Computers and Structures 267 (2022) 106798

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Physics-informed deep neural networks for simulating S-shaped steel dampers

Yao Hu [a,b,c], Wei Guo [a,b,*], Yan Long [a,b], Shu Li [a,b], Zi'an Xu [a,b]

[a] School of Civil Engineering, Central South University, Changsha 410075, China
[b] National Engineering Research Center of High-speed Railway Construction Technology, Changsha 410075, China
[c] Department of Infrastructural Engineering, The University of Melbourne, Parkville, VIC 3010, Australia

ARTICLE INFO

Article history:
Received 5 December 2021
Accepted 28 March 2022
Available online 14 April 2022

Keywords:
S-shaped steel dampers
Hysteretic behaviors
physics-informed DNNs
RNNs
LSTMs

ABSTRACT

Numerical simulation that combines finite element methods and experimental data has been recognized as effective in modeling hysteretic behaviors and capturing the principle mechanical trend of passive energy dissipation devices. However, the seismic design and mechanical characteristics assessment for passive energy dissipation devices require a laborious effort and massive computational resources to tune mechanical-oriented parameters. Particularly, the potential risk of departure from the actual system is rising for the desirable seismic design of dampers due to the numerical model simplification and assumption. To eliminate the potential weakness of numerical models, this paper explores a surrogate model by implementing physics-informed deep neural networks (DNNs) to approximate hysteretic behaviors of S-shaped steel dampers. The proposed physics-informed DNNs mainly consists of recurrent neural networks (RNNs) and long short-term networks (LSTMs), which can encode the Bouc-Wen model into the direct graph and incorporate the effect of design-oriented geometry parameters. To validate the generality of the network, the optimization model was calibrated numerically and experimentally, respectively, which exhibits good performance in predicting nonlinear behaviors of different dampers with reasonable accuracy. The proposed physics-informed DNNs can be an alternative to relieve the laboriousness of the seismic design and mechanical characteristics assessment of passive energy dissipation devices.

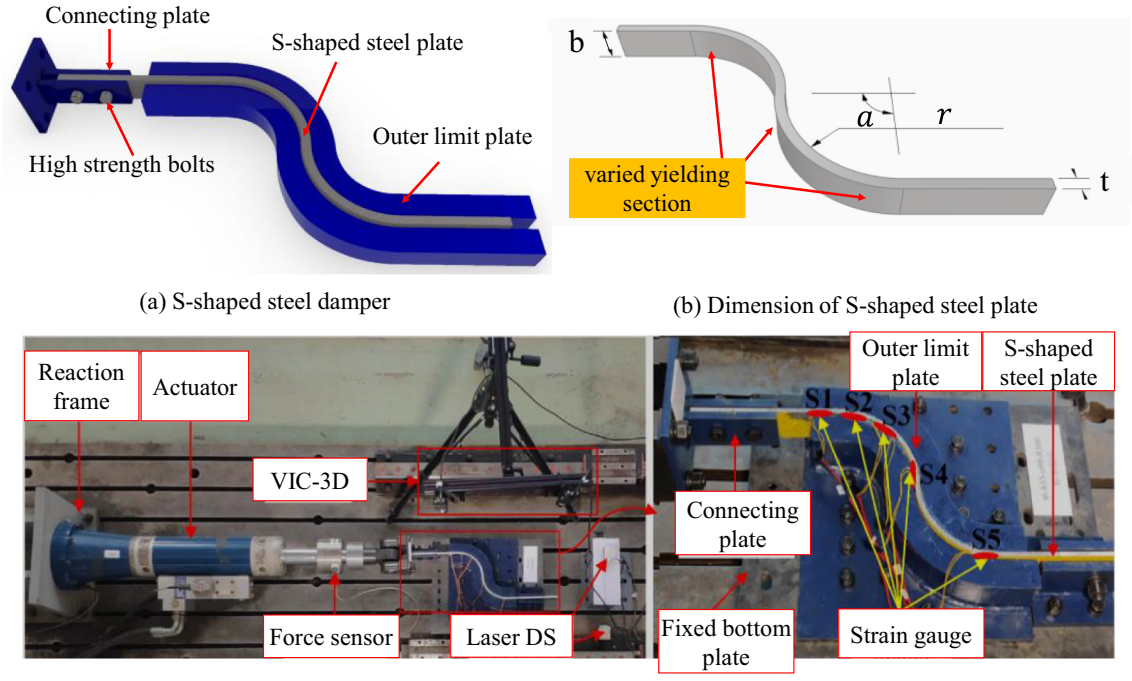
1. Introduction

Passive energy dissipation devices have been widely utilized to control the dynamic vibration and enhance the seismic performance of building structures [1–4]. The numerical simulation is recognized as effective for the desirable seismic design of dampers and supplementary parametric studies on the potential mechanical characteristics [5–6]. The common practice starts with developing numerical models calibrated by the experimental data within reasonable accuracy, comes along with the repeated tune-up of mechanical-oriented parameters such as stiffness, yield ratio, and strength by trial and error [7]. To satisfy desirable seismic goals, dampers need to be redesigned by performing a similar onerous tune-up of mechanical-oriented parameters. The geometric-oriented parameter is obtained based on the correlation that relies extensively on the regression analysis between physical and mechanical parameters. Additionally, short of the experimental data, there is a growing concern about the inherent departure resulting from the numerical model simplification and assumptions in the comparison of the actual systems.

Recently, the successful application of deep neural networks (DNNs) in engineerings such as the seismic response prediction of structures [8–10], seismic vulnerability assessment of buildings [11], and system identification and structural health monitoring [12–13] has motivated the investigation of the surrogate model with high fidelity driven by massive datasets. The data-driven DNNs, which can learn the potential pattern by using the pure multi-layers to map the input–output relationship, have demonstrated impressive power and promise in solving regression and classification issues. Such networks are successful in forecasting the damping force of magnetorheological (MR) fluid dampers and determining the optimal mechanical properties of tuned mass dampers (TMD) to reduce the displacement responses of building under different wind loads [14–15]. The limitation, however, has become prominent when applying data-driven neural networks in practical engineering due to the difficulty in meeting the demand of massive datasets and the lack of interpretable physical meaning [16]. Consequently, physics-informed DNNs have been developed to incorporate the laws of physics such as the partial dif-

* Corresponding author.
E-mail address: guowei@csu.edu.cn (W. Guo).

https://doi.org/10.1016/j.compstruc.2022.106798
0045-7949/



(a) S-shaped steel damper

(b) Dimension of S-shaped steel plate

(c) Experimental setup

Fig. 1. S-shaped steel dampers and experimental setup.

Table 1
Geometry parameters of S-shaped steel dampers.

No.	Specimens	Thickness t (mm)	Breadth b (mm)	Angle α (°)	Radius r (mm)
1	t6-b55-α 90-R100	6	55	90	100
2	t8-b55-α 90-R100	8	55	90	100
3	t10-b55-α 90-R100	10	55	90	100
4	t8-b50-α 90-R100	8	50	90	100
5	t8-b60-α 90-R100	8	60	90	100
6	t8-b55-α 45-R100	8	55	45	100
7	t8-b55-α 45-R150	8	55	45	150
8	t8-b55-α 45-R200	8	55	45	200
9	t8-b55-α6 0-R100	8	55	60	100

ferential equations (PDEs) into direct graphs [17–19], which has been implemented in the application of power systems [20], cumulative damage models [21], elasticity problems accounting for strain gradient effects [22], the nonlinear structural systems [23]. The achievement of PDEs in DNNs makes it available to derive the surrogate model in solving engineering problems with good accuracy and robustness by training deep neural networks driven by different physics-informed loss functions. For instance, Guo et, al. developed a loss function that governs PDFs of Kirchhoff plate bending problems for approximating the continuous transversal deflection [24]. Duchanoy et. al. applied the physics-informed DNNs to capture the hysteretic responses of a set of small-scale magnetorheological dampers (MRD) with physical parameters under different load conditions and various physical configurations [25]. Yucesan et. al. used the physics-informed DNNs to adjust the output of stiffness and damping coefficient of the torsional vibration dampers (TVD) to experimental data, demonstrating good effectiveness in eliminating the discrepancy induced by numerical simulations [26].

Inspired by this, we implemented a physics-informed DNN architecture to approximate the hysteretic behaviors of a set of novel S-shaped steel dampers designed by authors that can deform by achieving varied cross-section yielding and prevent the occurrence of unpredictable crack induced by the concentration of stress and plasticity [27–30]. The physics-informed DNNs are able to encode the Bouc-Wen model into the direct graph and forecast hysteretic behaviors by covering the effect of the design-oriented geometry parameters, which can effectively relieve the laboriousness of the seismic design and mechanical characteristics assessment of passive energy dissipation devices. The model generality is validated by the dataset provided by both the 3D finite element (FE) models and experimental tests, respectively. The optimization model demonstrates good performance in forecasting the nonlinear behaviors of S-shaped steel dampers with different geometry parameters.

The structure of the paper is organized as follows: Section 2 introduces the S-shaped steel dampers and compares the numerical results with the experimental data. In Section 3, a physics-informed DNN architecture is proposed and presented. Section 4 illustrates how different hyperparameters are configurated based on the grid search and experience knowledge from the previous research. Case studies are presented for the validation of proposed physics-informed based on numerical and experimental datasets in Section 5. Finally, Section 6 summarizes the conclusions.

