Title.

Forecasting measured sensor responses of structures using temporal deep learning and dual attention

Running title. Forecasting Sensor Responses

Viet-Hung Dang[a,*], Trong-Phu Nguyen[a], Thi-Lien Pham[b], X. Huan Nguyen[c,d]

[a] Department of Structural Mechanics, Hanoi University of Civil Engineering, Hanoi, Vietnam

[b] Faculty of Applied Science, University of Economics-Technology for Industries, Hanoi, Vietnam

[c] London Digital Twin Research Centre, Faculty of Science and Technology, Middlesex University, London, UK

[d] International School, Vietnam National University, Hanoi, Vietnam

*Corresponding author. E-mail: hungdv@huce.edu.vn

Abstract. The objective of this study is to develop a novel and efficient model for forecasting the non-linear behavior of structures in response to time-varying random excitation which is a challenging task for the civil engineering community. The backbone idea is to design a deep learning architecture to leverage the dual relationship within multiple time-series data which are the inter-relationship between external excitations and structure's vibration signals, as well as the intra-relationship between historical values with future values within times series data. The proposed method consists of two main steps: the first step applies a global attention mechanism to combine multiple measured time series and time-varying excitation into a weighted time series before feeding it to a temporal architecture, and the second step utilizes a self-attention mechanism followed by a fully connected layer to predict multi-step ahead values. The viability of the proposed method is demonstrated via two case studies involving synthetic data from a 3D reinforced concrete structure and experimental data from an 18-story steel frame. Furthermore, comparison and robustness studies were carried out, showing that the proposed method outperforms other conventional methods and maintains high performance in the presence of noise with an amplitude of less than 10%.

Keywords. Structural dynamic, time-varying excitation, deep learning, signal processing, response forecasting

1. Introduction

Large-scale structures such as long-range bridges, skyscrapers, and wide-span roof structures have been increasingly popular worldwide, usually classified as high-safety and expensive assets. Accurately predicting the behaviors of these structures with reduced time complexity has been a longstanding endeavor in civil engineering. This capability facilitates the execution of numerous subsequent critical tasks such as structural optimization, reliability analysis, filling missing measured data, preventing collapses, and so on. Motivated by this, in this study, we developed a novel and efficient surrogate model for forecasting the non-linear behavior of large-scale structures in response to time-varying random excitation. However, this task is highly challenging especially when taking into account the non-linearity and dynamic behavior of structures. Classical methods such as a finite element model are computationally expensive or even intractable because they require fine time and space discretization, to ensure analysis convergence; thus, its applicability for prediction of large-size structures is limited. On the other hand, the model-free, a.k.a, the data-driven and other alternative methods have been successfully applied in place of model-based methods in a spectrum of sectors such as structural damage detection [1-5], predicting structures' responses [6-9], structural reliability analysis [10, 11], experimental measurement [12, 13], and so forth.

In the context of structural analysis, Moller and Reuter [14], investigated a model-free approach, a.k.a, fuzzy ARMA, for predicting structural responses given historical uncertain time-series data. The effectiveness of the method was demonstrated through various case studies such as predicting the time-variant damage state of a reinforced concrete T-beam plate, forecasting the settlement of a slope bordering highways, and deformation of a pavement. Recently, in order to predict structures' response to an earthquake, Zhang et al. [15], developed a data-driven using the long short-term memory (LSTM) network thanks to its capability in capturing the long-range dependency in time-series. The method's performance was tested with both field data and synthetic data, showing high prediction accuracy with an error under 10%. Guo et al. [16] deep neural network to approximately predict the transversal deflection of Kirchhoff plate. Later, the authors expand the idea for predict vibration and buckling behaviors of Kirchhoff plates [17]. Besides, various authors have shown increasing interests in utilized physics-informed deep learning-based method to find solution of different underlying partial different equations of structural and mechanical systems [18-20].

In reality, the structure's responses to external excitations follow some underlying physical rules; therefore, Zhang et al. [21] combined the convolutional neural network with equations of motion, forming a physical-guided data-driven approach able to accurately predict buildings' behaviors under earthquakes. Furthermore, the method was applied to assess the serviceability of a full-size six-story building, providing a fragility curve useful for the maintenance and rehabilitation operation plan of the structure. Oh et al. [22] proposed a deep convolutional neural network-based method for forecasting displacement time-series of building structures prone to seismic excitation using their historically measured acceleration as inputs. The validity of the method was demonstrated through an experimental shaking-table database from the Seismic Disaster Prevention Center, South Korea. Nevertheless, there exist some pitfalls in popular deep learning architectures such as 1DCNN and LSTM when working with time series data. For example, 1DCNN could distill invariant local features but is not adept at capturing temporal relationships within time series data. Meanwhile, LSTM can retain long-term relationships but treats values at different time steps equally, which may not be optimal for predicting future values. It is observed that the correlation between different time series inputs and the time series output is not the same. For example, vibration signals recorded from nearby sensors show more similar patterns than those from distant sensors.

Given the reviewed difficulties faced by model-based methods and the drawbacks of some popular data-driven methods, this study aims to develop a novel and efficient surrogate model for forecasting the structures' dynamic behaviors. The intuitive idea behind this study is to find an effective and efficient way to leverage the inter-relationship between time-varying external excitations and the structures' vibration signals, as well as the intra-relationship between historical values with future values within times series data. To achieve this, we propose a deep learning-based framework featuring a dual-attention mechanism. The first global attention mechanism combines multiple structural time-series inputs and time-varying excitation into a softmax weighted time-series before feeding to an LSTM layer. The second attention layer, namely, self-attention, is applied to the time-series output obtained from LSTM to predict multi-step values ahead. Besides, the sliding window technique is adopted, where highly accurate prediction outputs of previous steps are appended to the input of the later step; by doing so, long-term forecasting can be undertaken.

Table 1 Comparing the proposed S-DAN framework with recent LSTM-based models for time-series forecasting in literature.

Work	Architecture	Input data	Output	Application	Year
Zhang et al. [15]	Dual LSTM	Univariate	Multi-step prediction	Structural analysis	2019
Yu et al. [23]	Physic-guided RNN + LSTM	Multivariate	Multi-step prediction	Structural analysis	2020
Xu et al. [24]	LSTM	Univariate	Classification	Structural damage detection	2020
Du et al. [25]	Encoder-decoder + LSTM + Attention	Multivariate	Classification	Human emotion recognition	2020
Gao and Ruan [26]	Encoder-decoder + LSTM + Attention	Multivariate	Multi-step prediction	Energy consumption	2021
Liu et al. [27]	LSTM + Attention + Feature extraction	Univariate	Multi-step prediction	Precipitation forecasting	2021
Zhang et al. [28]	CNN + Attention + LSTM	Univariate	Classification	Muscle fatigue detection	2021
Kong et al. [29]	Bidirectional LSTM + Attention	Multivariate	Anomaly detection	Internet of Things	2021
Hsu et al. [30]	TCN + LSTM + Attention	Multivariate	Remaining useful life	Semiconductor manufacturing	2022
Sun et al. [31]	Feature/temporal-attention + LSTM	Univariate	Remaining useful life	Power distribution system	2022
He et al. [32]	Dual attention	Multivariate	One-step prediction	Wind speed prediction	2023
This study	Dual attention +	Multivariate	Multi-step	Structural	2023

LSTM	prediction	analysis

Table 1 highlights the differences between the proposed dual-attention based framework and other recent works in the literature, which also leverage the LSTM with attention mechanism (LSTM-AM) to handle time series data. It can be seen that using LSTM along with the attention mechanism has been widely acknowledged for its superior accuracy across different domains such as weather forecasting, healthcare, semiconductor, and energy. In the domain of civil structures, some LSTM-AM models [15, 24] were originally designed to work with univariate time series; thus, their applicability to multivariate time series has not been justified. Actually, the idea of using a dual-attention mechanism has been used in literature, for example, in the work of He et al. [32] where it was utilized for predicting wind speed. On the one hand, this implies that the attention mechanism is effective in handling multiple time-series data. On the other hand, such an algorithm remains relatively under-explored in the field of civil engineering. It is noted that building a robust and functional framework requires domain knowledge, deep understanding of the data, and suitable data processing. Therefore, while these methods may be partly similar in terms of algorithm, their implementations can differ significantly. Moreover, in [32], the author only predicted only a single value each time whereas the proposed method performs multi-step prediction. In short, the main contributions of this study can be summarized as follows:

- A model-free framework is developed for forecasting structural responses viable for both linear and non-linear behaviors, without requiring sophisticated numerical models. It is capable of extracting underlying patterns embedded in historical data as well as the interaction relationship among multi-variate input and output time-series.
- The correctness and effectiveness of the proposed framework are demonstrated through two case studies involving synthetic data of a 3D reinforced concrete frame structure and experimental data of a 18-story steel structure. The obtained results clearly showed that the proposed method outperforms counterparts such as vector autoregression (VAR), Extreme Gradient Boosting (XGB), and LSTM by a margin of more than 30% in terms of accuracy.
- Parametric studies are conducted to provide insights into the effect of different parameters on prediction performance. Prediction errors increase with the prediction length and the intensity of the excitation; the proposed method still provides acceptable results with the presence of random noise having an amplitude no greater than 10% of the root mean square value of vibration data.

The rest of the paper is organized as follows: Section 2 presents the architecture of the deep neural network with dual attention mechanisms designed for forecasting time-varying structural behaviors. In Section 3, the performance of the proposed approach is justified through two case studies. Finally, the conclusions and perspectives are drawn in Section 4.

2. Methodology

This work aims to develop a data-driven approach for forecasting the dynamic response of structures under time-varying excitations, whose main components are schematically presented in Fig. 1.

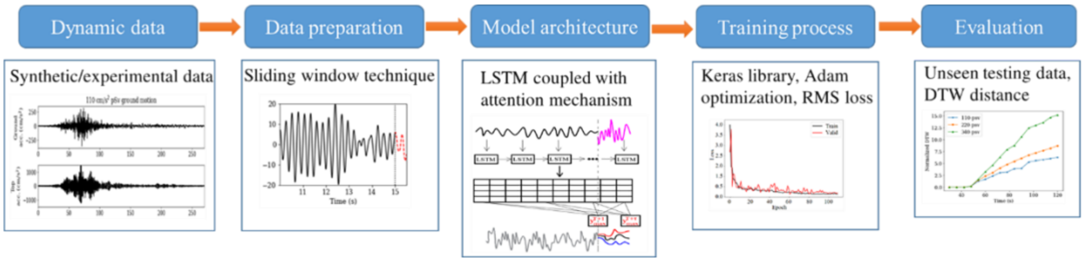


Fig. 1 Workflow of the proposed S-DAN framework for forecasting structures' responses.

2.1 Data-Driven Pipeline for Forecasting Structure's Response

At first, a structural database is required, including excitation and structural responses, which can be obtained either synthetically using numerical modeling or experimentally via a series of measurements. Secondly, the database is split into non-overlapping subsets, i.e., training, validation, and testing for the training and evaluation process. Unlike other data types, an additional preprocessing step is required for time-series data, involving the sliding window technique to further divide each subset into batches of input/output pairs. Thirdly, a deep learning architecture based on LSTM and attention mechanism is devised, encompassing multiple time series as input for forecasting structural dynamic responses. In the fourth step, the machine learning Keras library and Adam optimization are used to build and train the deep learning model. Finally, the model's performance is evaluated using unseen testing data and predefined measurement metrics such as Root Mean Square Error, Dynamic Time Warping, etc. Further details of each component are explicitly explained in the following sections.

