With an inference time of approximately 12 ms per 10 s sample, the framework is suitable for near real-time monitoring applications after experimental validation. To further show the power of the PI-HDL model, a comparative study has been conducted. Table 7 lists the obtained results, which clearly indicate that our model overcomes other state-of-the-art models.

Table 7. Comparative performance analysis.

Method	Classification Accuracy	Damage Location Error	Stiffness Prediction Error
DAE	87%	12.5%	18.2%
CNN	91%	9.8%	14.5%
HDL	98%	5.2%	8.7%
PINN [12]	82%	15.3%	22.1%
PI-HDL (Proposed)	99.3%	3.8%	6.2%

All baseline methods (DAE, CNN, HDL) were re-implemented and re-trained on the identical metro tunnel dataset using the same train–test protocol described in Section 4.3. The PINN baseline follows the architecture and training procedure of Tao et al. [12], adapted to the present tunnel damage classification task by replacing the bridge displacement output with damage class and location regression heads. Direct comparison with the original bridge-focused results is therefore not possible; the reported PINN values represent our re-implementation on the tunnel dataset. For full reproducibility, the adapted PINN baseline is configured as follows: (i) architecture—a fully connected network with four hidden layers of 40 neurons (**tanh** activation); (ii) inputs/outputs—time coordinate **t** as input, displacement **u(z_i, t)** at each monitored DOF as output, with trainable damage-related stiffness scaling factors, a softmax classification head, and two regression heads (damage location and stiffness reduction); (iii) loss functions— $L = L_{\text{data}} + L_{\text{physics}} + 0.1 \cdot L_{\text{bc}}$, where L_{data} is the mean-squared displacement residual, L_{physics} is the mean-squared equation-of-motion residual of Equation (1) at 2000 collocation points, and L_{bc} penalizes soil-spring and end-rotational boundary violations; and (iv) training settings—Adam ($\text{lr} = 10^{-3}$, 5000 epochs) followed by L-BFGS refinement (1000 iterations), on the identical train/test protocol of Section 4.3, averaged over the same five folds. All baselines (DAE, CNN, HDL) were re-implemented and re-trained on the identical dataset and protocol.

6.5. Limitations and Future Work

A fundamental limitation of the current validation protocol is that training and testing windows are extracted from the same finite-element simulation per structural condition. While the temporal separation (0–40 s for training/validation; 40–50 s for testing) ensures that no identical time samples appear in both sets, it does not establish generalization to independent simulation runs with different mesh realizations, perturbed material parameters, or alternative boundary condition implementations. The shared numerical discretization,

damping model, and soil-spring configuration between training and testing windows mean that the model may exploit simulation-specific artifacts (e.g., numerical dispersion characteristics, mesh mode shapes) rather than learning robust, transferable damage signatures. Consequently, the reported 99.3% accuracy reflects the model's ability to identify predefined structural conditions within a consistent numerical environment, not its readiness for field deployment on independent physical measurements.

Future work will address these limitations by (i) generating independent FE runs with $\pm 5\%$ perturbed soil stiffness and $\pm 3\%$ concrete modulus to validate cross-simulation generalization; (ii) constructing a denser parametric training grid (e.g., 20 m location spacing and 2% severity increments) to enable true continuous damage inversion; and (iii) performing experimental validation using instrumented tunnel segments under controlled moving load conditions.

7. Conclusions

This work introduces a Physics-Informed Hybrid Deep Learning (PI-HDL) framework for SHM of metro tunnel structures. The feature extraction capabilities of DAE and CNNs have been integrated, while embedding physics-inspired constraints derived from structural dynamics and soil–structure interactions.

Key contributions:

1. The DAE component learns latent features that satisfy physics-inspired smoothness constraints consistent with wave propagation characteristics, ensuring physically consistent signal representations through physics-regularized representation learning.
2. The CNN is trained with a composite loss function penalizing predictions inconsistent with governing structural dynamics equations, including stiffness–frequency relationships and moving load dispersion characteristics by means of physics-informed classification.
3. The physics constraints have enabled accurate damage detection of predefined damage conditions with limited training data and led to robust closed-set performance with simulated operational variations (train speeds, noise levels).
4. The model provides physics-based explanations for predictions through embedded physical constraints, addressing the “black-box” limitation of conventional deep learning SHM methods.
5. The approach is validated on comprehensive numerical simulations of a metro tunnel under moving train loads, demonstrating superior performance compared to purely data-driven and purely physics-based alternatives.