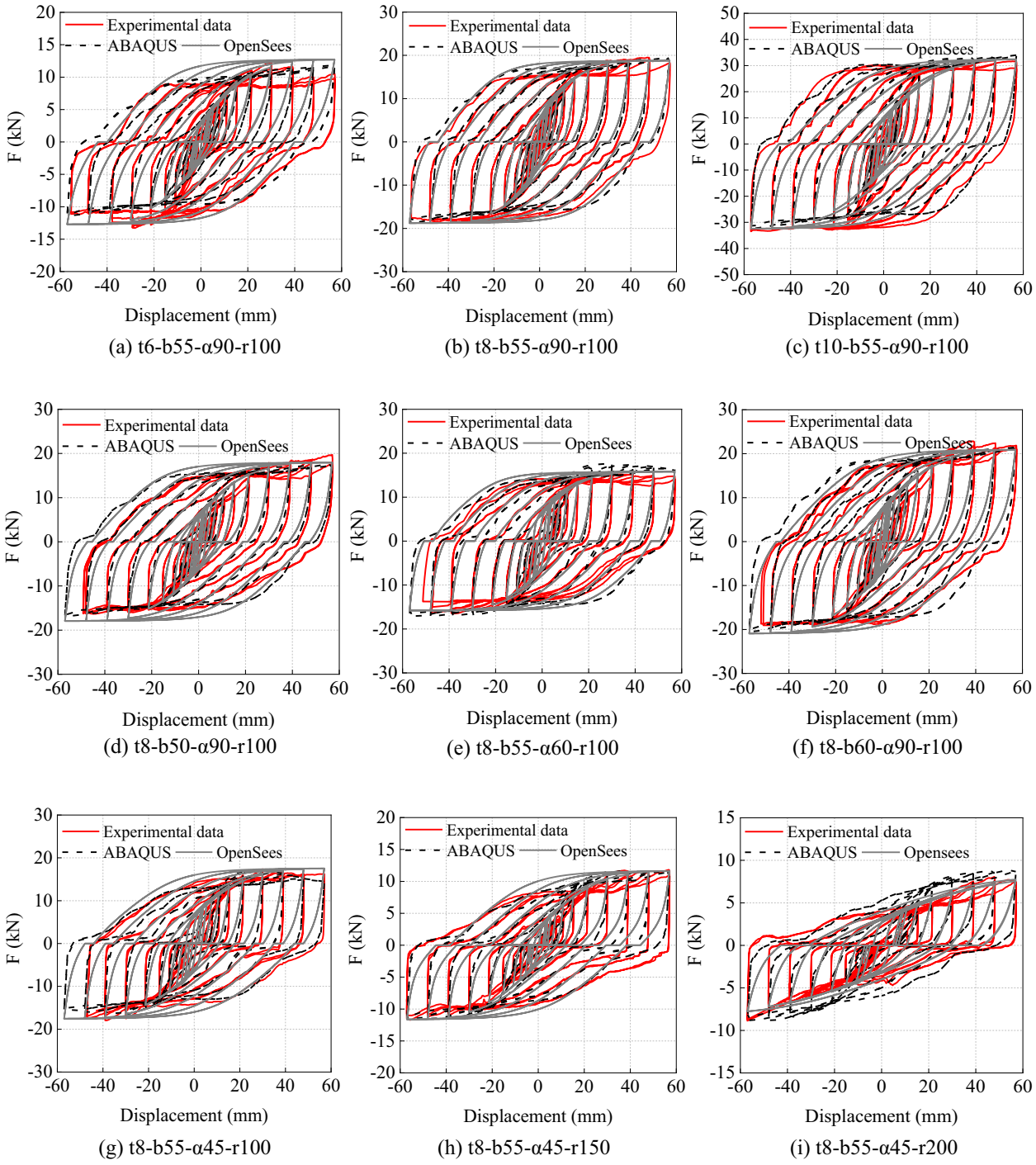


Fig. 2. Comparisons of numerical and experimental data.

2. The proposed novel S-shaped steel damper

2.1. S-shaped steel dampers and experimental setup

The proposed novel S-shaped steel damper designed by the principle of varied yielding cross-sections consists of a S-shaped steel plate, outer limit plate, fixed bottom plate, connecting plate, and high-strength bolts, as shown in Fig. 1(a). Restrained by the outer limit plate, one end of the S-shaped steel plate is linked to the connecting plate by the high-strength bolts while another end is set as free. The S-shaped steel plate can dissipate the external seismic energy at the sacrifice of varied cross-sections yield during the earthquakes. Flexural plastic deformation is developed along with the curvature of sections, as illustrated in Fig. 1(b). With the characteristics of varied yielding cross-sections, the S-shaped steel damper is found effective in eliminating the weakness of traditional metal dampers such as stress and plastic deformation concentration.

To investigate the hysteretic behaviors of S-shaped steel dampers, a total of nine specimens with different combinations of the thickness (t), breadth (b), angle (α), radius (r) was tested under the standard cyclic loading, as shown in Table 1. Each specimen is named after the geometry parameter such as t6-b50-α 90-R100, which indicates that the S-shaped steel damper has a thick-

ness of 8 mm, breadth of 55 mm, angle of 90-degree, and a radius of 100 mm for the flexure section. All specimens were fabricated by the Q235B steel with a nominal yielding strength of 235 Mpa. The experimental test on specimens was conducted in the National Engineering Laboratory of Central South University, see Fig. 1(c). The S-shaped steel damper was connected to the actuator by the grade 10.9 high strength bolts with a nominal yielding strength of 900 Mpa. The external force applied to specimens was monitored by the force sensor, and the displacement of specimens was monitored by three laser sensors. More experimental details can be found in the literature [31].

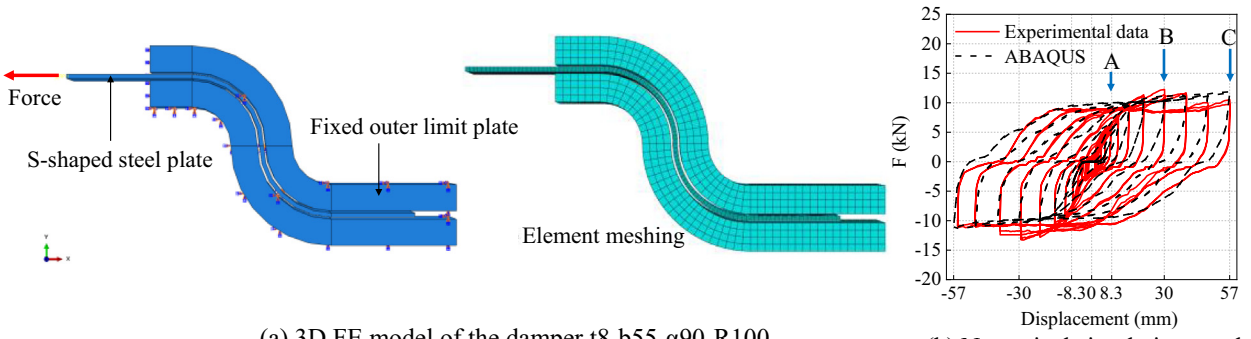
2.2. Numerical modeling based on experimental results

Fig. 2 presents the hysteretic curves of nine specimens of S-shaped dampers under the standard cyclic loading. With the propagation of displacement loading, all specimens yield at a relatively small displacement and gradually reach the peak strength. No obvious failure occurs to specimens of S-shaped steel dampers during the test, indicating that the S-shaped steel dampers have good energy dissipation capacity.

To further investigate the mechanical properties of S-shaped dampers, numerical models by the FE software ABAQUS and OpenSees were separately developed and calibrated by the experimental data. The cycle hardening material in ABAQUS that incorporates the kinematic and isotropic cyclic hardening behavior can achieve superior accuracy for capturing mechanical properties of steel plates [31,32], which was adopted for the 3D FE model of S-shaped steel dampers. The parameters that need to be determined are listed in Table 2, in which, σ_0 is the yield stress at the start point of plastic strain; C_k and γ_k are the coefficients of the kinematic hardening stress. The outer limit plate was simulated by the elastic material without considering the damage. 3D-stress 8-node nonlinear solid element (C3D8R) in ABAQUS was used to

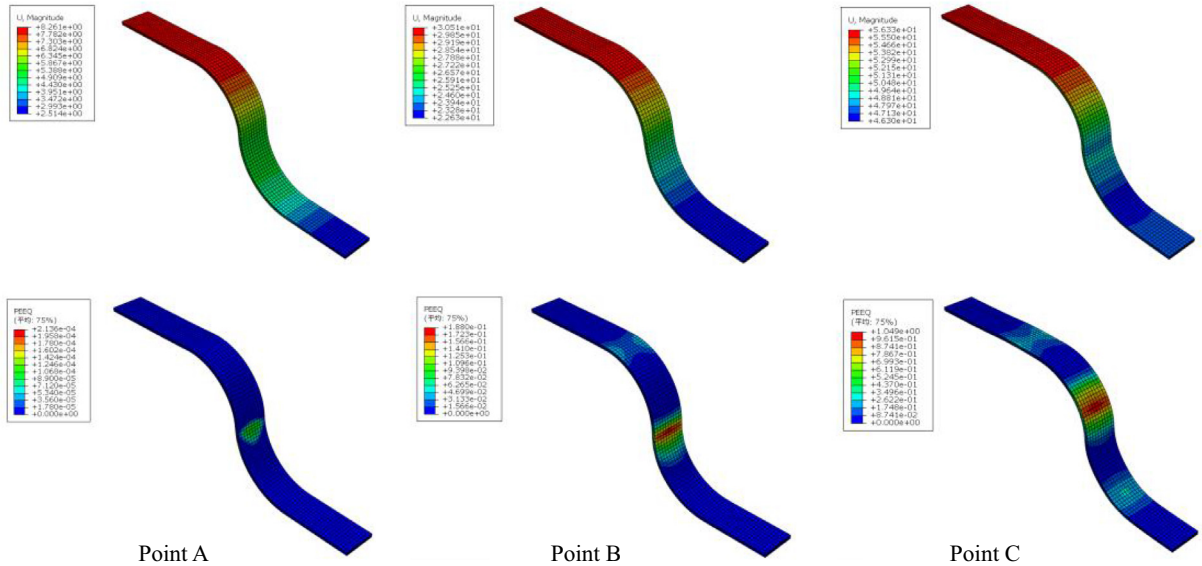
Table 2
Parameters of cycle hardening in ABAQUS.

σ_0(Mpa)	C_1(Mpa)	γ_1	C_2(Mpa)	γ_2	C_3(Mpa)	γ_3	C_1(Mpa)	γ_4
307	22,000	300	1600	120	6000	250	600	30



(a) 3D FE model of the damper t8-b55-α90-R100

(b) Numerical simulation result



(c) The deformation and PEEQ distribution of S-shaped steel plate at point A,B,C

Fig. 3. FE model and simulation results of S-shaped steel dampers developed by ABAQUS.