2.2 Mathematical notation

At first, the mathematical notations utilized throughout the paper are presented. Considering a structure equipped with N sensors across its body, one denotes measured quantities by $\boldsymbol{X} = [X_1, \dots, X_N]$, where $X_i = \left[x_{i,1}, \dots, x_{i,N_t}\right]^T$ with subscript i refers to sensor i, N_t is the total number of measured time instant, and $x_{i,j}$ signifies a measurand at time instant j of sensor i. Here, one considers displacement or acceleration as the quantities of interest, which are also the most commonly used in practice. The known time-varying excitations are denoted by $\boldsymbol{F} = [F_1, \dots, F_M]$, with $F_k = [f_{k,1}, \dots, f_{k,N_t}]$ where M is the number of excitations, $f_{k,l}$ represents a value of excitation k at time instant l. The aims of this study is to estimate a sequence of next values of a quantity of interest denoted by $y(N_t + \tau)$, with $\tau = 1, \dots, \tau_{max}$ based on historical measured values and excitations, which constitutes a multivariate time-series multi-step forecasting problem. In this study, it is assumed that excitations are known in advance. It is common in the structural design practice to perform many calculations with various known excitations to determine the structures' most critical responses and derive conservative design solutions. Afterwards, we introduce the important weight α assigned to time-series data which will be used later in the global attention layer, χ is the weighted time series obtained at the output of the global attention layer. For the LSTM layer, the associated hyperparameter is the dimensionality of the output, being denoted by N_{lstm}, and the output of the LSTM layer is signified by χ_{lstm}. For the self-attention layer, three variants of χ_{lstm} named query, key and value vectors are symbolized by $Q, K,$ and V. In addition, an attention matrix A will be calculated to calculate the output vector χ_{att} of the self-attention layer.

2.3 Deep Learning Architecture Using Dual Attention Mechanism

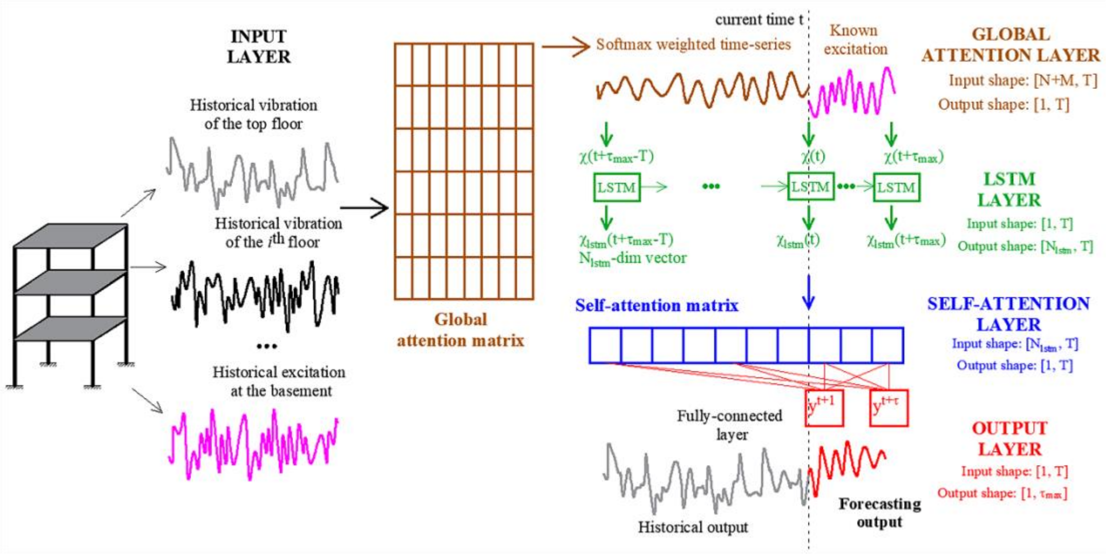


Fig. 2 Architecture of the deep learning model within the S-DAN framework featuring the dual-attention mechanism.

We elaborate on a data-driven method for structural analysis using LSTM coupled with a dual-attention mechanism, named S-DAN (Structural analysis - Dual Attention Network). The approach is based on the intuition that the attention mechanism allows for selectively concentrating on the most relevant components among multiple time-series via more important weights, while the LSTM architecture permits retaining long-range dependencies via continuous cell states. Thus, by combining long-range information and attention information, one expects to achieve better prediction performance. The architecture of the proposed method is schematically illustrated in Fig. 2, where different colors highlight different layers. There are five layers in total: the Input layer, Global attention layer, LSTM layer, Self-attention layer, and Output layer. For better clarification, each layer's input and output data shapes are specified, and the directed arrows are used to represent the data flow. The input layer comprises various time series such as vibration signals, time-varying excitation, and historical output. The details of the other layers are explained in the following paragraphs.

2.3.1 Global Attention Mechanism and LSTM

The global attention mechanism is used to combine multiple time-series into a new single time-series, which is mathematically described below. First, a fully-connected layer is applied to $\boldsymbol{X}$ and $\boldsymbol{F}$, as below:

$$\boldsymbol{E} = f(\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{F}) \tag{1}$$

where $\boldsymbol{W}$ is a weight matrix of size $(N + M, N + M)$, and $\boldsymbol{E}$ is an output matrix with a size of $(N + M, N_t)$. Next, at each time instant t, the importance weight $\alpha_{i,t}$ assigned to time-series i is calculated usingt the softmax function, as follows:

$$\alpha_{i,t} = softmax(E_{i,t}) = \frac{\exp(E_{i,t})}{\sum_{j=1}^{N+M} \exp(E_{j,t})} \tag{2}$$

where $t = 1, \dots, N_t$. It can be seen that the components of each row of α sum up to 1. After that, the new softmax weighted time series χ is calculated by:

$$\chi_t = \sum_{i=1}^{N+M} \alpha_{i,t} E_{i,t}) \tag{3}$$

Logically, for a given structural element, its surrounding excitation forces and nearby components' responses will receive larger weights than those of other components from afar. In addition, a time-series of low amplitude will apparently have less impact than those with high amplitude. Besides, it is noted that excitations are supposed to be known in advance by S-DAN.

2.3.2 Long Short-Term Memory Layer

Next, χ goes through a LSTM layer, aiming to identify the inherent longterm dependency within its setting. In the context of structural analysis, it could be intrinsic periodicities or the prevalence of a special vibration mode triggered by excitation. LSTM is a variant of the recurrent neural network family consisting of a chain of connected identical cells. Each cell behaves like a small neural network with its own weight matrix, nonlinear activation, and its output regarded as inputs of its successor. Thus, the chain-like nature of LSTM is naturally suitable for time-series data. The central idea of LSTM is to calculate two separate outputs at each cell: one instantaneous hidden output and another cell state output whose values are between 0 and 1; a value of 0 corresponds to ignoring, while 1 corresponds to full retention. Cells with a state near one will have a significant influence on later cells in the network. Further theoretical foundations of LSTM can be found in [33]. The LSTM layer's hyper-parameters consist of the activation function type, dropout rate, and the dimensionality of the output. The latter, denoted by N_{lstm}, has a significant impact on the model performance. It defines the output shape of the LSTM layer, which is also the input shape of the self-attention layer, as shown in Fig 2. The effect of N_{lstm} will be investigated in the next section.

2.3.3 Self-Attention Mechanism and an FC Layer

The output of the LSTM layer, dubbed by χ_{lstm} will be fed into the second attention layer, namely, self-attention, where the influence of the value at each time step on others will be assessed. The realization steps of the self-attention are depicted in Fig. 3, involving three linear transformations of χ_{lstm} which result in three vectors, namely query Q, key K, and value V , then deriving the attention matrix $\boldsymbol{A}$ with shape (N_t, N_t). A_{ij} can be interpreted as how much the value at time instant i correlates with the value at time instant j when performing forecasting tasks. $\boldsymbol{A}$ is normalized using the softmax function so that the components of each row sum to 1.

Afterward, multiplying the attention matrix with vector V provides the output of the self-attention layer χ_{att}.

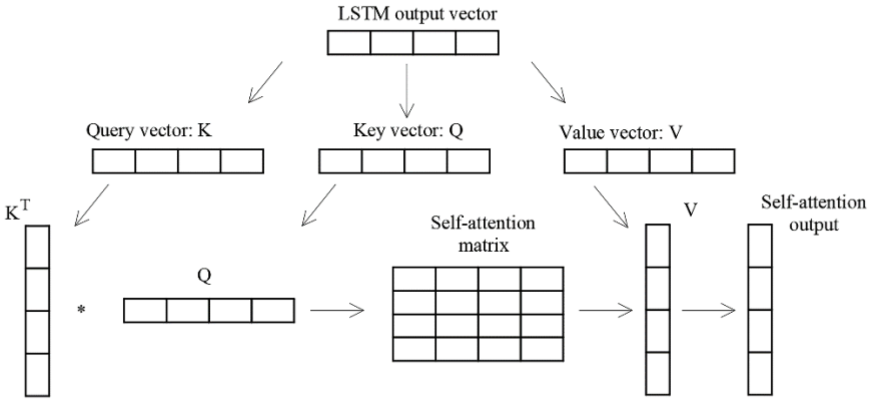


Fig. 3 Applying self-attention layer to the LSTM layer output.

Mathematically, these above steps could be expressed as follows [34]:

$$\begin{aligned} Q &= \boldsymbol{W}_Q \times \chi \\ K &= \boldsymbol{W}_K \times \chi \\ V &= \boldsymbol{W}_V \times \chi \end{aligned} \tag{4}$$

$$\boldsymbol{A} = softmax\left(\frac{Q \times K^T}{\sqrt{N_{hidden}}}\right) \tag{5}$$

$$\chi_{att} = V \times \boldsymbol{A} \tag{6}$$

Finally, χ_{att} goes through a fully connected layer to predict the next values of the time-series of concern. To quantify the deviation between predicted values and actual values, the commonly used root mean square (RMS) loss function is adopted.

Table 2 Number of trainable parameters of the proposed S-DAN framework.

Layer	Input shape	Output shape	Number of parameters
Global attention layer	$[N+M, N_t]$	$[1, N_t]$	$(N+M) \times (N+M)$
LSTM layer	$[1, N_t]$	$[N_{lstm}, N_t]$	$4 \times N_{lstm} \times (1 + N_{lstm})$
Self-attention layer	$[N_{lstm}, N_t]$	$[1, N_t]$	$3 \times N_{lstm} \times N_{hidden}$
Output layer	$[1, N_t]$	$[1, \tau_{max}]$	$(N_t + 1) \times \tau_{max}$

N_{hidden} is the number of neurons in the one-layer feedforward network for calculating vectors *Q*, *K*, *V* per Eq. (4).

Table 2 enumerates in detail the input/output shapes and the number of trainable parameters of each layer of the proposed S-DAN approach. Specifically, for the global attention layer, the input and output shapes are $[N+M, N_t]$ and $[1, N_t]$, respectively. The trainable parameters of this layer come from the weight matrix in Eq. (1) with $(N+M) \times (N+M)$ parameters. Next, for the LSTM layer, the input and output shapes of data are $[1, N_t]$ and $[N_{lstm}, N_t]$, respectively. Since each LSTM involves four feedforward transformations for computing input gate, forget gate, output gate, and cell state; thus, the number of trainable parameters is $4 \times N_{lstm} \times (N_{lstm} + 1)$. For the self-attention layer, trainable parameters come from constructing three vectors Q, K, V according to Eq (4). Thus, the number of trainable parameters is $3 \times (N_{lstm} + 1) \times N_{hidden}$ with N_{hidden} being the number of hidden neurons of W_Q as well as of W_K and W_V . After multiplying with the attention matrix, the data with shape

$[N_{lstm}, N_t]$ is averaged over the feature axis, resulting in an output with shape $[1, N_t]$. Finally, the output vector will predict the sequence of size $[1, \tau_{max}]$ via a fully connected layer whose number of parameters is around $(N_t + 1) \times \tau_{max}$.

Algorithm 1 Pseudocode of the proposed S-DAN framework.

Begin

  Data:

  Prepare groups of ground motion and structures' response time-series data.