The PI-HDL framework is particularly suited where safety requirements demand interpretable and trustworthy predictions. However, the current study is validated using high-fidelity numerical simulations only. Future work will explore extension to nonlinear soil behavior and concrete damage plasticity, and experimental validation on laboratory-scale or operational metro tunnels is required before field deployment. Open-set generalization to unseen damage locations and severities remains an important direction for future research.

Author Contributions: Conceptualization, O.A. and A.R.R.; methodology, A.R.R.; software, A.R.R.; validation, O.A., and A.R.R.; formal analysis, O.A.; investigation, A.R.R.; resources, A.R.R.; data curation, O.A.; writing—original draft preparation, A.R.R.; writing—review and editing, O.A.; visualization, A.R.R.; supervision, O.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The processed datasets, together with the Python implementation of the PI-HDL algorithm, are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the College of Law and College of Engineering, University Diyala, Iraq for its support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chang, P.C.; Flatau, A.; Liu, S.C. Review paper: Health monitoring of civil infrastructure. *Struct. Health Monit.* **2003**, *2*, 257–267. [\[CrossRef\]](#)
2. Wang, D.; Xiang, W.; Zhu, H. Damage identification in beam type structures based on statistical moment using a two step method. *J. Sound Vib.* **2014**, *333*, 745–760. [\[CrossRef\]](#)
3. Nair, K.K.; Kiremidjian, A.S.; Law, K.H. Time series-based damage detection and localization algorithm with application to the ASCE benchmark structure. *J. Sound Vib.* **2006**, *291*, 349–368. [\[CrossRef\]](#)
4. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [\[CrossRef\]](#)
5. Pathirage, C.S.N.; Li, J.; Li, L.; Hao, H.; Liu, W.; Ni, P. Structural damage identification based on autoencoder neural networks and deep learning. *Eng. Struct.* **2018**, *172*, 13–28. [\[CrossRef\]](#)
6. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [\[CrossRef\]](#)
7. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
8. Al-Adly, A.I.; Kripakaran, P. Physics-informed neural networks for structural health monitoring: A case study for Kirchhoff–Love plates. *Data-Centric Eng.* **2024**, *5*, e6. [\[CrossRef\]](#)
9. Zhou, W.; Xu, Y.F. Damage identification for plate structures using physics-informed neural networks. *Mech. Syst. Signal Process.* **2024**, *209*, 111111. [\[CrossRef\]](#)
10. Yuan, L.; Ni, Y.-Q.; Rui, E.-Z.; Zhang, W. Structural damage inverse detection from noisy vibration measurement with physics-informed neural networks. *J. Phys. Conf. Ser.* **2024**, *2647*, 192013. [\[CrossRef\]](#)
11. Li, Y.F.; He, W.Y.; Ren, W.X.; Shao, Y.H. Moving load induced dynamic response analysis of bridge based on physics-informed neural network. *Adv. Eng. Inform.* **2025**, *65*, 103215. [\[CrossRef\]](#)
12. Tao, Y.; Chen, W.H.; Li, Z.B.; He, W.Y. Dynamic displacement reconstruction of bridge based on physics-informed recurrent neural network. *Adv. Bridge Eng.* **2025**, *6*, 12. [\[CrossRef\]](#)
13. Xu, X.; Liu, C. Physics-guided deep learning for damage detection in CFRP composite structures. *Compos. Struct.* **2024**, *331*, 117889. [\[CrossRef\]](#)
14. Nhung, T.C.; Nguyen, H.B.; Minh, T.Q. Reconstructing Health Monitoring Data of Railway Truss Bridges using One-dimensional Convolutional Neural Networks. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 15510–15514. [\[CrossRef\]](#)
15. Alazzawi, O.; Wang, D. Health monitoring of an assembly-type structure using raw time-domain EMI signals and a novel hybrid deep learning algorithm. *Smart Mater. Struct.* **2024**, *33*, 025012. [\[CrossRef\]](#)
16. Hetenyi, M. *Beams on Elastic Foundation: Theory with Applications in the Fields of Civil and Mechanical Engineering*; University of Michigan Press: Ann Arbor, MI, USA, 1946.
17. Wang, S.; Li, J.; Luo, H.; Zhu, H. Damage identification in underground tunnel structures with wavelet based residual force vector. *Eng. Struct.* **2019**, *178*, 506–520. [\[CrossRef\]](#)
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1026–1034.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.