Table 3
Parameters of steel4 material in OpenSees.

f_y	E_0	b_k	R_0	r_1	r_2	b_i	b_l	rho_i	R_i	l_yp
F_y	k_e	k_p/f_e	15	0.90	0.15	0.0	0.0	$(F_{max}-F_y)/F_y$	5	0.0

model the S-shaped steel dampers and outer limit plate. The mesh size of models was divided by four elements along with plate thickness and accounts for 10% of the width of the steel plates, which can achieve a good trade-off between computational accuracy and efficiency, as shown in Fig. 3(a). The trend in the distribution of the PEEQ (equivalent plastic strain) and displacement (U) in Fig. 2(b-c) reflects the characteristics of variable section yield of the S-shaped steel dampers along with the propagation of standard cyclic loading. Similarly, Steel4 material with combined kinematic and isotropic hardening behaviors in OpenSees was adopted to simulate the force–displacement relationship of S-shaped steel dampers by repeatedly adjusting the mechanical parameters listed in Table 3. The definition can refer to the manual [33]. TwoNodeLink element was used to model the S-shaped steel dampers. Fig. 2 compares numerical results obtained by ABAQUS and OpenSees with the experimental data. Due to the model simplification, a larger discrepancy between numerical models by OpenSees and experimental results is observed as compared to the 3D FE model from ABAQUS. The gray lines that represent the numerical model by OpenSees exhibit a higher maximum force and faster strength degradation, indicating a potential departure between the numerical model and the actual system. The concern about such departure can result in erroneous results for the desirable seismic design of building-damper systems, since the design-oriented geometry parameters of the dampers may not correlate well with mechanical-oriented parameters. In addition to the inherent discrepancy between the numerical model and actual system, it is time-consuming for either the tune-up of the numerical model in OpenSees or constructing a 3D model in ABAQUS.

To eliminate the weakness of numerical models, a physics-informed DNN framework was proposed to explore a surrogate model that is able to simulate the hysteretic behaviors of dampers by directly incorporating the effect of design-oriented geometry parameters, which will be introduced in the following section.

3. Physics-informed DNNs

3.1. Bouc-Wen model

The Bouc-Wen model uses differential equations to describe hysteretic characteristics and is widely utilized in approximating the nonlinear behaviors of passive energy dissipation devices [34–36]. However, the numerical simulation based on the Bouc-Wen model requires laborious tune-up of mechanical-oriented parameters and shape parameters that control the hysteretic behaviors. The restoring force F is expressed as a format of Eqs. (1)–(2).

$$F = F_e + F_h = a_0 kx + (1-a_0)kz \tag{1}$$

$$\dot{z} = \beta_1 \dot{x}(t) - \beta_2 \dot{x}(t)|z(t)|^n - \beta_3 |\dot{x}(t)|z(t)|z(t)|^{n-1} \tag{2}$$

In which F_e and F_h are the elastic and hysteretic components of the restoring force, respectively; k is the stiffness coefficient; a_0 is the ratio of post-yield stiffness to the initial tangent stiffness;z is the hysteretic displacement represented by a set of differential equations; x and $\dot{x}$ are the horizontal displacement and velocity of dampers, respectively, and β_1, β_2, β_3, n are shape parameters controlling the hysteric behaviors of the Bouc-Wen model.

To have a better understanding of the surrogate model, the Bouc-Wen model will be encoded into the direct graph of the physics-informed DNNs to approximate the hysteretic behaviors of S-shaped steel dampers. The nonlinear behavior of the Bouc-Wen model is mainly determined by the hysteretic parameter z.

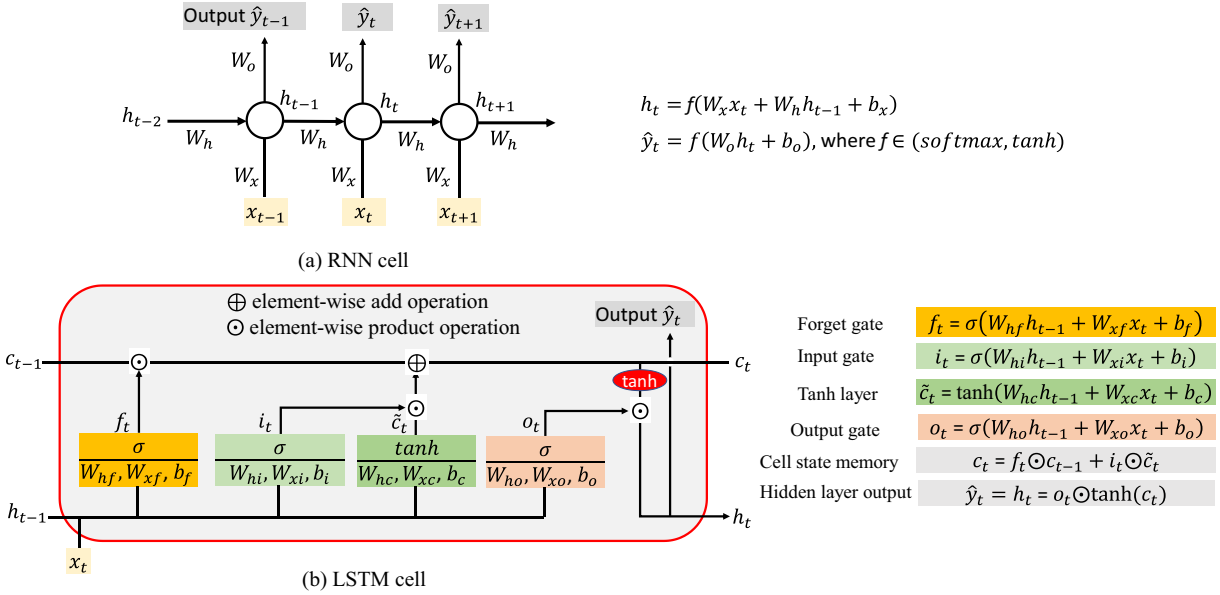


Fig. 4. Graphical representation of RNN cell and LSTM cell.

However, z is a non-measurable parameter, which is intractable to be extracted from the experiment. Previous studies on the parameter identification of the Bouc-Wen model have shown that z can be approximated using the linear least square method [37–38] using the following equations.

$$\int_{t_0}^{t_i}\dot{z}(t)dt=\beta_1\int_{t_0}^{t_i}\dot{x}(t)dt-\beta_2\int_{t_0}^{t_i}\dot{x}(t)|z(t)|^n dt-\beta_3\times\int_{t_0}^{t_i}|\dot{x}(t)|z(t)|z(t)|^{n-1}dt \tag{3}$$

$$\text{Letting } D=\int_{t_{i-1}}^{t_i}\dot{z}(t)dt=z(t_i)-z(t_0) \tag{4}$$

$$A_i=\dot{x}(t_i)|z(t_i)|^n,\quad B_i=|\left(\dot{x}(t_i)|z(t_i)|z(t_i)|^{n-1}\right. \tag{5}$$

The velocity $\dot{x}$ can be obtained based on the central difference method, which is expressed as:

$$\dot{x}_i=(x_{i+1}-x_{i-1})/2\Delta t \tag{6}$$

$$I_1=\int_{t_0}^{t_i}\dot{x}(t)dt=x(t_i)-x(t_0) \tag{7a}$$

$$I_2=-\int_{t_0}^{t_i}\dot{x}(t)|z(t)|^n dt=-\frac{1}{2}\sum_{i=0}^{i-1}(A_i+A_{i+1})\Delta t \tag{7b}$$

$$I_3=-\int_{t_0}^{t_i}|\dot{x}(t)|z(t)|z(t)|^{n-1}dt=-\frac{1}{2}\sum_{i=0}^{i-1}(B_i+B_{i+1})\Delta t \tag{7c}$$

$$D=\beta_1 I_1+\beta_2 I_2+\beta_3 I_3,\ Z_i=Z_0+D \tag{8}$$

It can be seen in Eq. (8) that the hysteretic variable D is a function of shape parameters represented as a form of the matrix $\{\theta\}=\{\beta_1,\beta_2,\beta_3\}^T$. To obtain the hysteretic variable D, the shape parameter n with regard to I_2 and I_3 can be predetermined as the sensitivity analysis [37] on shape parameters has shown that the integer n mainly affects the smoothness of hysteretic curves and has a minor influence on the restoring force value. Name $I_i=\{I_1,I_2,I_3\}^T$, and the hysteric variable D_i at the time i can be estimated on a basis of the linear least square method:

$$MinE(\theta)=\sum_{i=1}^{m}\left(I_i^T\theta-D_i\right)^2 \tag{9}$$

The approximation equations above have been successfully applied to parameter identification of the Bouc-Wen model for mild steel dampers and rubber bearing [38]. With the advance in deep neural networks, the success in incorporating nonlinear partial differential equations in DNNs makes it possible to facilitate the potential feature learning and nonlinear regression [17–18,39]. The boost in physics-informed DNNs offers an opportunity to explore a surrogate model that can capture the nonlinear behaviors of metallic dampers with satisfactory accuracy and computational efficiency. To investigate this, a physics-informed neural network based on the Bouc-Wen model is constructed in the next section, which is expected to model the nonlinear behaviors of S-shaped steel dampers.

3.2. Physics-informed neural network architecture

The recurrent neural networks (RNN) are well known to deal with time-varying and sequential data [40], as illustrated in Fig. 4(a). Feathered by the feedback loops with recurrent connections, the RNN possesses an impressive capability of processing sequential data by transferring information from the last hidden state to the next state. The internal state h_t can be dynamically updated by the flow of information at the previous time h_{t-1} and the current input information x_t. The output $\hat{y}_t$ of the RNN cell can be regularized by activation functions such as tanh and Soft-Max units, which endows the RNN with the capacity of learning nonlinear features. However, due to the simple form, the RNN commonly comes with the possible gradient vanishing and exploding issues, making it incapable of coping with long-term sequence issues. To overcome such weakness, the long short-term network (LSTM), as a variant of RNN, has been developed to solve long-term dependence problems [41], as shown in Fig. 4(b). Each LSTM cell is composed of an internal cell c_t and three gates including a forget gate f_t, an input gate i_t and an output gate o_t. The state of internal cell c_t relies primarily on the forget gate and input gate. The forget gate updates the internal cell state by deciding what information should be forgotten or retained; the input gate controls the flow of input information into the current cell state via a tanh layer $\tilde{c}_t$; the output gate determines the flow of information into the output of the LSTM cell. The three gates are all regularized via the sigmoid function ranging from 0 to 1, reflecting the different degrees of information retention. Finally, the information of c_t is updated by the element-wise product of forget gate and the cell state at the previous time step c_{t-1} and supplemented with added information from the input gate via a tanh layer; the output $\hat{y}_t$ and hidden layer output h_t of LSTM cell are obtained by the element-wise product of the output gate o_t and the updated cell state c_t through a tanh layer, in which $x_t\in R^m$ and $h_t\in R^n$ are the input and hidden state at the time step;$\{W_x,W_{xf},W_{xi},W_{xc},W_{xo}\}\in R^n\times mand\{W_h,W_o,W_{hf},W_{hi},W_{hc},W_{ho}\}\in R^{n\times n}$ denote the weight matrix for the current input and recurrent input, respectively, and $\{b_x,b_f,b_i,b_c,b_o\}\in R^n$ represents the bias vector of neurons.