  Split data into batches of input/output pairs using the sliding window technique.

  Apply Max-min normalization.

  Splitting data into train, validation, and test data.

  Function S-DAN model

    Define global attention, LSTM, self-attention, fully connected layers.

    Employ Adam Optimizer, RMS loss function.

    Define training parameters: learning rate decay, number of epochs, early stopping, mini-batch size.

  End Function

  Function *Training*

    For $i \leftarrow 1$ to number of epochs do

      For $j \leftarrow 1$ to number of mini-batches do

        Forecast output of each mini-batch.

        Calculate the loss function.

        Update weight using back-propagation.

      End

      Calculate validation loss function.

      Save the best so-far model, check stopping criteria.

    End

  End Function

  Function *Evaluation*

    1st forecasting using initially known part of testing data.

    Append forecasting results to input data.

    While $t < N_t$ do

      Forecasting next multiple-steps results.

      Append forecasting results to input.

    End

    Evaluate measurement metrics.

  End Function

End

Algorithm 1 summarizes the realization steps of the proposed framework. The implementation of S-DAN is carried out with the help of the machine learning library Keras 2 [35] written in Python thanks to its expressiveness, flexibility, and robustness. The adopted hyper-parameters of the model are a learning rate of 10^{-4}, a loss function of root mean square error, $N_{lstm} = 128$, an input length of 500, an output length of 50, and a mini-batch size of 256. In order to achieve high performance, some additional steps are conducted apart from those mentioned in the Algorithm: data normalization to suppress the scale difference of input variables and learning rate decay to refine the training when no reduction in the loss function is observed. On the other hand, some steps closely related to specific data under investigation, such as data windowing in data preparation, Dynamic Time Warping distance, and long-term forecasting by iterating the inference, will be clarified in more detail through the next two case studies.

3. Performance Evaluation: Case Studies

In this section, the applicability of the proposed method is validated through two case studies involving synthetic data of a 3D numerical reinforced concrete frame and experimental data of a 18-story steel building structure under seismic ground excitation. For each case study, the data preparation is first presented; then, the prediction accuracy is quantified. After that, the effect of key parameters on the model's performance is estimated, thus providing practical guidance for real-world applications.

3.1 Case Study 1: Synthetic Data of 3D Reinforced Concrete Frame

The first case study investigates the response of a six-story two-bay structure under various ground motions, as experimentally studied in [36]. To be specific, the output of interest is the top floor displacement, while the input data consist of ground motion and displacement time series of other floors. All stories have the same height of 0.75 m, resulting in a total height of 4.5 m, the bay widths in X-direction are 1.125 m and 1.425 m, and those in Y-direction are 1.275 m and 0.9 m, respectively, as can be seen in Fig. 4. The columns' cross-section is a 100×100 mm^2 rectangle; the beams in X-direction are 64.5 mm wide and 125 mm high, while the beams in Y-direction are 50 mm wide and 112.5 mm high. The floors are considered as rigid diaphragms, meaning that nodes belonging to the same floor have identical lateral displacements.

3.1.1 Numerical Model

To perform dynamic structural analysis, one utilizes the open-source program OpenSees [39] from the Pacific Earthquake Engineering Research Center because of its effectiveness and efficacy, which are widely acknowledged within the civil engineering community. The details of the FEM are described as follows. The spatial frame structure is modeled in a three-dimensional environment; each node has 6 degrees of freedom. Based on the geometry of the structure, there are in total 54 column-beam joints, 72 non-linear beam elements, and 54 non-linear column elements. All columns are fixed at their bases. In terms of material, the nonlinear constitutive law of steel is constructed using the bilinear model of Filippou et al. [37]. The constitutive law of concrete is simulated using the Kent-Scott-Park model [38] (Fig. 5). Specifically, steel rebar has a diameter of 4 mm, a yield strength f_y of 274.11 MPa, and an elastic modulus E_s of 182 GPa, while the concrete has a compressive strength f_c of 35.96 MPa, and elastic modulus E_c of 24.25 GPa.

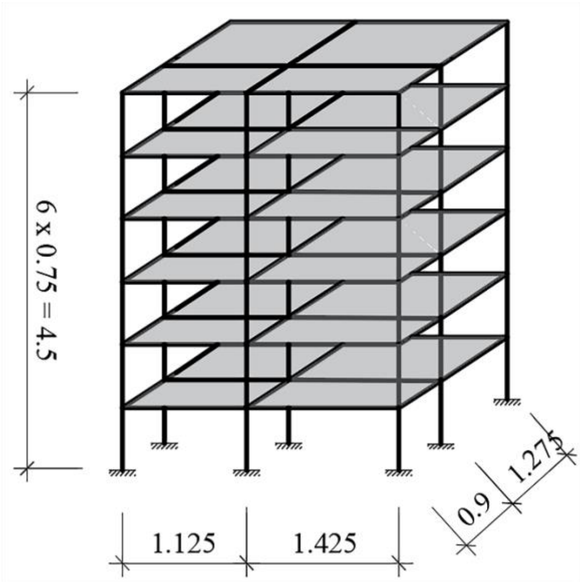
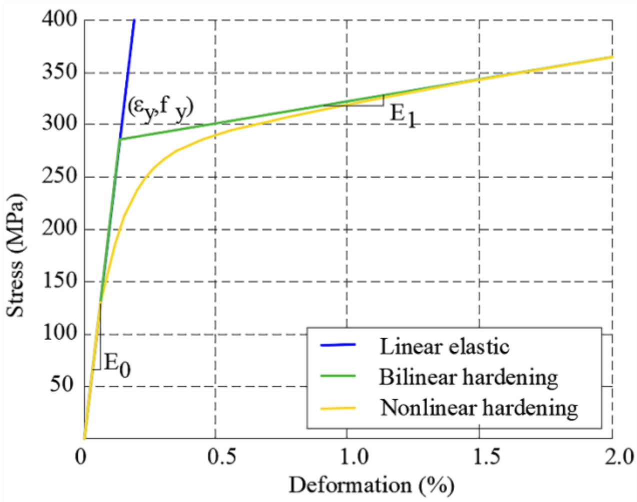
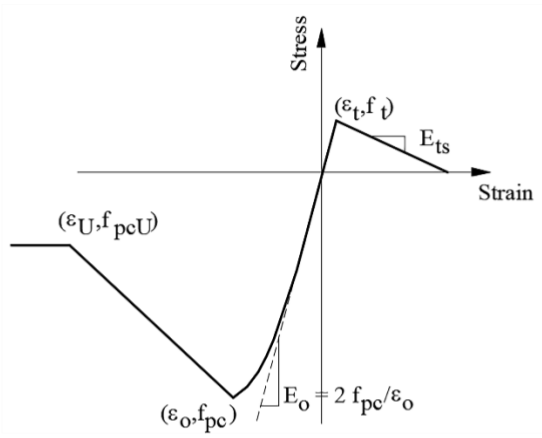


Fig. 4 Graphical representation of the six-story two-bay reinforced concrete frame structure from [36].





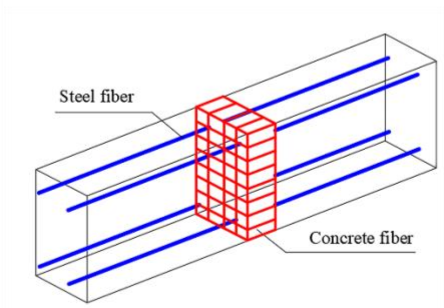


Fig. 5 Illustration of section modeling in the OpenSees model accounting for material nonlinearity. At the top is shown the bilinear model for steel [37]. At the bottom is shown the Kent-Scott-Park model for concrete [38] and the fiber approach for reinforced concrete sections.

Regarding section modeling, the section of reinforced concrete elements is simulated using the fiber approach (Fig. 5(c)), which can account for moment curvature, axial force-deformation, and their interaction at the same time. This approach is superior to the uniaxial section approach, which calculates bending and normal stresses independently. The forced-based distributed plasticity beam-column element in OpenSees is utilized to account for the plasticity potentially developing in the structural members when excitations increase beyond an elastic threshold. With such an element, the cross-section is assumed to be prismatic both before and after deformation; the integration along the element is calculated by using the Gauss-Lobatto quadrature rule. The plasticity will spread along the length of elements, and the iterative flexibility formulation is adopted to ensure the compatibility condition of the elements. The floors are considered rigid diaphragms, meaning that nodes belonging to the same floor have identical lateral displacements. For validation, the first two natural frequencies of the replicated model are 3.41 and 3.67 Hz, which closely match those of the tested model in [36], i.e., 3.45 and 3.72 Hz.

Next, this model is excited by different ground motions, and its nodes' displacements are recorded, forming the database for S-DAN. The excitations are real ground motions recorded and published by the Center for Engineering Strong Motion Data [40]. To increase the variety of the database, ten ground motions from different regions in the world are utilized: Kobe 1995 in Japan; El Salvador 2001; Fairbank 2000, Indiana 2002, San Simeon 2003 in the USA; Lima 1974, Santiago 1985 in Chile; Rarakau 2012 in New Zealand; Taiwan 1986; Karditsa 1995 in Greece; and Tonalapa 1993 in Mexico. In addition, different load scale factors ranging from 0.5 to 2.0 with an increment of 0.1 were applied to input ground motions. Scaling, in this context, means directly increasing the amplitudes of ground motions without changing other characteristics, such as frequency content. Afterward, an extensive suite of simulations with these ground motions and different scale factors is carried out for the six-story RC frame presented above. In each simulation, the system of structural dynamic equations is solved numerically using the iterative Newton-Raphson algorithm in conjunction with the Newmark integration method with coefficients of $\gamma = 0.25$ and $\beta = 0.5$. The time step of dynamic analysis is initially set to 0.01s, whereas the time duration of each simulation is equal to the length of the input ground motion. Besides, Rayleigh damping with a damping ratio of 0.02 is utilized to assign damping for elements of the structure. After that, simulation results from Karditsa, Fairbank, and Taiwan ground motions' peak values are separated as unseen test data, while those from other ground motions constitute the training data for the S-DAN model. The selection of ground motions for test data is objectively random with no predefined criteria.

3.1.2 Data Preparation

In this subsection, the data preparation for S-DAN is explored in detail, showing the shape and values of input data, as well as corresponding output values. It is noteworthy that the learning process follows a supervised approach, requiring the preparation of input and output pairs in advance. After that, the proposed network is trained to map given inputs to their respective outputs as closely as possible.

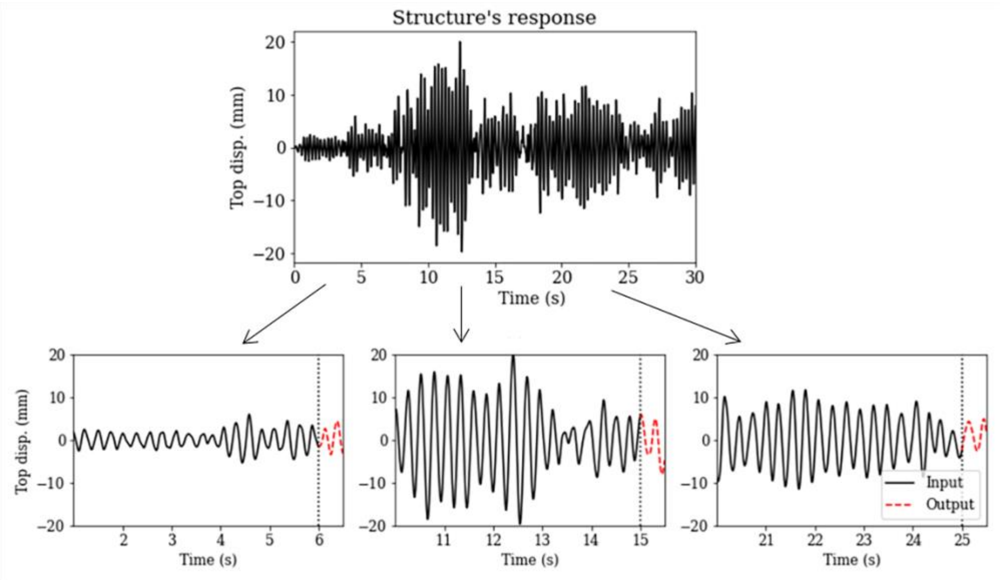


Fig. 6 Illustration of the window sliding technique employed to split original time-series into input sequences and respective outputs.