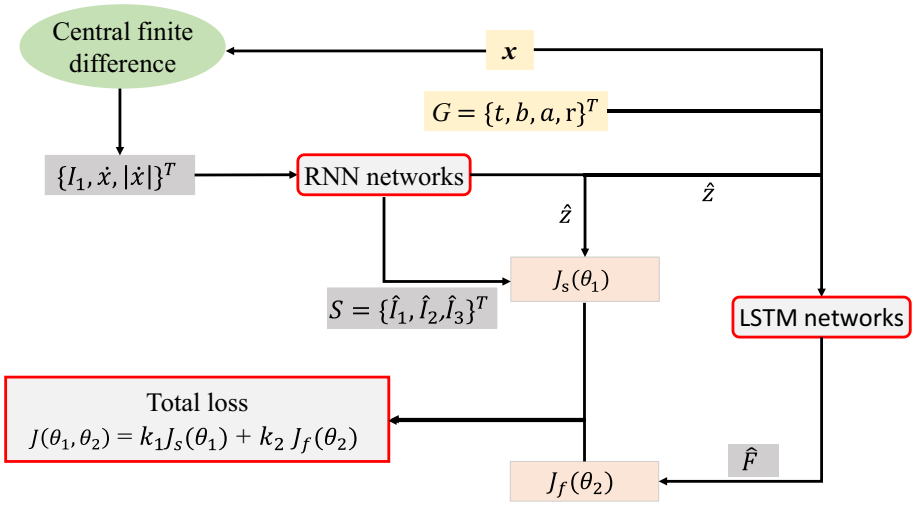
Physics-informed neural network architecture is constructed as shown in Fig. 5, consisting of RNNs and LSTM networks and a tensor differentiator based on the central finite-difference method. The available physics law based on the Bouc-Wen model in Section 3.1 is encoded into the direct graph to drive the training process. Firstly, the tensor differentiator based on central finite difference is used to obtain the velocity $\dot{x}$ as the input data based on Eq. (6). The RNNs are used to develop the nonlinear mapping from the input data $\mathrm{d}=\{I_1,\dot{x},|\dot{x}|\}^T$ to the state space variables $\mathrm{S}=\left\{\widehat{I}_1,\widehat{I}_2,\widehat{I}_3\right\}^T$, and then map S to z, as expressed in Eq. (10). The physics loss function of RNNs based on Mean Squared Error (MSE) can be written as a format of Eq. (11).

$$z=f_{RNNs}(S;W_{R2},b_{R2}),S=f_{RNNs}(d;W_{R1},b_{R1}) \tag{10}$$

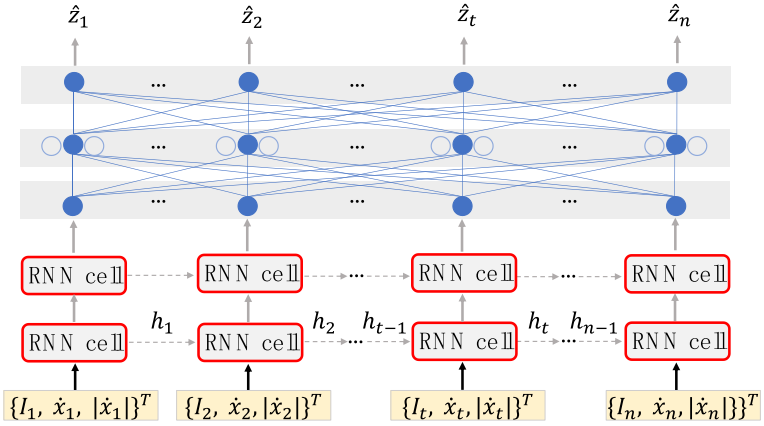
$$J_s(\theta_1)=\sum_{i=1}^{n_m}||\widehat{I}_1-I_1||_2^2+\sum_{i=1}^{n_m}||\widehat{I}_2-I_2(z,\dot{x})||_2^2+\sum_{i=1}^{n_m}||\widehat{I}_3-I_3(z,\dot{x})||_2^2 \tag{11}$$

In which $\theta_1=\{W_{R1},b_{R1},W_{R2},b_{R2}\}$, W_{R1},W_{R2} and b_{R1},b_{R2} denote the trainable weights and bias parameters of the RNNs, respectively. n_m is the number of data samples, all variables have the same number of n sample points ranging from t_1 to t_n;I_1, I_2 and I_3 can be obtained by Eq. (7).

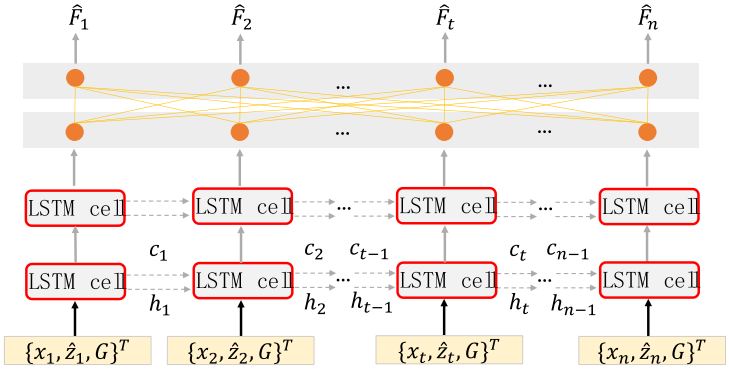
A sacked input layer with the geometry parameters $G=\{\mathrm{t},\mathrm{b},a,\mathrm{r}\}^T$, the displacement x, and hysteretic displacement z as the input data is used to map the external force $\widehat{F}$, which is achieved by the deep LSTM networks. The data loss function is expressed as follows:



(a) Physics-informed deep neural networks



(b) RNNs



(c) LSTM networks

Fig. 5. The physics-informed DNN architecture.

$$\widehat{F} = f_{LSTMs}(x, \hat{z}, G; \omega_L, b_L) \tag{12}$$

$$J_f(\theta_2) = \sum_{i=1}^{n_m} ||\widehat{F} - F_{true}||_2^2 \tag{13}$$

in which $\theta_2 = \{b_L, W_L\}$ indicates the trainable weights and bias parameters for the LSTM networks; The total loss function that combines the data loss and physics loss by different weight coefficients is minimized to achieve the training process of the physics-informed DNNs, as shown in Eq. (14)–(15).

Table 4
The selection of hyperparameters for the physics-informed DNNs.

Hyperparameter tuning for model training and network design			
Hyperparameter for training	Adam/RMSprop	Learning rate	Weight coefficient (k_1, k_2)
RNNs	Hidden layers and Neutrons for each layer	Activation function	Dropout ratio
LSTM networks	Hidden layers and Neutrons for each layer	Activation function	Dropout ratio

$$J(\theta_1, \theta_2) = k_1 J_s(\theta_1) + k_2 J_f(\theta_2) \tag{14}$$

$$\left\{\widehat{\theta}_1, \widehat{\theta}_2\right\} = argminJ(\theta_1, \theta_2) \tag{15}$$

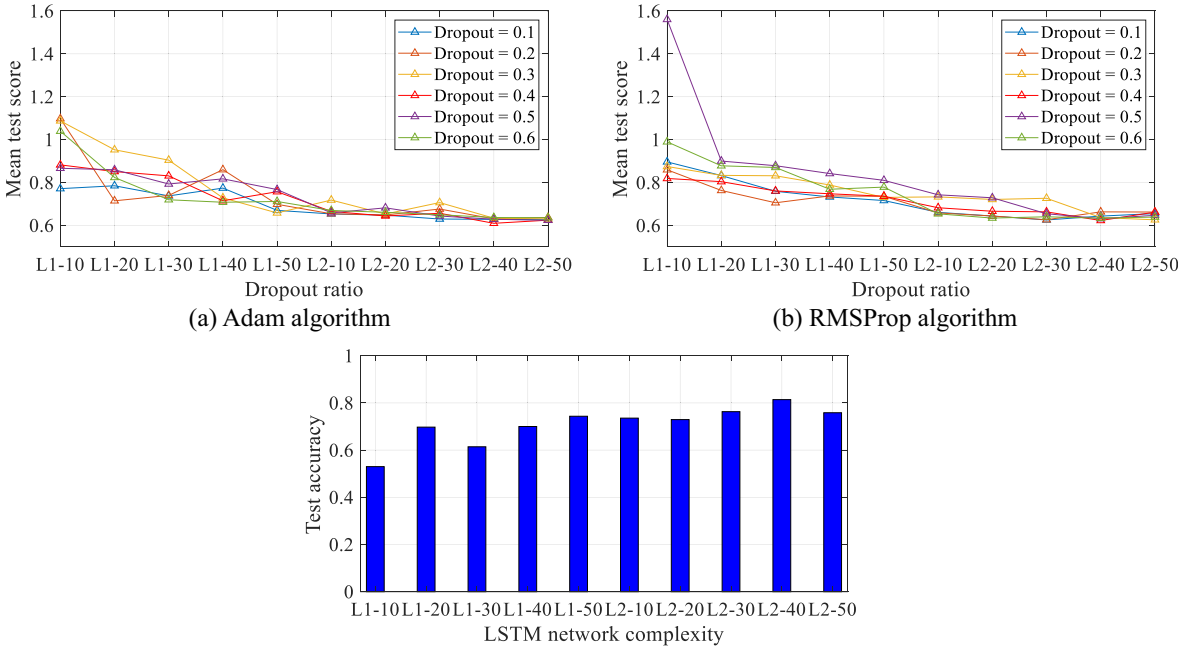
where k_1, k_2 are the user-defined weight coefficients for controlling the update of model, and the selection of the two hyperparameters will be discussed in the following section.