In order to prepare the dataset for training and validation of the S-DAN model, we carried out an extensive series of numerical simulations using the previously mentioned FEM in OpenSees with 10 different ground motions and 15 different scale factors ranging from 0.5 to 2.0 with an increment of 0.1. Subsequently, we applied the sliding window technique to the obtained numerical results to prepare labeled input/output pairs. Taking the time history of the top floor displacement as an example, Fig. 6 depicts its 3000-length time series obtained by numerical simulation with a total duration of 30 seconds and a sampling frequency of 100 Hz. The first 500-length records and their respective immediately subsequent 50-length time series constitute the first pair of input and output. Next, by shifting one time step, the sequence from 2nd to 501st time instants, and its following 50 time-step sequences, compose the second input/output pair. After that, by shifting a 500-length window one step each time from the beginning toward the end of an original 3000-length time series, one can obtain 2950 pairs of supervised data. Next, by combining the time history from all six floors and ground motion, a 3D tensor input of shape (2950, 500, 7) is formed, along with its corresponding 3D tensor output with a shape of (2950, 50, 1). In total, approximately 442500 input samples, form the training and validation database for S-DAN with a split ratio of 90:10.

3.1.3 Dynamic Time Warping Distance

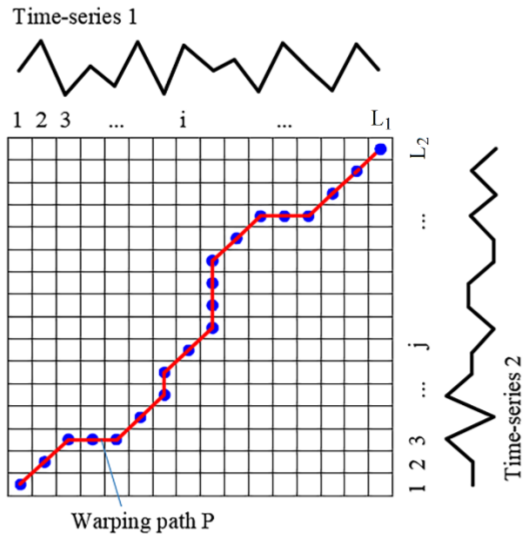


Fig. 7 Visual representation of the dynamic time warping path between two time series.

The problem involving earthquakes is inherently dynamic and non-linear; thus, it is nearly impossible in general cases to obtain an ideal solution that provides a perfect match between predicted values and actual ones. That is why comparing these values point-by-point might not properly assess the model performance. An informative alternative to evaluate the similarity between time series is using the dynamic time warping (DTW) distance, which is widely adopted in a range of applications [41]. The principle of the DTW distance can be briefly explained as follows. Given two time-series Y_1 with a length of L_1 and Y_2 with a length of L_2, we first calculate the Euclidean distance, a.k.a, the L_2 norm, between the first point of Y_1 and every points in Y_2. Next, we calculate the distances between the second point of Y_1 and all points in Y_2 except those of previous time instants. The process is realized in a monotonically increasing fashion. The same steps are then iterated for all points of Y_1. Afterwards, the first and second steps are repeated, but the roles of Y_1 and Y_2 are reversed. After completing these three steps, one obtains a matrix of Euclidean distance with a shape of $(L_1,\ L_2)$. After that, the path with minimum Euclidean distance going from the first position (1, 1) to the last position $(L_1,\ L_2)$ of the matrix is calculated. This path is referred to as the warping path; and its length is regarded as the DTW distance between Y_1 and Y_2 (Fig. 7). Algorithmically, the process mentioned above is automatically realized with the help of the library Fastdtw [42]. Here, one normalizes the DTW distance to estimate the similarity between two time series in a more general way regardless of their length and absolute amplitudes, as follows:

$$\overline{DTW} = \frac{DTW}{Y_1^{rms} \times L_1} \times 100\% \tag{7}$$

where $\overline{DTW}$ and DTW are the normalized and original dynamic time warping distances, respectively, Y1rms is the root mean square amplitude of the reference time series, and L_1 is its length. The $\overline{DTW}$ roughly provides a sense of how much the predicted time series relatively differs from the reference one. A small DTW indicates that these time series are similar. Especially if $DTW = 0$, they are perfectly identical. Otherwise, the larger DTW values, the more they differ from each other.

3.1.4 Forecasting Results

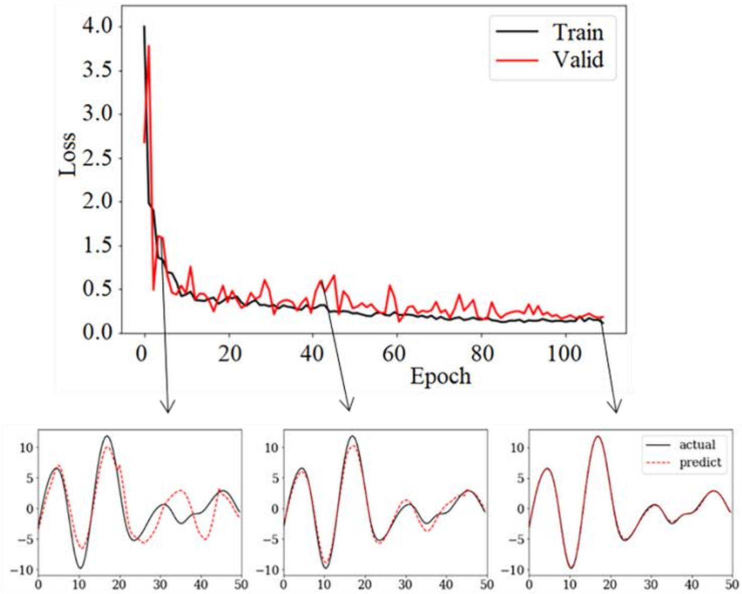


Fig. 8 Evolution of training and validation loss functions during the training process. At the bottom are shown forecasting results generated by the S-DAN model at different training stages.

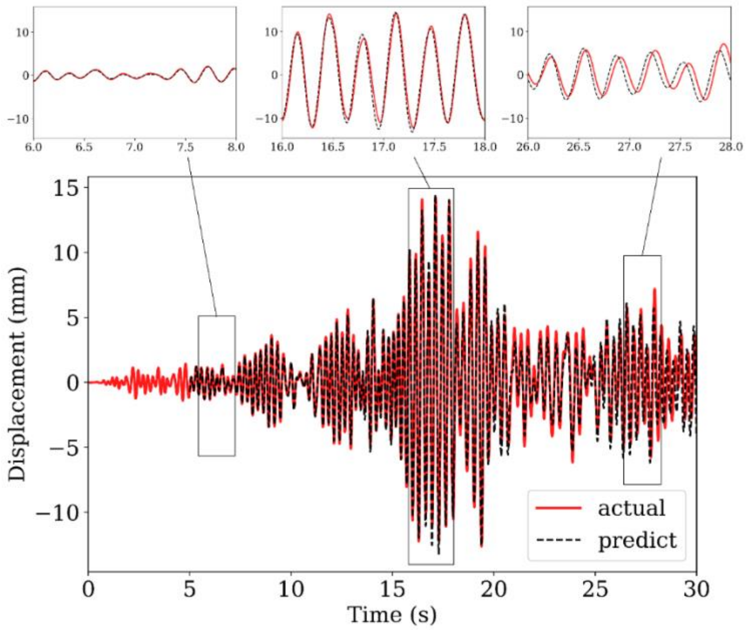


Fig. 9 A representative example of forecasting results of the top floor displacements of the structure under the Karditsa earthquake with a load factor of 1.0.

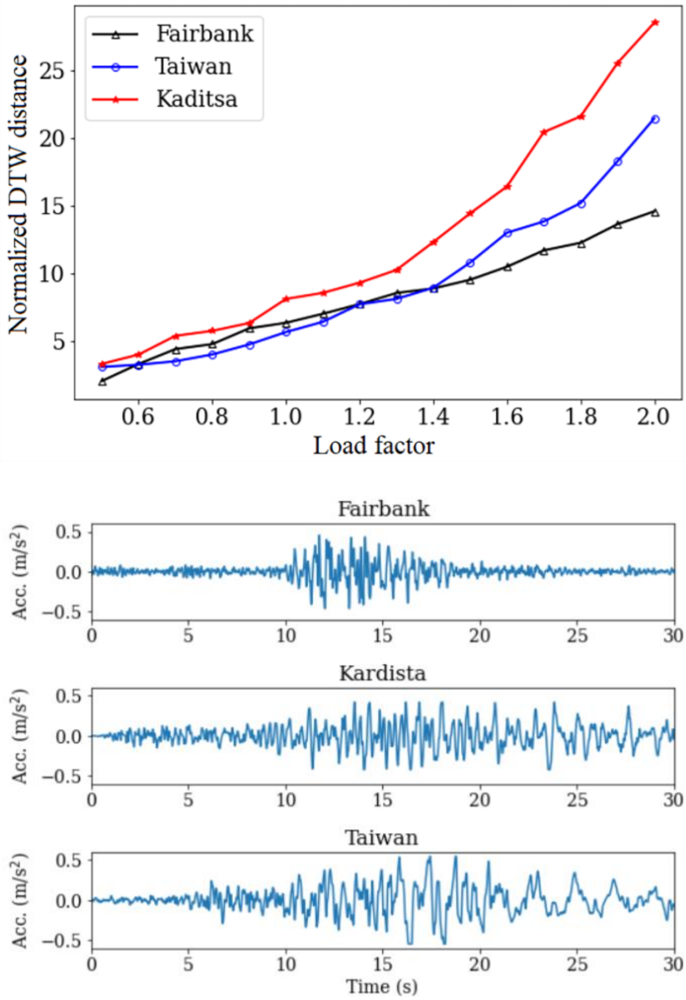


Fig. 10 Impact of the load factor on the normalized DTW distance (on the top) and examples of three ground motions from the testing data, namely, Fairbank, Karditsa, and Taiwan.

After preparing the database, the proposed approach is trained using the Adam optimizer with parameters such as a mini-batch size of 128, an initial learning rate of 10^{-4}, which is divided by 2 when the validation loss does not decrease. Early stopping is applied after ten consecutive epochs of non-decreasing validation loss. The training process stops after 110 epochs, as shown in Fig. 8. The figure shows that the value of the loss function drastically drops for the first ten epochs, followed by a gradually decreasing trend before becoming stable after around epoch 95. Although the validation loss fluctuates during the training process, it closely aligns with the training loss by the end, indicating that the overfitting problem is precluded to some extent. To highlight the significance of the loss value, the insets depict forecasting results obtained by the model trained at different epochs: 5, around 40, and 110. As the loss function reduces, forecasting results approach actual values, i.e., the model performance improves. In terms of computation time, the training process takes 181 minutes on a high-performance computer equipped with a 2080Ti GPU, Intel Xeon 4.3 GHz, and 32 GB RAM.

Next, one employs the trained model to predict the structure's response under unseen ground motions, i.e., Fairbank, Taiwan, and Karditsa, with different scale load factors. For each test case, the input data consist of the ground acceleration and the first 500 values of the top floor's vibrations computed by the finite element method (FEM). The remaining parts of the FEM results represent the actual responses against which predictions from S-DAN will be compared. The input data are fed into S-DAN, forecasting the next 50 values of the floors' vibration. Subsequently, these predicted 50 values are appended to the previous time-series vibrations, forming new 500-length time-series inputs and are used to predict the following 50 values. This process is repeated until the final time step is reached. Fig. 9 illustrates forecasting results for the test case with the Karditsa earthquake and a load factor of 1.0. In the figure, the red curve is the actual time series obtained by FEM, and the dashed black curve, starting from step 501, denotes the predicted results by S-DAN. It can be seen that there is a satisfying agreement between the results. More specifically, from time instant 6 s to around 8 s, a nearly perfect overlap between two curves is observed, as shown in the leftmost inset. However, as the excitation becomes stronger, deviations between results increase, as shown in the rightmost inset.

Table 3 Effect of the sliding window length on the performance of the S-DAN framework

Window length	**50/5**	**100/10**	**250/25**	**500/50**	**750/75**
Training time (min)	40.5	60	119	181.3	320
Validation loss (mm)	0.22	0.23	0.24	0.24	0.30
Inference time(*) (s)	55.7	29.0	12.5	7.2	
Window length	**1000/100**	**1500/150**	**2000/200**	**2500/250**	
Training time (min)	425	740	77.5	NA	
Validation loss	0.36	0.41	0.51	NA	
Inference time(*) (s)	4.5	3.7	3.3	NA	

(*): Inference time required for 60 s-long vibration responses.

Table 4 Effect of the LSTM output dimensionality on the performance of the S-DAN framework

N_{lstm}	**8**	**16**	**32**	**64**	**128**	**256**	**384**	**512**
Training time (min)	166.2	169.2	172.1	176.8	181.3	185.2	189.7	192.5
Validation loss (mm)	1.83	1.29	0.96	0.61	0.24	0.23	0.20	0.20
Inference time(*) (s)	7.15	7.18	7.20	7.26	7.27	7.28	7.30	7.32

(*) : Using window length 500/50.

Next, one applies S-DAN to the unseen test data and uses the normalized time warping distance to quantitatively estimate the model performance. It is acknowledged that the more important the external load applied to the structure, the higher the degree of non-linearity the structure's behavior will exhibit. In order to quantitatively assess the performance of the proposed S-DAN framework in handling non-linear behaviors, we test S-DAN with different excitations of various intensity degrees characterized by load factors. Fig. 10 plots the computed $\overline{DTW}$ for different load factors. The black curve with triangular symbols represents the Fairbank ground motion, the blue one with circles corresponds to the Taiwan ground motion, and the red one with star symbols correspond to the Karditsa ground motion. It can be seen that there is an increasing trend between DTW and the load factor. This is because stronger excitation induces more damage to the structure causing its responses to exhibit more non-linear and unusual patterns. These patterns may not be learned by the model, leading to larger forecasting errors. It is also noticed that DTW increases more rapidly with long-lasting high amplitude ground motions (Taiwan and Karditsa) than with those characterized by a short period of high intensity (Fairbank). Specifically, S-DAN can provide predicted results with a deviation of less than 10% in terms of amplitude at low load factors, (i.e., ≤ 1.2) for all testing ground motions. However, at a load factor of 2.0, the errors obtained with Karditsa are around twice that of Fairbank (28.2% vs. 14.8%). In summary, for weak excitation where the structures behave linearly, low prediction errors are obtained; when the excitation becomes stronger, the prediction errors increase. Later, we will compare the S-DAN method with competing methods to clarify its performance in predicting the structures' dynamic responses.

In fact, there are various LSTM variants, such as one-to-many, many-to-one, and many-to-many LSTM. This study adopts the many-to-many LSTM, which concatenates the output of LSTM cells into a new time series rather than considering only the output of the last LSTM cell. Besides, one of the key parameters of LSTM is the dimensionality of the LSTM cell output, denoted by N_{lstm}. This parameter defines the shape of the LSTM layer output as $[N_{batch} \times N_{lstm} \times T]$ where N_{batch} is the batch size, and T is the length of the time series. To investigate the effect of N_{lstm} on the model performance, one repeats the training process with N_{lstm} in the range [8, 512], and then the validation loss, training time, and inference time are compared. Tables 3 and 4 display the comparison results, showing that the validation loss RMS considerably decreases from 1.83 to 0.24 when increasing N_{lstm} from 8 to 128. After that, the loss marginally improves with N_{lstm} above 128. In contrast, the training time considerably rises with high values of N_{lstm}, e.g., the training time for N_{lstm}=512 is about 3.3 times higher than that for N_{lstm} = 128. Therefore, N_{lstm} = 128 is selected because it provides a good balance between performance and training time.

Besides, the sliding window technique is used to prepare training and validation data for the training process of the proposed approach. Hence, it is informative to investigate the effect of the window length on the model's performance and time complexity. Table 3 details the calculation results for different window lengths in the range of [50-2500]. Note that the ratio between input/output length is fixed to 10; for example, if the input length is 500, then the output length is 50. It can be seen that the longer the input length, the longer the training time, while the inference time becomes shorter. This is because longer window lengths require fewer recursive steps to fully predict the structure's response under a ground motion record. For example, the inference time significantly decreases from 23.6s to 3.05s for 50-length and 500- length sliding windows, respectively. However, using a long input may require a more complex model with wider or deeper neural network layers; otherwise, it can negatively impact the performance. For example, the validation losses are nearly similar for window lengths from 50 to 500 but decrease with increasing window lengths. Moreover, using large input data also necessitates a larger memory and storage footprint, which is not available (NA) on regular computers, as in the case of a window length of 2500. Based on these observations, one selects a window length of 500 for preparing datasets and building the surrogate model.

3.1.5 Comparison between S-DAN with counterparts

One compares S-DAN with three other methods widely used in the literature for forecasting problems, namely the statistic model Vector Autoregression (VAR), the machine learning algorithm Extreme Gradient Boosting (XGB), and the deep learning algorithm Long-Short Term Memory (LSTM). VAR is a generalized version of the popular autoregression model that aims to predict future values based on linear functions of historical ones. In this study, the implementation of VAR is realized with the help of the Statsmodels library [43]. The LSTM approach [33] is a variant of the Recurrent Neural Network adapted for long time series. Meanwhile, XGB, firstly introduced by Tianqi Chen [44], is now considered one of the most efficient and flexible machine learning algorithms acknowledged by several researchers. As the name suggests, the term "boost" means that XGB aggregates multiple models to outperform any single one, "Gradient" signifies that the gradient descent algorithm is used during the training process to minimize model errors. "Extreme" denotes that XGB is designated to work in a highly parallel way to utilize the hardware resources efficiently. Note that for a fair comparison, the input and output are the same for all methods, i.e., using 500 steps of historical data plus known excitations to predict 50 steps ahead of the time-series output. More specifically, considering a current time instant t, the input data consist of previously computed values of output $Y[t-499], \dots, Y[t]$, known excitation $F[t-449], \dots, F[t+50]$, and also known time-series from other sensors $X[t-449], \dots, X[t+50]$, if available. Meanwhile, the prediction outputs are $Y[t+1], \dots, Y[t+50]$. In order to ensure a fair comparison between methods, hyperparameter optimization was carried out in a preliminary study for selecting an adequate set of hyperparameters for each considered machine learning algorithm. The Bayesian Optimization technique and the practical GPyOpt library were employed for this purpose. A small sub-dataset, approximately one-tenth the size of the original database was randomly selected in advance, to conduct the hyperparameter optimization step. Deeper explanations and technical details about hyperparameter optimization can be found in [45]. The adopted values of the hyperparameters are enumerated in Table 5.

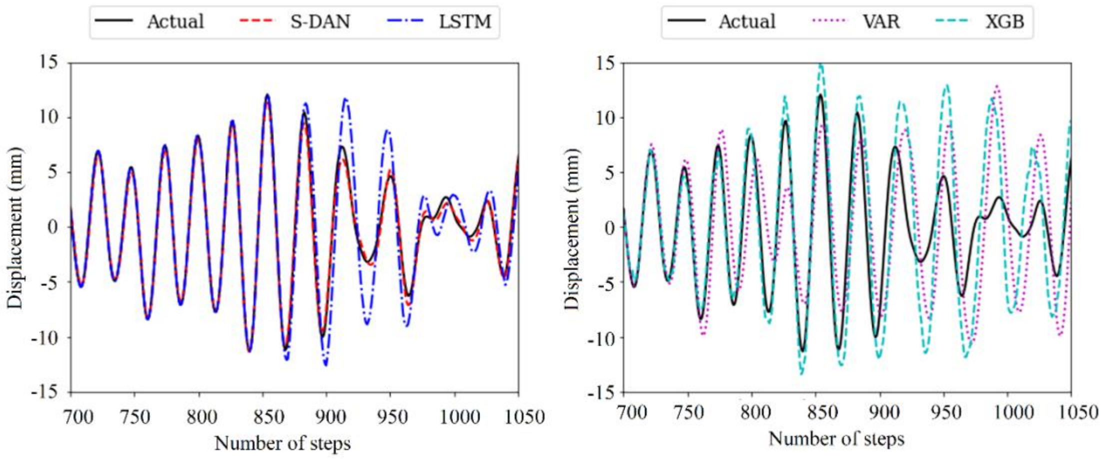


Fig. 11 Prediction results obtained by S-DAN, VAR, XGB, LSTM, and FEM.

Table 5 Hyperparameters of machine learning algorithms.

Method	Hyperparameters
S-DAN	L_in : 500, L_out : 50, N_lstm : 128, N_hidden : 64, learning rate: 0.001, batch size: 512
LSTM	L_in : 500, Lout : 50, N_lstm : 128, learning rate: 0.001, batch size: 512
SVR	model order: 500
XGBoost	learning rate: 0.12, maximum depth: 4, Number of tree: 350

Table 6 Comparison results between S-DAN and VAR, XGB, LSTM for the first case study involving a 3D RC frame structure.

Ground motion	Metric	S-DAN	LSTM	XGB	VAR
Fairbank	$\overline{DTW}$ (%)	6.3 ±0.3	10.8 ±0.65	25.2 ±0.3	50.8 ±0.0
	MSE (mm)	0.21±0.017	0.38±0.03	1.3±0.01	2.9 ±0.0
	MAE (mm)	0.27±0.014	0.46±0.02	1.17±0.01	1.49 ±0.0
	MAPE (%)	12.9±0.9	30.0±2.01	45.2±1.5	64.5 ±0.0
Kardista	$\overline{DTW}$ (%)	5.6±0.3	9.3±0.54	29.6±0.7	63.2 ±0.0
	MSE (mm)	0.24±0.017	0.47±0.025	0.95±0.03	1.93 ±0.0
	MAE (mm)	0.37±0.018	0.59±0.02	1.02±0.02	1.58 ±0.0
	MAPE (%)	11.3±1.02	33.1±1.05	39.8±0.98	51.2 ±0.0
Taiwan	$\overline{DTW}$ (%)	8.0±0.71	13.3±0.9	35.1±1.1	68.0±0.0
	MSE (mm)	0.19±0.013	0.32±0.019	0.95±0.02	1.53±0.0
	MAE (mm)	0.31±0.018	0.48±0.021	0.72±0.015	0.98±0.0
	MAPE (%)	10.9±0.9	31.4±1.7	43.2±1.2	57.3±0.0
Training time (min)		181.3	165.5	26.8	4.6
Inference time*(s)		7.2	5.7	2.4	0.6

(*): Inference time required for 60 s-long vibration responses.