4. Hyperparameter tuning

4.1. Grid search

To optimize the physics-informed DNNs, hyperparameters that are used for model training and network design in Table 4 need to be tuned and determined. Hyperparameter tuning is a grand challenge since there is no explicit rule of thumb on how to well configure different interacting hyperparameters. Grid search can be regarded as a heuristic optimization method in setting different combinations of different hyperparameters[42]. Although grid search performs an exhaustive search on the hyperparameter sets specified by users and is limited to a small-scale dataset, it is applicable for selecting hyperparameters of the proposed physics-informed DNNs. Considering the computational resource, the strategy is using grid search to explore the best configurations of more important hyperparameters in the model training, and the less important hyperparameters are generally set as default values or based on experience knowledge from the previous studies at the beginning and will be elaborately adjusted after the physics-informed DNNs are devised.

For the model training, much importance should be attached to the selection of optimization algorithms such as stochastic gradient descent [43], AdaGrad [44], RMSProp [45], and Adam [46]. Referring to the previous research, Adam is commonly used as it has the merits of AdaGrad dealing with sparse gradients and the ability of RMSProp to handle non-stationary objectives. Particularly, most default values for hyperparameters of Adam are believed sufficient to achieve reliable and good results on most neural network architectures [47–48]. To validate this, the optimization algorithms that include the Adam and RMSProp are con-



(a) Adam algorithm (b) RMSProp algorithm

(c) Model accuracy of different LSTM network complexity

Fig. 6. Model performance considering different combinations of hyperparameters.

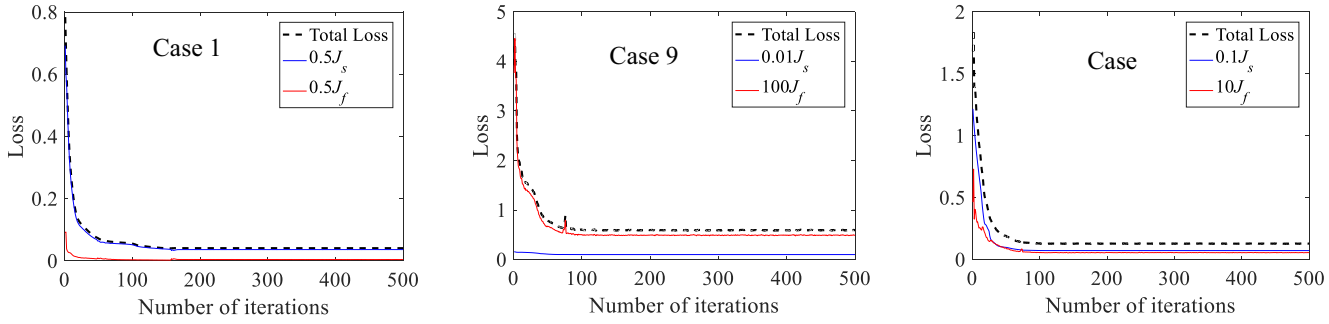


Fig. 7. The influence of different weight coefficients on loss curves.

sidered in the grid search. For the design of neural network structure, choosing appropriate numbers of layers and neutrons is challenging as the physics-informed DNNs are composed of two different networks which both play a critical role in the model performance. From the perspective of computing efficiency and learning capacity, it is observed that RNNs consisting of two RNN layers of ten neurons are sufficient. For brevity, only the number of layers and neurons for LSTM networks are investigated using grid search. Additionally, to eliminate the occurrence of overfitting and enhance the model robustness, a set of dropout ratios that is applied to hidden layers of LSTM networks is involved in the grid search. Therefore, a combination of hyperparameters that include the optimization algorithm, the number of hidden layers and neutron for LSTM networks and the dropout ratio are investigated using grid search as these hyperparameters are more influential in determining the model training and performance. For other hyperparameters, the rectified linear function (ReLU) is used as the activation function as it can effectively eliminate the vanishing of the gradient [49]. The learning rate and weight coefficients will be elaborately tuned once the hyperparameters for controlling the model training and structure is well configurated. Moreover, considering the inherent randomness induced by the initialization, the optimization process is independently performed three times for each hyperparameter tuning.

Table 5
The influence of different weight coefficients on model performance.

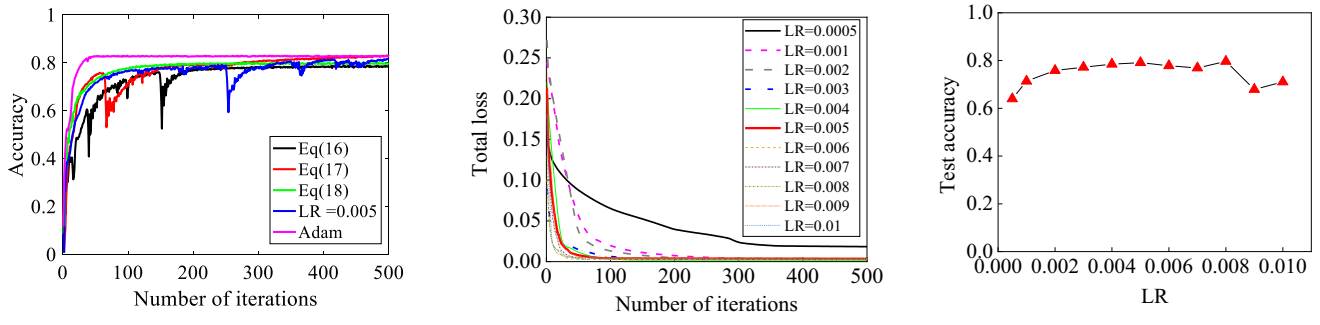
Case studies	Weight coefficient $\boldsymbol{k}_1$ for physical loss $\boldsymbol{J}_s$	Weight coefficient $\boldsymbol{k}_2$ for data loss $\boldsymbol{J}_f$	Model accuracy
case 1	0.5	0.5	0.66
case 2	0.8	0.2	0.73
case 3	0.2	0.8	0.64
case 4	0.7	0.3	0.73
case 5	0.6	0.4	0.74
case 6	1	1	0.74
case 7	10	0.1	0.60
case 8	0	1	0.22
case 9	0.01	100	0.44
case 10	1	10	0.69
case 11	1	20	0.79
case 12	1	100	0.80
case 13	0.5	10	0.82
case 14	0.1	10	0.83
case 15	0.5	20	0.82

4.2. Optimization algorithm and network structure

Fig. 6 (a-b) illustrates the test performance that is quantified by the mean squared error (MSE) for different combinations of hyperparameters by performing the three-fold cross-validation to a set of thirty-five data samples developed by ABAQUS. It can be seen that the Adam algorithm exhibits better test performance compared to the RMSProp algorithm when the network is not deep. As the network becomes larger and wider, the improvement of the algorithm and dropout ratio on model performance gradually reaches saturation. This indicates that the deep neural network has sufficient capacity in capturing the potential feature, and the increase of layers and neutrons for the network is not helpful in enhancing the model performance. Referring to the evaluation results based on grid search, the Adam algorithm is adopted for the model training. Fig. 6(c) compares the model performance of different LSTM network structures on the test dataset, demonstrating that LSTM networks have two hidden LSTM layers of forty neurons is appropriate for the model training.

4.3. Weight coefficient

The physics-informed DNNs are trained by minimizing the total loss function that incorporates the physical loss and data loss. For this multi-task learning problem, different weight coefficients k_i need to be assigned to the physical loss function and data loss function so as to balance the joint learning of the RNNs and LSTM networks, thus avoiding a scenario where one of the tasks dominates the optimization procedure. The most common method in the previous studies related to multi-task learning problems is to set the weight coefficient equally [50]. However, a potential risk is that multi-task learning may suffer from an imbalanced learning rate [51]. A better strategy to tackle such an issue is to make the loss gradient reduce equally thus ensuring the multi-learning tasks are equally updated. To investigate this, a set of different weight coefficients are studied, including (a) the physical loss $k_1 J_s$ dominates the training process for cases 1–7; (b) the data loss $k_2 J_f$ controls the model training cases 8–9; (c) Both physical loss and data loss contribute to the model training with a similar loss magnitude for cases 10–15, as illustrated in Table 5. For brevity, Fig. 7(a-c) presents three case studies on the influence of different weight coefficients on the loss curves for the total loss function, the physical loss and data loss. Apparently, when one of the learning tasks dominates the model training, an imbalanced learning process is observed. By contrast, the balance between the reduction of loss gradient for multi-task learning can effectively enhance the model



(a) Model performance for adaptive learning rate methods

(b) Influence of LR for Adam on total loss

(c) Influence of LR for Adam on model performance

Fig. 8. Influence of different adaptive LR methods and LR.

performance. By comparing the model performance in Table 5, $k_1 = 0.1, k_2 = 10$ is adopted in this paper.

4.4. Learning rate

The learning rate (LR) controls how much the model weights are updated for each iteration. Generally, it is believed that setting too small LR will significantly slow the convergence of the network, and too large value may result in an unstable training process. A preferable strategy is to conduct an adaptive learning rate method in the model training, allowing the network to dynamically reduce the learning rate during the training process so as to improve the model performance. This can be achieved by using either pre-defined learning rate decay or adaptive learning rate methods. The learning rate decay require users to set a LR schedule that changes during the training process, such as the step decay and exponential decay [48], as expressed in Eqs. (16)–(18), in which LR and LR_0 are the updated and initial learning rates, respectively; k_t and k_{total} are the current time steps and the total number of iterations. Alternatively, adaptive learning rate methods can be achieved by selecting adaptive gradient descent algorithms such as AdaGrad, RMSProp and Adam. Among these methods, the adaptive learning rate methods are found more effective to achieve higher model performance without having to tune additional hyperparameters of the learning rate schedules in advance. For the Adam algorithm, it is believed that the actual learning rate can be adjusted with the update of estimates of the first and second moments of the gradients for each iteration [46]. Basically, it is sufficient to use the Adam algorithm for achieving a better optimization process. However, the previous study has shown that setting a learning rate decay to the Adam algorithm can effectively improve the model performance, which should not be overlooked [52]. Based on this, we set the Adam algorithm with a learning decay by using a function of tf.callbacks.ReduceLROnPlateau in TensorFlow, which can dynamically control the update of LR with a decay factor of 0.1 by monitoring the MSE value for every 10 epochs. For better illustration, the model performance by adopting different learning rate schedules in Eqs. (16)–(18) and a constant learning rate of 0.005 is investigated and compared with the Adam algorithm with a learning rate decay, as shown in Fig. 8(a). Compared to other learning rate schedules, the results demonstrate that the Adam algorithm exhibit more reliable model performance. To enhance the model performance, Fig. 8(b-c) show the influence of different learning rate for the Adam algorithm on convergence and accuracy. From the perspective of test accuracy, LR ranging from 0.003 to 0.008 shows good effectiveness, and this paper adopts 0.005 as the LR.