Fig. 11 presents enlarged forecasting results from steps 750 to 1100 (7.5s to 11s) for an example of the Karditsa earthquake and a load factor of 1.0. Results from five methods S-DAN, LSTM, VAR, XGB, and FEM, are displayed in dashed red, dash-dot blue, dotted green, dashed cyan, and solid black curves, respectively. It can be seen that initially, there is a good consistency between results up to step 800. From around step 800, the errors of VAR become apparent and become more pronounced, while XGB maintains relatively good accuracy until step 1000. After that, significant discrepancies between XGB and FEM are observed. On the other hand, deviations between LSTM and FEM are considerably lower than those of XGB. Meanwhile, the curve of S-DAN approximately coincides with that of FEM throughout the whole interval being considered. Moreover, Table 6 shows various measurement metrics, including $\overline{DTW}$, MSE, Mean Absolute Error (MAE), and mean absolute percentage error (MAPE) obtained by these four methods for testing data. It can be seen that S-DAN achieves the lowest values, i.e., the best forecasting results. The error made by S-DAN is, on average, only about two-thirds that of the second-best method (LSTM) in terms of DTW . However, in terms of CPU times, S-DAN requires 9.5% more training time than LSTM. Meanwhile, VAR is very fast, but its error is too high; XGB, despite its fast training time, improves the prediction results, but its accuracy is still substantially lower than those of LSTM and S-DAN. In short, the results confirm the outperformance of S-DAN compared to currently used methods in forecasting civil structure's responses. Note that though the case study focuses on forecasting the top floor's response, it is straightforward to create another variant of S-DAN for other floors' displacements. This can be done by preparing corresponding data with outputs being the time series of interest and input being excitations and other floors' historical time series.

In terms of model complexity, as most the reviewed ML-based methods for forecasting the structures' dynamic responses do not explicitly provide the number of trainable parameters, one compares the total number of parameters in the S-DAN method with that of a conventional single hidden layer MLP network to gain insight into the S-DAN's model complexity. With $N = 6, M = 1, T = 500, \tau = 50, N_{lstm} = 128, N_{hidden} = 64,$ according to Table 2, the number of trainable parameters in the S-DAN method is around 7×7+4×128× 129+3×128×64+500×50 = 115673. Meanwhile, the number of parameters in the MLP network with an architecture of [3500, 64, 50] is 3500×64+64×50 = 227200. The [3500, 64, 50] architecture corresponds to an input layer with 3500 neurons for 7 time series of length 500, a hidden layer of 64 neurons, and an output layer with 50 neurons representing a 50-length output. Thus, it can be seen that the proposed method possesses a reasonable complexity, requiring half the number of parameters compared to the conventional MLP counterparts.

3.2 Case Study 2: Experimental Data of 18-story Steel Frame Structure

3.2.1 Experimental Data Description

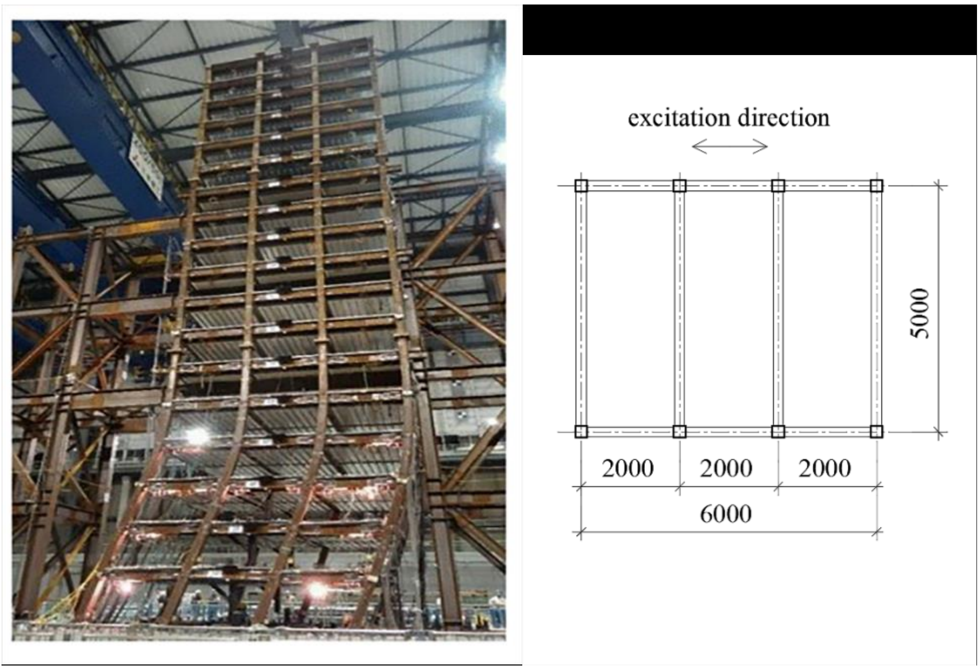


Fig. 12 Representation of the 18-story two-bay frame structure from [46].

In this subsection, the proposed method is applied to experimental data from a high-rise steel frame structure prone to ground motions, realized at the Hyogo Earthquake Engineering Research Center [46]. In analogy to the first example, the top floor acceleration will be predicted using ground motions and measured accelerations on other floors. The frame has 18 stories with a total height of 25.35 m, three spans of 2 m width in the loading direction, and one 5 m span in the other direction, as shown in Fig. 12. The columns are constructed from built-up hollow sections, while the beams are I-shaped and welded to the columns. The total weight of the structure is approximately 4200 kN. The structure is subjected to ground motions with characteristics of earthquake waves recorded at the Tokyo Shiba Elementary school by MeSOnet in 2011. Furthermore, nine levels of amplitudes were used, corresponding to pseudo spectral velocities (PSV) within the range of [40, 81, 110, 180, 220, 250, 300, 340, 420] cm/s. These excitations will induce various damages to the structure, such as yielding at beam ends, fracture, local buckling of columns, global buckling at lower stories, and eventually, a collapse mechanism.

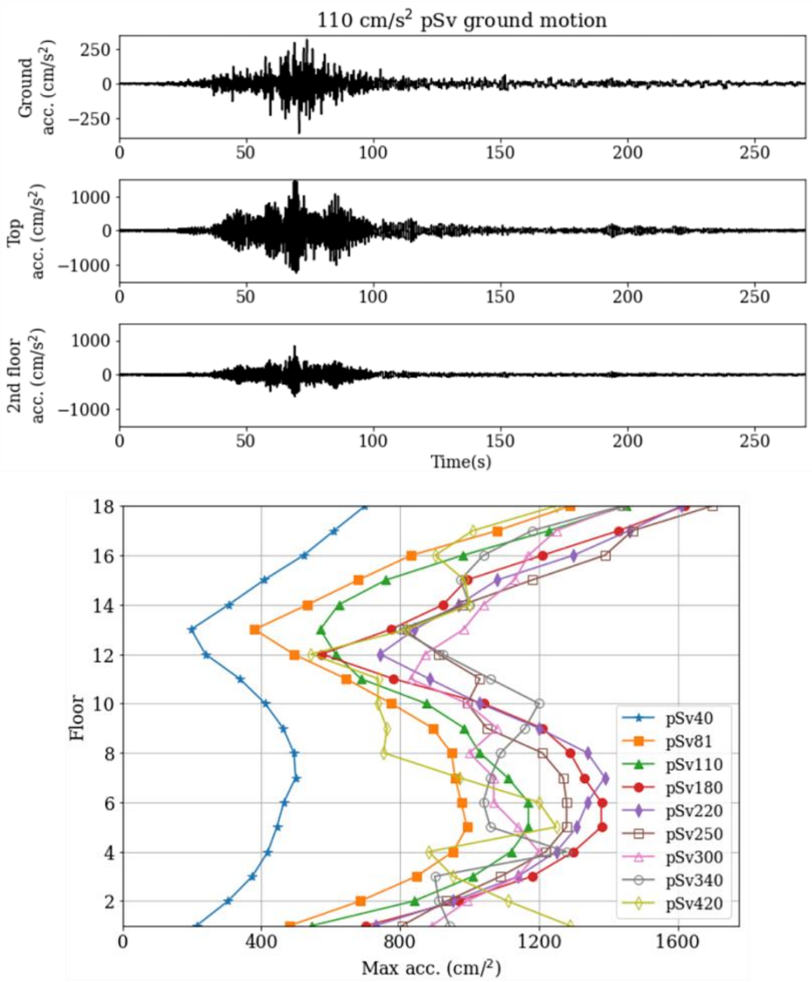


Fig. 13 Representative examples of experimentally measured data featuring a 110 cm/s2- PSV ground motion and the corresponding structure's accelerations at the top and second floor. At the bottom are shown peak accelerations recorded at all 18 floors for ground motions of different intensities.

Figure 13 represents the 110 cm/s^2 PSV-ground motion acceleration on the left and the corresponding measured vibration signals at the 18th and 2nd floors at the bottom left corner. Furthermore, the variation of peak accelerations recorded at all 18 floors caused by different ground motion intensities is depicted on the right. It can be seen that the essential part of the signals is between [30-120] s, while before and after this range, vibration amplitudes are insignificant. Therefore, one only considers the 30-120 s segment of the time series, significantly reducing computational costs in terms of both time and memory. With a sampling frequency of 100 Hz, this segment of interest has a total length of 9000.

3.2.2 Data Preparation

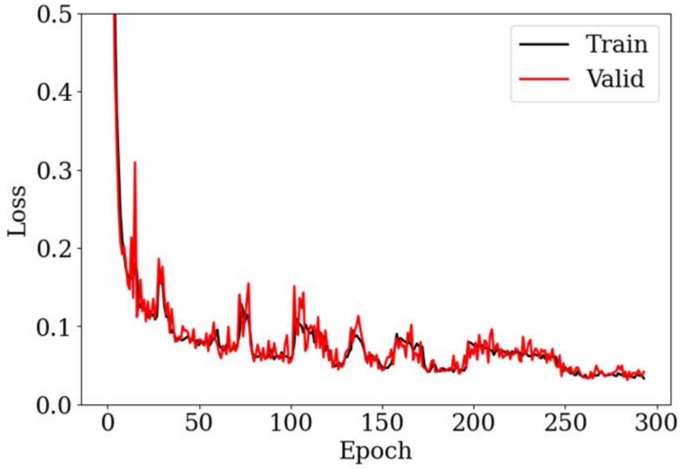


Fig. 14 Evolution of training and validation loss functions during the training process for the second case study.

Among experimental signals, those corresponding to three PSV values [110, 220, 340] cm/s2 are separated and regarded as test data unseen by S-DAN during the training process. The other signals of [40, 81, 180, 250, 300, 420] cm/s^2 PSV are grouped into training data. Similarly to the first example, the sliding window technique was employed to prepare the required vibration database. Each vibration signal is divided into multiple 500-length sub-time series accompanied by their subsequent 50-length time-series, forming input and output pairs. After that, one combines signals from different floors and ground motions to form multivariate inputs for the S-DAN model. Given a 9000-length signal, with one step forward each time, we can obtain about 8450 input/output pairs. With 9 levels of ground motions, the total number of data in the database N_{sample}, is around 76050 samples. The shape of the input data is (N_{sample}, 19, 500), where N_{sample} is the number of samples in the input database. The shape of the output data is (N_{sample}, 1, 50). It is noted that the ground motion is known in advance by S-DAN. After that the S-DAN model is trained and validated with this database, and its learning curves are presented in Fig. 14.

3.2.3 Forecasting Results

Next, the performance of the trained model is assessed on unseen testing data. Fig. 15 displays forecasting results obtained for the top floor's accelerations. The first row shows experimentally measured signals in red, and the second row represent results obtained by S-DAN. In the figure, the red parts from 30 s to 45 s denote an initial warm-up period, and the black parts from 45 s to 120 s represent multiple-step predicted outputs. The third row of the figure magnifies the comparison between predicted results and experim5545ental ones in the range [90 s-100 s]. Overall, good consistency between results is achieved, especially for low PSV values, i.e., 110 cm/s^2. On the other hand, for higher PSV, discrepancies become more apparent as the structure is damaged and exhibits non-linear behavior, i.e., corresponding acceleration signals showing more irregularities. For example, in the case of the 340 cm/s^2 PSV ground motion, the peak of the structure's responses between 70 s-80 s is not captured by S-DAN.