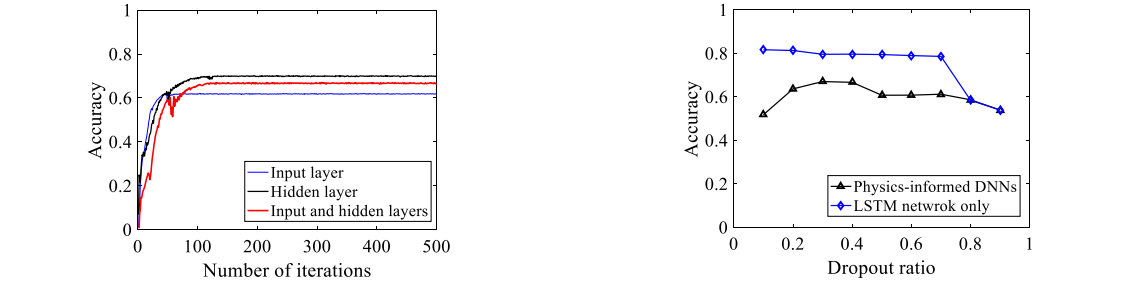
$$\mathrm{LR} = \mathrm{LR}_0 \cdot \mathrm{floor}(k_t / k_{total}) \tag{16}$$

$$\mathrm{LR} = \mathrm{LR}_0 \cdot 0.9^{30 \cdot k_t / t_{total}} \tag{17}$$

$$\mathrm{LR} = \mathrm{LR}_0 \cdot e^{-0.01 k_t} \tag{18}$$

4.5. Dropout ratio

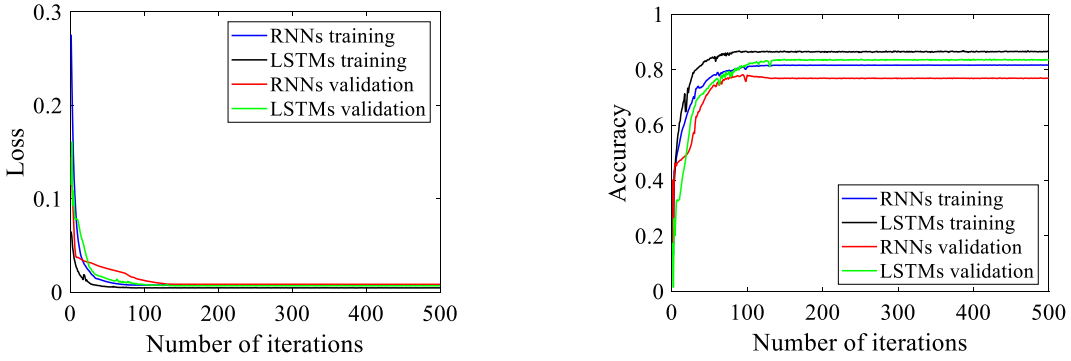
A major concern with the training of deep neural networks is the overfitting, which can lead to complex co-adaptations in which the network weights collaborate with one another to memorize the training dataset, thus the neural networks may



(a) Dropout ratio of 0.4 applied to different layers of RNNs

(b) Dropout ratio applied to the whole network and LSTM network only

Fig. 9. Influence of dropout ratio applied to different layers of the physics-informed networks.



(a) Loss curves

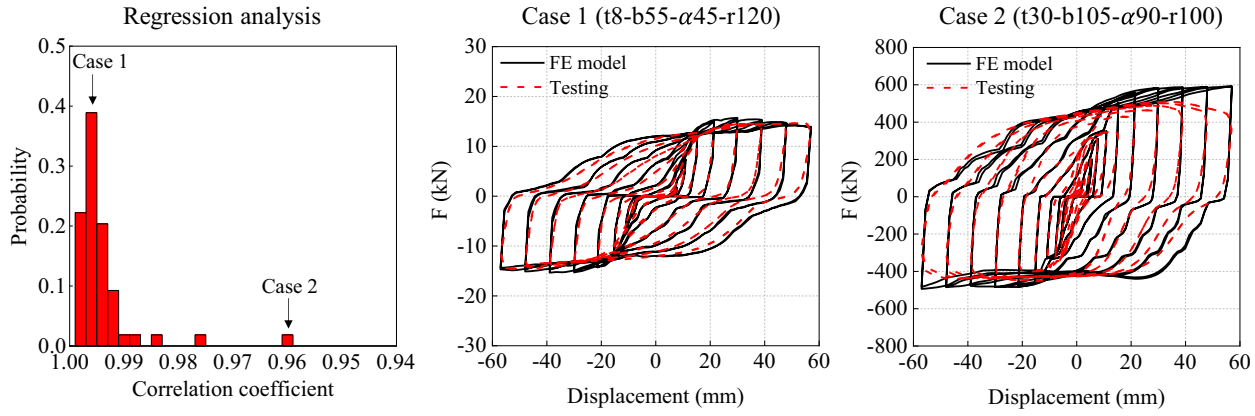
(a) Model accuracy

Fig. 10. Loss curves and model accuracy of RNNs and LSTM networks om training and validation datasets.

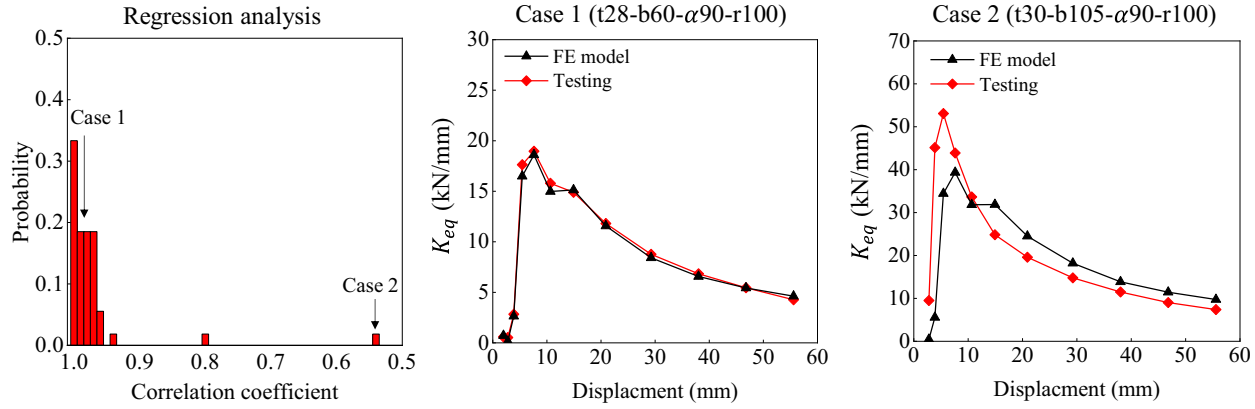
Table 6
Different geometry parameters of S-shaped dampers.

S-shaped dampers	Thickness t (mm)	Breadth b (mm)	Angle α(°)	Radius r (mm)
Group 1	[6:2:24]	55	90	100
Group 2	8	[50:5:100]	90	100
Group 3	8	55	90	[100:10:140]
Group 4	[10:2:30]	[55:5:105]	90	100
Group 5	[10:2:28]	[105: −5:60]	90	100

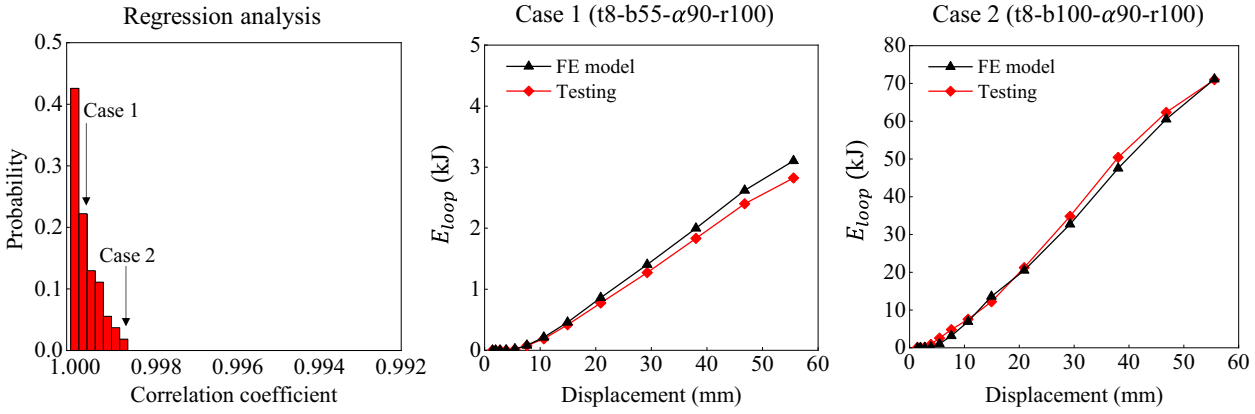
Note: [6:2:24] represents the parameter of t varied from 6 to 24 with an increment of 2, t = 6,8,10.