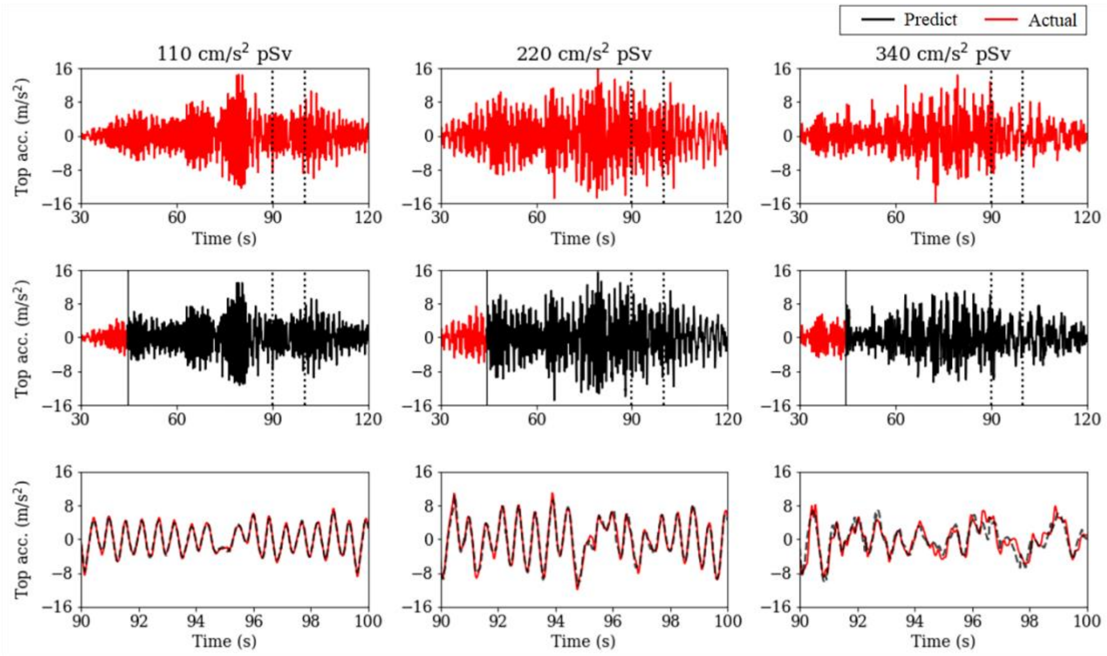


Fig. 15 Predictive results of the top floor's vibration obtained by using the S-DAN framework. The experimental accelerations and the corresponding forecasting results are displayed in the first and second rows, respectively. To facilitate visual comparison, the signal part from the 90s to the 100s is magnified.

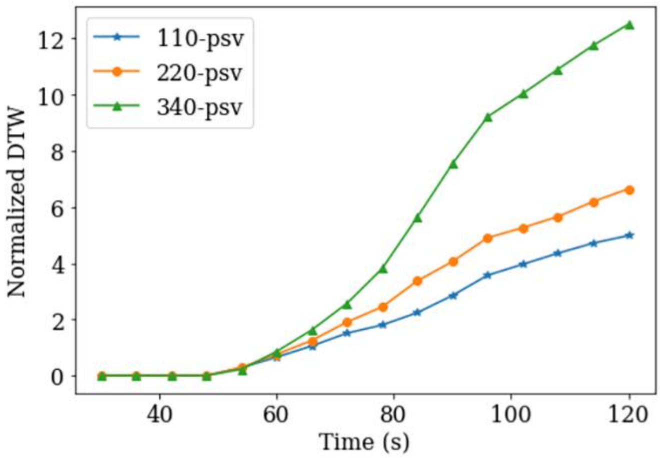


Fig. 16 Influence of time-series length on the obtained DTW distance.

Fig. 16 illustrates the time evolution of DTW between forecasting results and experimental ones for three test cases. Because the first 15 s of experimental signals are used as input, there is no difference between results, i.e., DTW = 0 for this interval. Afterward, DTW proportionally increases with the length of the predicted time series due to error accumulation. The DTW curves exhibit different slopes for various ground motion intensities. Specifically, the DTW curve associated with 340 PSV-excitation rises sharply, and at the end of the time series, DTW is nearly 2.0 and 2.5 times those of 220 PSV-excitation and 110 PSV-excitation, respectively (12.5 vs. 6.5 and 5.0). Furthermore, when considering a single DTW curve, the portion corresponding to the strongest period [60 s-90 s] of the ground motion has a steeper slope than other portions.

3.2.4 Comparison between S-DAN with Counterparts

Analogous to the first case study, the performance of S-DAN is directly compared with other popular methods, namely VAR, XGB, and LSTM, to demonstrate its accuracy and efficiency. It is recalled that the excitations are known in advance and included in input data for all methods. Table 7 enumerates comparison results using four measurement metrics, i.e., DTW , MSE, MAE, and MAPE on testing data, which shows a significant improvement in accuracy achieved by S-DAN. Specifically, DTW of S-DAN is around 45% lower on average compared to that of the second-best method, LSTM, and substantially lower than results from XGB and VAR. It is noted that for highly nonlinear case such as the 340-PSV ground motion, which leads to buckling at lower stories of the structure, the errors commit by the S-DAN framework are lower than those of the competing methods by a clear margin. Specifically, the DTW by S-DAN is 12.5%, compared to 20.5%, 36.5% and 51.5% for LSTM, XGB, VAR, respectively. The superiority of S-DAN over its counterparts can be rationalized as follows: VAR is basically a linear statistical method, making it suitable for situations where the structure behaves in the elastic range. However, when plasticity or damage occurs, a linear method is no longer adequate. While the XGB algorithm can perform a non-linear mapping between time-series input and output, it does not account for the chronological relationship, which is one of the most important features of time-series data. The LSTM algorithm can take into account both non-linear behavior and chronological connectivity, thus providing reasonable results. However, LSTM only exploits features of long-term relationships through its cell state values lying between [0, 1]. The S-DAN method, on the other hand, allows for capturing richer temporal information, including the importance (key vector), the appropriateness (query vector), and amplitude (value vector), as explained in the Methodology section. That is why the attention mechanism significantly boost forecasting accuracy.

Table 7 Comparison results between S-DAN and VAR, XGB, LSTM for the second case study.

Ground motion	Metric	S-DAN	LSTM	XGB	VAR
110-PSV	$\overline{DTW}$ (%)	5.0±0.2	9.9±0.54	19.9±0.36	27±0.0
	MSE (mm)	0.22±0.02	0.48±0.034	1.95±0.03	2.93±0.0
	MAE (mm)	0.37±0.015	0.53±0.026	1.07±0.014	1.36±0.0
	MAPE (%)	9.3±0.41	19.2±1.15	38.6±1.0	52.7±0.0
220-PSV	$\overline{DTW}$ (%)	6.5±0.32	12.1±0.73	22.9±0.45	35.5±0.0
	MSE (mm)	0.25±0.013	0.35±0.02	0.93±0.037	1.43±0.0
	MAE (mm)	0.36±0.014	0.57±0.04	1.09±0.016	1.65±0.0
	MAPE (%)	8.9±0.44	14.9±0.098	33.1±1.02	49.5±0.0
330-PSV	$\overline{DTW}$ (%)	12.5±0.65	20.5±1.43	36.5±1.1	51.5±0.0
	MSE (mm)	0.7±0.025	1.4±0.12	2.6±0.1	3.7±0.0
	MAE (mm)	0.59±0.026	0.91±0.041	1.79±0.02	2.05±0.0
	MAPE (%)	19.2±1.34	26.0±2.65	41.7±1.46	62.1±0.0
Training time (min)		160.4	110.3	12.1	2.3

Inference time*(s)	16.5	11.6	5.2	1.2

(*): Inference time required for 90 s-long vibration responses.

In terms of time complexity, during the training process, S-DAN requires the longest training time, up to 160.4 minutes, which is approximately 1.5 times longer than that of LSTM. The XGB and VAR methods require training times that are one and two orders of magnitude faster than S-DAN, respectively. Nevertheless, at inference time, it takes only a few seconds for S-DAN to forecast a 9000-length time series.

3.2.5 Robustness Study

In the following part, the robustness of S-DAN is investigated concerning time series input data contaminated by noise. Herein, one considers white noise, which is a classical yet essential problem in time series analysis. The noise amplitude is defined based on the root mean square (RMS) value of the vibration data, as follows:

$$X_{noise} = X + \alpha \times \eta \tag{8}$$

where X is the measured vibration signal, X_{noise} is added-noise data, η is the white noise vector with zero mean and unit variance, and α is the noise amplitude dependent on the RMS of the original data. The noise effect study consists of the following realization steps: i) accelerations of floors 1 to 17 are contaminated by external noises; thus, the input data consist of noisy vibration data plus original ground motion acceleration (no noise), as it is supposed that the excitation is well-controlled in laboratory conditions. ii) The noisy data are fed into the S-DAN model to predict the top floor acceleration as done above. iii) Next, DT W between computed results and experimental ones is calculated. In the civil engineering, a noise level in the range of [2%-10%] is usually considered for different applications, e.g., dynamic structural analysis [47], and damage detection [48]. In practice, such noise may be caused by environmental factors (temperature, humidity), sensor sensibility, and transmission instability but does not cover systematic errors such as human errors or device inaccuracies. In this work, we investigate the impact of noises with amplitude α in a range of (0% up to 20%).

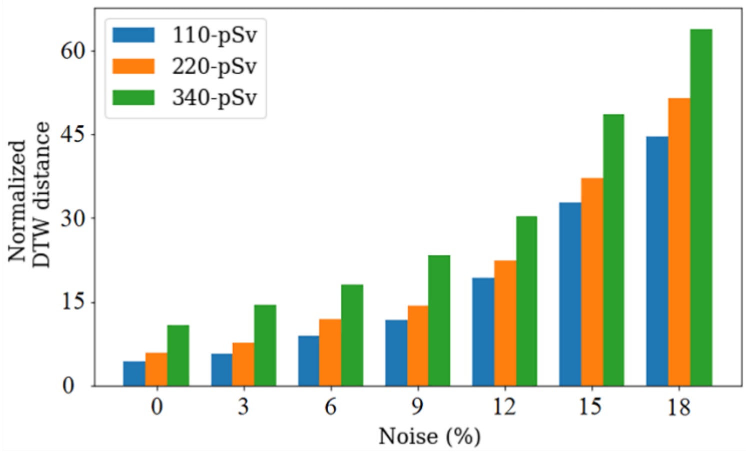


Fig. 17 Impact of noise on the normalized DTW distance between forecasted and experimental results

Fig.17 depicts the evolution of DT W versus the noise amplitude obtained by applying the S-DAN approach to testing data. Under low- and moderate-intensity excitations (110-PSV and 220-PSV), S-DAN can provide controlled prediction results with DT W of less than 15% for a noise level of 9%. However, with stronger earthquakes (340- PSV excitations), prediction performance starts to suffer more from errors, reaching around 20% for a noise level of 9%. It is noted that prediction results with relative errors of no more than 20% are still widely considered acceptable for structures' seismic nonlinear responses [22]. These results are obtained by using a model trained with original data (without noise) and then tested with noisy data. In order to further improve the model's noise robustness, some strategies such as noise injection, data augmentation, or enhancing S-DAN with a noise filter could be applied.

In summary, this case study has demonstrated that S-DAN outperforms counterpart methods in terms of accuracy, can deliver forecasting results with a fast inference time of a few seconds, and is generally robust against noise with amplitudes of less than 10%.