(a) Performance of predicated force F



(b) Performance of predicated equivalent stiffness K_{eq}



(c) Performance of predicated energy per loop E_{loop}

Fig. 11. Performance of proposed physics-informed DNNs.

exhibit excellent model performance but have worse performance on the test dataset [53–54]. To address this problem, great research interest has been attracted to the study of dropout in reducing the overfitting, as this approach is found effective in compressing the neural network by randomly dropping out part of cells and connections [55]. To investigate the influence, the dropout ratio ranging from 0.1 to 0.9 applied to the different layers of RNNs and LSTM networks is studied. For brevity, Fig. 9(a) illustrates the convergence of dropout ratio applied to the different layers for RNNs, which demonstrates that the application of dropout ratio to the RNNs (either input layer or hidden layers or both) performs poorly. Fig. 9(b) compares the influence of different dropout ratios applied to the hidden layers for whole networks (both RNNs and LSTMs) and two hidden layers of LSTM networks only. It can be seen that the application of the dropout ratio to the RNNs will significantly deteriorate the model performance. This is likely due to the RNNs is not complex enough to produce the overfitting issues. From the perspective of neural network structure, the hierarchical architecture determines that the output values of RNNs can be regarded as a part of the input for LSTM networks. The use of the dropout ratio for RNNs will harm the capacity of capturing the potential nonlinear feather due to randomly dropping some connection units, thus causing the loss of critical information when following the LSTM networks. Therefore, as suggested by previous research [11], it is not recommended to set a dropout ratio to the input layer. On the other hand, too large dropout ratios will result in poor performance. Although the dropout ratio from 0.1 to 0.7 has similar accuracy, the dropout ratio ranging from 0.4 to 0.6 is more effective in optimizing the model as it can eliminate the occurrence of noises resulting from a lower drop ratio. The dropout ratio of 0.5 is adopted for the proposed neural networks.

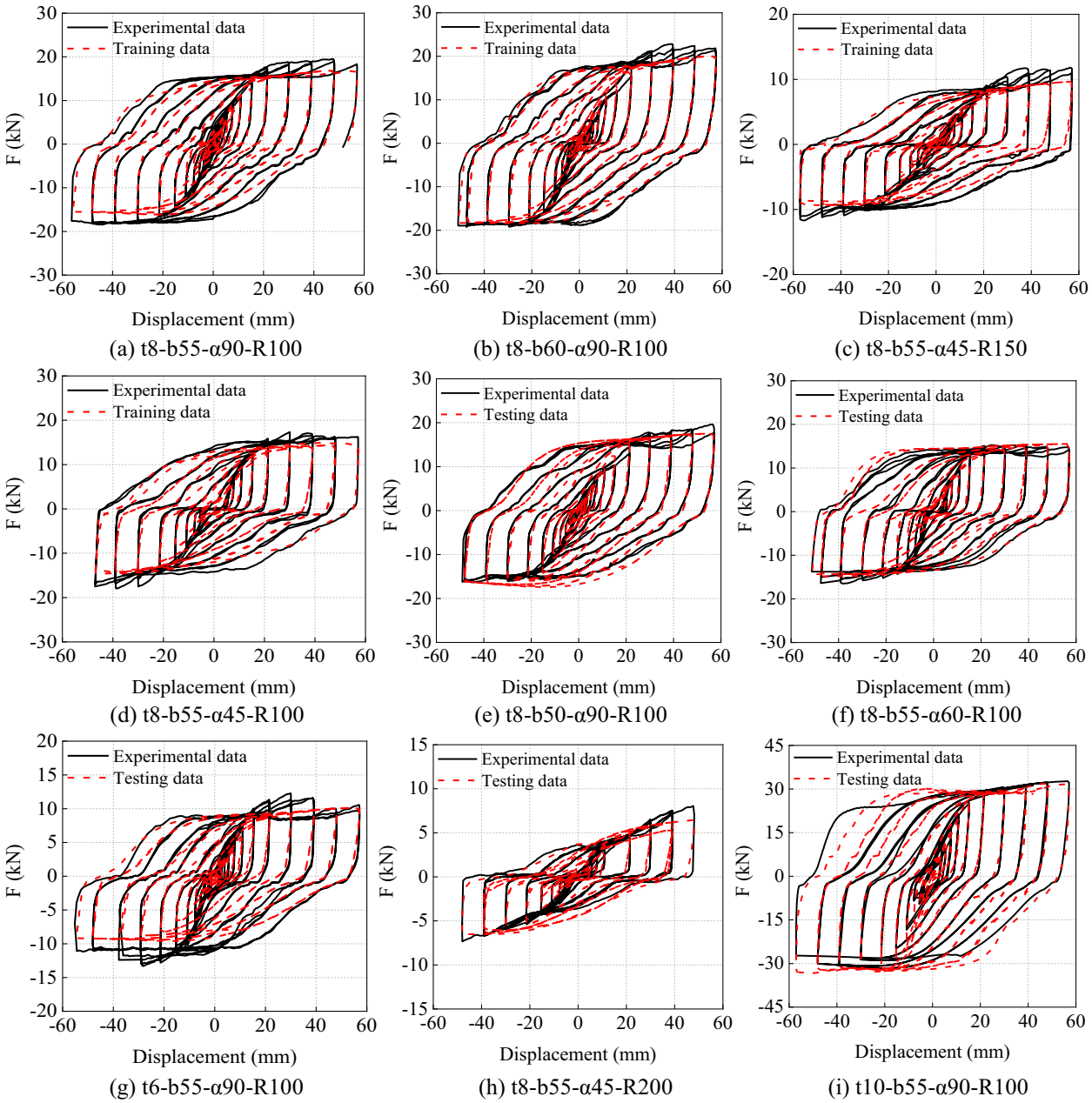


Fig. 12. Training and testing results using the proposed physics-informed DNNs.

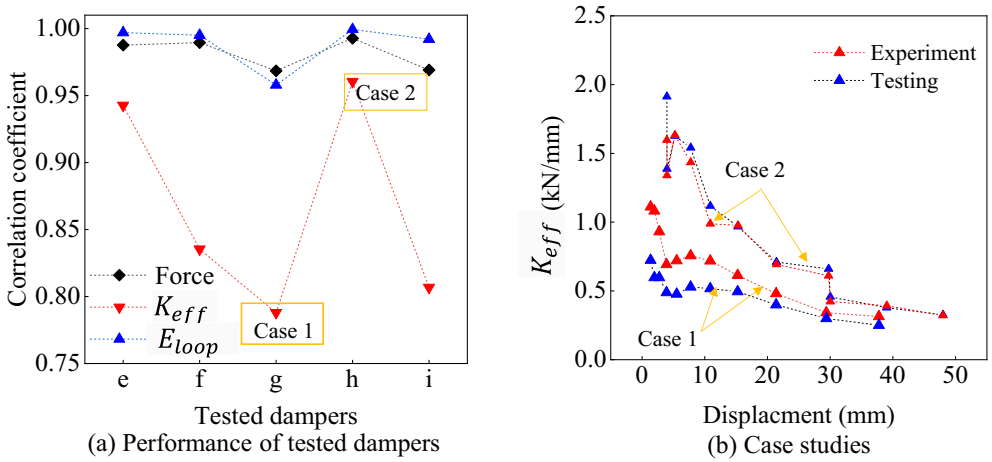


Fig. 13. Performance of tested dampers.

5. Case studies

5.1. Numerical validation

To verify the performance of the proposed neural networks, the elaborate 3D models of S-shaped steel dampers were developed to extend the dataset. Five groups of FE models that cover different geometry parameters of t, b, α and r were developed by ABAQUS, as shown in Table 6. The FE models were firstly calibrated by the experimental data and were extended to generate a suite of fifty-four samples of S-shaped steel dampers with different geometry parameters. For the training dataset, two samples of S-shaped steel dampers were randomly selected from groups 1–4 separately and one sample for group 5, and the other nine samples were selected for the validation. The remaining samples were used for the test dataset. Fig. 10 demonstrates the loss curves and model accuracy of the RNNs and LSTM networks on the training and validation datasets, which indicates that both RNNs and LSTM networks can converge quickly with good stability and model performance. Fig. 11 presents comparative results of the test dataset from the perspective of predicted force F, equivalent stiffness K_{eff} and consumption energy for each loop E_{loop}. The correlation coefficient R is used to evaluate the performance of regression, and a higher value of R represents the better performance. In terms of predicted force, R are all greater than 0.95, indicating good performance of proposed physics-informed neural networks in capturing the force–displacement relationship of S-shaped steel dampers, as shown in Case 1. The main divergence in case 2 is mainly induced by the failure of capturing the maximum force value. This is probably due to the force jump from the loading status to the unloading status. The small discrepancy in the pinching stage of the curve can be attributed to the slipping of dampers, which is not considered in the training. The equivalent stiffness K_{eff} is defined as the $(|F^+| + |F^-|)/(|d^+| + |d^-|)$, in which F^+ and F^- are the maximum and minimum force for each loading cycle, respectively; d^+ and d^- are the corresponding maximum and minimum displacement. It can be seen in Fig. 11(b) that major correlation coefficients for K_{eff} are more than 0.9 except for the same case 2 for the damper t30-b105-α 90-r100 with a poor correlation coefficient of 0.55, indicating a larger discrepancy between the testing results and the FE model results. This makes sense as the equivalent stiffness K_{eff} is related to the maximum force. Despite the possible discrepancy in predicting F and K_{eff}, the correlation coefficient for predicted E_{loop} is overall high as presented in Fig. 11(c). This shows that the proposed neural network can simulate the seismic performance of dampers with reasonable accuracy.

5.2. Experimental validation

To validate the generality of the network, the proposed physics-informed DNNs were validated by the experimental data. Different from the numerical dataset, the experimental dataset incorporates some noises induced by the instability of the loading system and random error recorded by sensors, which is not as smooth. Due to the limited available datasets, four specimens of dampers in Fig. 12(-a-d) were used for training and the other five in Fig. 12(e-i) for testing. The comparative results in Fig. 12 show that both the training and testing results correlate well with the experimental data. Moreover, compared to the numerical model developed by OpenSees, the proposed physics-informed DNNs can improve the discrepancy at the stage of strength degradation to some extent. The predicated F and E_{loop} have high correlation coefficient values of more than 0.95, as presented in Fig. 13(a). A larger discrepancy of equivalent stiffness K_{eff} is observed between the testing and experimental results for the damper t6-b55-α 90-r100, see case 1 in Fig. 13(b). The larger discrepancy mainly occurs between the maximum force predicted by the physics-informed DNNs and experimental data, especially for the elastic stage. The same phenomenon is also observed in the elastic stage for case 2. It is found that such discrepancy is mainly resulting from the noises existing in experimental data. Despite this, the physics-informed DNNs demonstrate good effectiveness and reasonable accuracy in forecasting the potential nonlinear behaviors of S-shaped steel dampers even with a small-scale dataset.

6. Conclusion

This paper presents physics-informed deep neural network (DNN) architecture to simulate the hysteretic behaviors of the novel S-shaped steel dampers. To train the network, hyperparameter tuning based on grid research and experience knowledge is elaborately investigated. The optimization model derived from the proposed physic-informed DNNs is validated numerically and experimentally, which demonstrates good performance in forecasting the nonlinear behaviors of S-shaped steel dampers such as the displacement force relationship, equivalent stiffness and energy dissipation capacity. As the proposed physics-informed DNN architecture incorporates the influence of geometry parameters, it can be taken as an alternative for the calibration of seismic design of S-shaped steel dampers to avoid the inherent departure from the actual system due to the numerical simplification and assumption. In addition, the physics-informed DNNs can be applied to the parametric study on potential mechanical characteristics of S-shaped steel dampers promptly and efficiently.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors are grateful for the financial support from the National Natural Science Foundation of China (Project No. 51878674, 52022113) and the Fundamental Scientific Research Expenses of IME, China Earthquake Administration (Project No. 2020EEEVL0403).

References

[1] Guo W, Hu Y, Li Y, et al. Seismic performance evaluation of typical dampers designed by Chinese Code subjected to the main shock-aftershocks. Soil Dyn Earthquake Eng 2019;126. https://doi.org/10.1016/j.soildyn.2019.105829.
[2] Chen Y, Chen C, Jiang H, et al. Study of an innovative graded yield metal damper. J Constr Steel Res 2019;160:240–54.
[3] Zhai Z, Guo W, Yu Z, et al. Seismic performance assessment of steel strip dampers equipped in high-rise steel frame. J Constr Steel Res 2020;177(2):106437.
[4] Silwal B, Ozbulut OE. Aftershock fragility assessment of steel moment frames with self-centering dampers. Eng Struct 2018;168(AUG.1):12–22.
[5] Zhai Z, Guo W, Yu Z, et al. Experimental and numerical study of S-shaped steel plate damper for seismic resilient application. Eng Struct 2020;221:111006. https://doi.org/10.1016/j.engstruct.2020.111006.
[6] Guo L, Wang J, Wang W, et al. Experimental, numerical and analytical study on seismic performance of shear-bending yielding coupling dampers. Eng Struct 2021;244(2):112724.
[7] Guo W, Wu J, Hu Y, et al. Seismic performance evaluation of typical dampers designed by Chinese building code. Earthq Eng Eng Vib 2019;18:433–46.
[8] Fr A, Lc B. Artificial neural network model to predict the flutter velocity of suspension bridges - ScienceDirect. Comput Struct 2020;233:106236. https://doi.org/10.1016/j.compstruc.2020.106236.
[9] Oh BK, Glisic B, Park SW, et al. Neural network-based seismic response prediction model for building structures using artificial earthquakes. J Sound Vib 2020;468:115109. https://doi.org/10.1016/j.jsv.2019.115109.
[10] Zhang R, Chen Z, Chen S, et al. Deep long short-term memory networks for nonlinear structural seismic response prediction. Comput Struct 2019;220:55–68.
[11] Xu Y, Lu X, Cetiner B, et al. Real-time regional seismic damage assessment framework based on long short-term memory neural network. Comput-Aided Civ Infrastruct Eng 2020;36(4):504–21.
[12] Seventekidis P, Giagopoulos D, Arailopoulos A, et al. Structural health monitoring using deep learning with optimal finite element model generated data. Mech Syst Sig Process 2020;145:106972. https://doi.org/10.1016/j.ymssp.2020.106972.
[13] Abdeljaber O, Avci O, Kiranyaz MS, et al. 1-d cnns for structural damage detection: verification on a structural health monitoring benchmark data. Neurocomputing 2017;275(JAN.31):1308–17.
[14] Wang DH, Liao WH. Modeling and control of magnetorheological fluid dampers using neural networks. Smart Mater Struct 2005;14(1):111. https://doi.org/10.1088/0964-1726/14/1/011.
[15] Ramezani M, Bathaei A, Ghorbani-Tanha AK. Application of artificial neural networks in optimal tuning of tuned mass dampers implemented in high-rise buildings subjected to wind load. Earthq Eng En Vib 2018;17:903–15.
[16] Zhang R, Liu Y, Sun H. Physics-Informed Multi-LSTM Networks for Metamodeling of Nonlinear Structures. Comput Methods Appl Mech Eng 2020;369:113226. https://doi.org/10.1016/j.cma.2020.113226.
[17] Raissi M, Perdikaris P, Karniadakis GE. 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.
[18] Nascimento RG, Fricke K, Viana F. A tutorial on solving ordinary differential equations using python and hybrid physics-informed neural network. Eng Appl Artif Intell 2020;96(153):103996. https://doi.org/10.1016/j.engappai.2020.103996.
[19] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. Comput Methods Appl Mech Eng 2020;362(15). 112790.1–112790.29.
[20] Misyris GS., Venzke A, Chatzivasileiadis S. Physics-Informed Neural Networks for Power Systems. 2019. arXiv preprint arXiv:1911.03737v3.
[21] Dourado A, Viana F. Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue. J Comput Inf Sci Eng 2020;20(6):1–15.
[22] Nguyen-Thanh VM, Anitescu C, Alajlan N, Rabczuk T, Zhuang X. Parametric deep energy approach for elasticity accounting for strain gradient effects. Comput Methods Appl Mech Eng 2021;386(2):114096.
[23] Zhang R, Liu Y, Sun H. Physics-guided Convolutional Neural Network (PhyCNN) for Data-driven Seismic Response Modeling. Eng Struct 2020;215:110704. https://doi.org/10.1016/j.engstruct.2020.110704.
[24] Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of kirchhoff plate. Comput, Mater Continua 2019;59(2):433–56.
[25] Duchanoy C, Moreno-Armendáriz M, Moreno-Torres J, et al. A Deep Neural Network Based Model for a Kind of Magnetorheological Dampers. Sensors 2019;19(6):1333. https://doi.org/10.3390/s19061333.
[26] Yucesan YA, Viana F, Manin L, et al. Adjusting a torsional vibration damper model with physics-informed neural networks. Mech Syst Sig Process 2021;154(2):107552.
[27] Liu Y, Yang S, Guang D, et al. Modified strip model for indirect buckling restrained shear panel dampers. J Constr Steel Res 2021;175:106371. https://doi.org/10.1016/j.jcsr.2020.106371.
[28] Zhang C, Aoki T, Zhang Q, et al. Experimental investigation on the low-yield-strength steel shear panel damper under different loading. J Constr Steel Res 2013;84:105–13.
[29] Xu L, Nie X, Fan J. Cyclic behavior of low-yield-point steel shear panel dampers. Eng Struct 2016;126:391–404.
[30] Kishiki S, Takayama D, Yamada S, et al. Experimental evaluation of cyclic deformation capacity of u-shaped dampers subjected to bi-directional loadings. J Struct Construct Eng (Trans AIJ) 2012;77(680):1579–88.
[31] Guo W, Li S, Zhai Z, et al. Seismic performance of a new S-shaped mild steel damper with varied yielding cross-sections. J Build Eng 2022;45:103508. https://doi.org/10.1016/j.jobe.2021.103508.
[32] Deng K, Pan P, Li W, et al. Development of a buckling restrained shear panel damper. J Constr Steel Res 2015;106:311–21.
[33] Mazzoni S, McKenna F, Scott MH. et al. Open System for Engineering Simulation User-Command-Language Manual, Version 2.0, Pacific Earthquake Engineering Research Center, Univ California, Berkeley (CA); 2009.
[34] Wen YK. Method for random vibration of hysteretic systems. J Eng Mech Divis 1976;102(2):249–63.
[35] Kwok NM, Ha QP, Nguyen MT, et al. Bouc-Wen model parameter identification for a MR fluid damper using computationally efficient GA. Isa Trans 2007;46(2):167–79.
[36] Hossain MR, Ashraf M, Padgett JE. Risk-based seismic performance assessment of Yielding Shear Panel Device. Eng Struct 2013;56(6):1570–9.
[37] Zhu X, Lu X. Parametric Identification of Bouc-Wen Model and Its Application in Mild Steel Damper Modeling. Procedia Eng 2011;14:318–24.
[38] Qiang Y, Zhou L, Wang X. Parameter identification of hysteretic model of rubber-bearing based on sequential nonlinear least-square estimation. Earthq Eng Eng Vib 2010;9(3):375–83.
[39] Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. Comput, Mater Continua 2019;59(1):345–59.
[40] Gal Y, Ghahramani Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. Statistics, 2015, arXiv:1512.05287v5, 285-290.
[41] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput 1997;9(8):1735–80.
[42] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Machine Learn Res 2012;13(1):281–305.
[43] Gardner WA. Learning characteristics of stochastic-gradient-descent algorithms: a general study, analysis, and critique. Signal Process 1984;6(2):113–33.
[44] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Machine Learn Res 2011;12(2121–2159):2011.
[45] Tieleman T, Hinton G. Lecture 6.5-RMSPprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks Machine Learn 2012;4(2):26–31.
[46] Kingma D, Ba J. Adam: A Method for Stochastic Optimization. Comput Sci, 2014, arXiv:1412.6980v8.
[47] Ruder S. An overview of gradient descent optimization algorithms. Comput Sci, 2016, arXiv preprint. arXiv:1609.04747.
[48] Yu T, Zhu H. Hyper-parameter optimization: a review of algorithms and applications. Comput Sci, 2020, arXiv:2003.05689.
[49] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, vol. 15, 2011, p. 315–323.
[50] Chen S, Zhang Y, Yang Q. Multi-task learning in natural language processing: an overview. Comput Sci, 2021, arXiv:2109.09138.
[51] Vandenhende S, Georgoulis S, Gansbeke WV, Proesmans M, Dai D, Gool LV. Multi-task learning for dense prediction tasks: a survey. Comput Sci, 2020, arXiv:2004.13379.
[52] Loshchilov I, Hutter F. Decoupled weight decay regularization. Comput Sci, 2017, arXiv:1711.05101.
[53] Hinton GE, Srivastava N, Krizhevsky A, Sutskeve I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. Comput Sci, 2012, arXiv:1207.0580, 2012.
[54] Labach A, Salehinejad H, Valaee S. Survey of dropout methods for deep neural networks. Comput Sci, 2019, arXiv:1904.13310.
[55] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Machine Learn Res 2014;15(1):1929–58.