4 Conclusion

In this study, a data-driven method for multi-step forecasting the responses of structures under time-varying excitation was developed. Throughout the manuscript, different aspects of the proposed S-DAN framework were explicitly presented, including the overall workflow, the underlying intuition of capturing inter- and intra-relationships between multiple time-varying signals, data preparation using the window sliding technique, the deep learning architecture featuring a dual-attention mechanism, algorithm description via pseudocode, and implementation details. The viability of the proposed method was quantitatively demonstrated through two case studies involving synthetic data from a 3D reinforced concrete frame structure and experimental data from an 18-story steel frame structure. The obtained results proved that the S-DAN method consistently outperforms competing approaches including LSTM, XGB, and VAR as the normalized DTW distance between the actual responses with those predicted by S-DAN is significantly lower than those of other approaches. Furthermore, additional studies providing more insights into the performance of the proposed method were carried out: i) S-DAN could maintain good prediction accuracy with input data disturbed by noise with an amplitude of less than 10% of their RMS; ii) for the elastic regime when structures are subjected to low-intensity excitation, predicted results nearly coincide with actual results; however, in the highly nonlinear regime as in the case of structures subjected to the high-intensity earthquake, higher prediction errors would occur; iii) investigating the trade-off between accuracy and efficiency when using long input data. More specifically, using long input data reduces the number of recursive steps, thus shortening the inference time and reducing the risk of error accumulation, but it increases model complexity with a significantly higher number of parameters.

Although achieving promising results, the current version of S-DAN still has two limitations that should be improved in the next study step to increase its practicality. The first limitation is that the inference time is still higher than that of competing methods because the attention mechanism is cubically proportional to the input data length. Therefore, exploring new variants of attention mechanisms such as flash attention, spare attention, and fast attention could be beneficial to reduce computational resources. Secondly, the robustness of S-DAN against noise with an amplitude greater than 10% should be improved possibly by combining it with a denoising autoencoder component. This component can reconstruct clean vibration signals from noisy ones, before passing through the S-DAN model for predicting the structures' responses.

Acknowledgement
This work was financially supported by the Hanoi University of Civil Engineering (Vietnam), ID 28{2023/KHXD-TD.

Data Availability. Source code and processed data are available from the corresponding author upon reasonable request

Declaration of interests. 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.

References

[1] Ghandourah, E., Bendine, K., Khatir, S., Benaissa, B., Banoqitah, E.M., Alhawsawi, A.M., Moustafa, E.B. Novel approach-based sparsity for damage localization in functionally graded material. Buildings, 2023, 13(7), 1768

[2] Benaissa, B., Hocine, N.A., Khatir, S., Riahi, M.K., Mirjalili, S. Yuki algorithm and pod-rbf for elastostatic and dynamic crack identification. Journal of computational science, 2021, 55, 101451

[3] Al Thobiani, F., Khatir, S., Benaissa, B., Ghandourah, E., Mirjalili, S., Wahab, M.A. A hybrid pso and grey wolf optimization algorithm for static and dynamic crack identification. Theoretical and applied fracture mechanics, 2022, 118, 103213

[4] Khatir, A., Capozucca, R., Khatir, S., Magagnini, E., Benaissa, B., Le Thanh, C., Wahab, M.A. A new hybrid pso-yuki for double cracks identification in cfrp cantilever beam. Composite Structures, 2023, 311, 116803

[5] Ho, L.V., Trinh, T.T., De Roeck, G., Bui-Tien, T., Nguyen-Ngoc, L., Wahab, M.A. An efficient stochastic-based coupled model for damage identification in plate structures. Engineering Failure Analysis, 2022, 131, 105866

[6] Ghandourah, E., Khatir, S., Banoqitah, E.M., Alhawsawi, A.M., Benaissa, B., Wahab, M.A. Enhanced ann predictive model for composite pipes subjected to low-velocity impact loads. Buildings, 2023, 13(4), 973

[7] Benaissa, B., Khatir, S., Jouini, M.S., Riahi, M.K. Optimal axial-probe design for foucault-current tomography A global optimization approach based on linear sampling method. Energies 16(5), 2023, 2448

[8] Tran, V.-T., Nguyen, T.-K., Nguyen-Xuan, H., Wahab, M.A. Vibration and buckling optimization of functionally graded porous microplates using bcmo-ann algorithm. Thin-Walled Structures, 2023, 182, 110267

[9] Dang, B.-L., Nguyen-Xuan, H., Wahab, M.A. An effective approach for varans-vof modelling interactions of wave and perforated breakwater using gradient boosting decision tree algorithm. Ocean Engineering, 2023, 268, 113398

[10] Nguyen, T.-T., Dang, V.-H., Nguyen, H.X. Efficient framework for structural reliability analysis based on adaptive ensemble learning paired with subset simulation. Structures, 2022, 45, 1738-1750
[11] Dang, H.V., Trestian, R., Bui-Tien, T., Nguyen, H.X. Probabilistic method for time-varying reliability analysis of structure via variational bayesian neural network. Structures, 2021, 34, 3703-3715
[12] Wang, S., Wang, H., Zhou, Y., Liu, J., Dai, P., Du, X., Wahab, M.A. Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching. Measurement, 2021, 169, 108362
[13] Nguyen, D.H., Wahab, M.A. Damage detection in slab structures based on two-dimensional curvature mode shape method and faster r-cnn. Advances in Engineering Software, 2023, 176, 103371
[14] M¨oller, B., Reuter, U. Prediction of uncertain structural responses using fuzzy time series. Comput. Struct., 2008, 86(10), 1123-1139
[15] Zhang, R., Chen, Z., Chen, S., Zheng, J., B¨uy¨uk¨ozt¨urk, O., Sun, H. Deep long short-term memory networks for nonlinear structural seismic response prediction. Computers & Structures, 2019, 220, 55-68
[16] Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V.M., Guo, H., Hamdia, K., Zhuang, X., Rabczuk, T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning Concepts, implementation and applications. Computer Methods in Applied Mechanics and Engineering, 2020, 362, 112790
[17] Guo, H., Zhuang, X., Rabczuk, T. A deep collocation method for the bending analysis of kirchhoff plate. arXiv preprint arXiv2102.02617, 2021
[18] Zhuang, X., Guo, H., Alajlan, N., Zhu, H., Rabczuk, T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of kirchhoff plates with transfer learning. European Journal of MechanicsA/Solids, 2021, 87, 104225
[19] Guo, H., Zhuang, X., Chen, P., Alajlan, N., Rabczuk, T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. Engineering with Computers, 2022, 38(6), 5173-5198
[20] Guo, H., Zhuang, X., Chen, P., Alajlan, N., Rabczuk, T. Analysis of three-dimensional potential problems in non-homogeneous media with physics-informed deep collocation method using material transfer learning and sensitivity analysis. Engineering with Computers, 2022, 38(6), 5423-5444
[21] Zhang, R., Liu, Y., Sun, H. Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. Engineering Structures, 2020, 215, 110704
[22] Oh, B.K., Park, Y., Park, H.S. Seismic response prediction method for building structures using convolutional neural network. Structural Control and Health Monitoring, 2020, 27(5), 2519
[23] Yu, Y., Yao, H., Liu, Y. Structural dynamics simulation using a novel physics-guided machine learning method. Engineering Applications of Artificial Intelligence, 2020, 96, 103947
[24] Xu, Y., Lu, X., Cetiner, B., Taciroglu, E. Real-time regional seismic damage assessment framework based on long short-term memory neural network. Computer-

Aided Civil and Infrastructure Engineering, 2021, 36(4), 504-521
[25] Du, X., Ma, C., Zhang, G., Li, J., Lai, Y. K., Zhao, G., ... & Wang, H. An efficient LSTM network for emotion recognition from multichannel EEG signals. IEEE Transactions on Affective Computing, 2020, 13(3), 1528-1540.
[26] Gao, Y., Ruan, Y. Interpretable deep learning model for building energy consumption prediction based on attention mechanism. Energy and Buildings, 2021, 252, 111379
[27] Liu, C., Zhang, L., Niu, J., Yao, R., Wu, C. Intelligent prognostics of machining tools based on adaptive variational mode decomposition and deep learning method with attention mechanism. Neurocomputing, 2020, 417, 239-254
[28] Zhang, Y., Chen, S., Cao, W., Guo, P., Gao, D., Wang, M., Zhou, J., Wang, T. Mffnet Multi-dimensional feature fusion network based on attention mechanism for semg analysis to detect muscle fatigue. Expert Systems with Applications, 2021, 185, 115639
[29] Kong, F., Li, J., Jiang, B., Wang, H., & Song, H. Integrated generative model for industrial anomaly detection via Bidirectional LSTM and attention mechanism. IEEE Transactions on Industrial Informatics, 2021, 19(1), 541-550.
[30] Hsu, C.-Y., Lu, Y.-W., Yan, J.-H. Temporal convolution-based long-short term memory network with attention mechanism for remaining useful life prediction. IEEE Transactions on Semiconductor Manufacturing, 2022, 35(2), 220-228
[31] Sun, S., Liu, J., Wang, J., Chen, F., Wei, S., & Gao, H. Remaining useful life prediction for AC contactor based on MMPE and LSTM with dual attention mechanism. IEEE Transactions on Instrumentation and Measurement, 2022, 71, 1-13.
[32] He, J., Yang, H., Zhou, S., Chen, J., Chen, M. A dual-attentionmechanism multi-channel convolutional lstm for short-term wind speed prediction. Atmosphere, 2022, 14(1), 71
[33] Hochreiter, S., Schmidhuber, J. Long short-term memory. Neural Computation, 1997, 9(8), 1735-1780
[34] Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F. Transformers are rnns Fast autoregressive transformers with linear attention. arXiv preprint arXiv2006.16236, 2020
[35] Fran¸cois, C. Keras the Python deep learning API. https//keras.io/. Accessed 2020-08-15
[36] Wang, C., Xiao, J., Sun, Z. Seismic analysis on recycled aggregate concrete frame considering strain rate effect. International Journal of Concrete Structures and Materials, 2016, 10(3), 307-323
[37] Filippou, F.C., Popov, E.P., Bertero, V.V. Effects of bond deterioration on hysteretic behavior of reinforced concrete joints, 1983
[38] Hisham, M., Yassin, M. Nonlinear analysis of prestressed concrete structures under monotonic and cycling loads. University of California, Berkeley. Ph. D. thesis, 1994
[39] McKenna, F. Opensees a framework for earthquake engineering simulation. Computing in Science & Engineering, 2011, 13(4), 58-66
[40] Haddadi, H., Shakal, A., Stephens, C., Savage, W., Huang, M., Leith, W.,

Borcherdt, R. Center for engineering strong-motion data (cesmd). In Proceedings of the Fourteenth World Conference on Earthquake Engineering, 2008
[41] Marteau, P.-F. Time warp edit distance with stiffness adjustment for time series matching. IEEE Trans. Pattern Anal. Mach. Intell., 2008, 31(2), 306-318
[42] Salvador, S., Chan, P. Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis, 2007, 11(5), 561-580
[43] Seabold, S., Perktold, J. statsmodels Econometric and statistical modeling with python. In 9th Python in Science Conference, 2010
[44] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y. Xgboost extreme gradient boosting. R package version 0.4-2, 1-4, 2015
[45] The GPyOpt authors, Gpyopt A bayesian optimization framework in python. Retrieved from http//github.com/SheffieldML/GPyOpt, 2016
[46] Nakashima, M., Kurata, M., Rathje, E., Sharifi Mood, M. Quantification of Collapse Margin for Steel High-rises. https//doi.org/10.17603/DS2T375. Accessed 2021-06-15
[47] Xu, H., Ren, W.-X., Wang, Z.-C. Deflection estimation of bending beam structures using fiber bragg grating strain sensors. Advances in Structural Engineering, 2015, 18(3), 395-403
[48] Dang, V.-H., Vu, T.-C., Nguyen, B.-D., Nguyen, Q.-H., Nguyen, T.-D. Structural damage detection framework based on graph convolutional network directly using vibration data. Structures, 2022, 38, 40-51