Journal of Science and Technology in Civil Engineering, HUCE (NUCE), 2022, 16 (2): 106–116

PREDICTING DYNAMIC RESPONSES OF FRAME STRUCTURES SUBJECTED TO STOCHASTIC WIND LOADS USING TEMPORAL SURROGATE MODEL

Dang Viet Hung[a], Nguyen Truong Thang[a,*]

[a]*Faculty of Building and Industrial Construction, Hanoi University of Civil Engineering, 55 Giai Phong road, Hai Ba Trung district, Hanoi, Vietnam*

Article history:
Received 01/3/2022, Revised 07/4/2022, Accepted 13/4/2022

Abstract

Determining structures' dynamic response is a challenging and time-consuming problem because it requires iteratively solving the governing equation of motion with a significantly small time step to ensure convergent results. This study proposes an alternative approach based on the deep learning paradigm working in a complementary way with conventional methods such as the finite element method for quickly forecasting the responses of structures under random wind loads with reasonable accuracy. The approach works in a sequence-to-sequence fashion, providing a good trade-off between the prediction performance and required computation resources. Sequences of known wind loads plus time history response of the structure are aggregated into a 3D tensor input before going through a deep learning model, which includes a long short-term memory layer and a time distributed layer. The output of the model is a sequence of structures' future responses, which will subsequently be used as input for computing structure' next response. The credibility of the proposed approach is demonstrated via an example of a two-dimensional three-bay nine-story reinforced concrete frame structure.

Keywords: dynamics of structures; reinforced concrete; frame; vibration; deep learning; surrogate.

https://doi.org/10.31814/stce.huce(nuce)2022-16(2)-09

1. Introduction

Wind load is the most common horizontal load encountered by building structures in practice. Therefore, it is of great importance to reasonably predict structures' dynamic responses and ensure that these responses do not exceed some safety thresholds when the structures are subjected to time-varying wind loads. In the authors' opinion, the wind tunnel is still considered as one of the most reliable, robust, yet expensive means to estimate the impact of wind on the structure. Computationally, the conventional approach is to use the finite element model to directly calculate structure' dynamic behavior. However, it is well-known that analyzing structures' dynamic response is a challenging and time-consuming problem because it requires iteratively solving the governing equation of motion with a significantly small time step to reach convergent results. To decrease the computation time, some authors [1] adopted the modal analysis to approximate structures' behavior with reasonably small errors.

*Corresponding author. *E-mail address:* thangnt2@huce.edu.vn (Thang, N. T.)

Another promising alternative to predict structures' dynamic responses is to use data-driven approaches based on historical data to approximate future values. Some popular and practical data-driven models are autoregressive, moving average, the autoregressive moving average, and autoregressive integrated moving average models. Wan and Ni [2] used the probabilistic Gaussian process model for forecasting structural stress of the Canton Tower. Recently, the data-driven methods using deep learning algorithms have increasingly received attention among the engineering community thanks to their accuracy and practicality because they could implicitly capture underlying temporal relationships within time-series data without requiring any additional preprocessing steps, stationary hypotheses, or complex predefined formulas. Such approaches have been successfully applied in a wide range of domains, e.g., structural damage detection [3], forecasting structural reliability [4], predicting the load-carrying capacity of structures [5], and so on. Koeppe et al. [6] developed a recurrent neural network as a surrogate model for evaluating structures' behavior which could provide results with under 5% deviation compared to those of FEM. Bui et al. [7] used long short-term memory (LSTM) model to forecast the deformation of a hydropower dam, which outperforms several competing models.

However, most of these data-driven reviewed works took into account one time-varying input and were only able to provide one time-series output, such as a displacement time history at a preselected critical location. When working with wind loads, it is necessary to take into account time-varying wind loads, and it is also desirable to calculate displacement time histories of all floors. By doing so, other criteria to assess the structure's safety could be derived, such as maximum displacement, inter-story drift, etc. This study proposes a Deep Learning-based surrogate model, which, unlike previous studies, converts scalars related to structural parameters into time-series data of constant value, then concatenated with time-varying wind loads, forming a 3D fusion-feature tensors, before going through a long short-term memory layer. The main contributions of the article are summarized as follows:

- This study developed a multi-input multi-output temporal surrogate model for predicting structure's dynamic behavior with reduced complexity and reasonable accuracy;

- The efficiency and efficacy of the proposed method are demonstrated through a case study of a nine-story reinforced concrete frame structure.

The remainder of this article is organized as follows. Section 2 briefly presents the dynamic analysis of structures under wind loads; Section 3 describes the main components of the proposed temporal surrogate model; Section 4 demonstrates the applicability of the proposed approach via a case study. Finally, Section 5 draws the conclusion and gives some ideas for future work.

2. Temporal surrogate model for predicting structures' responses

2.1. Dynamic analysis of structures under wind loads

Consider an N-story frame structure modelled as an N-degree of freedom structure where the masses are supposed to be concentrated at the story levels as illustrated in Fig. 1.

The governing equation of motion describing the behavior of this structure is written as follows [8]:

$$M\ddot{X}(t) + C\dot{X}(t) + KX(t) = P(t) = H \times F(z,t) \tag{1}$$

where M, C and K are the structure's mass, damping, and stiffness matrices, respectively. $X(t)$ is the vector of floors' horizontal displacement, and $F(z,t)$ is a vector of time-varying wind load calculated at each story, H is a matrix describing the application of wind loads on corresponding degrees of freedom. For the damping of the structure, the Rayleigh's proportional damping model is used, in

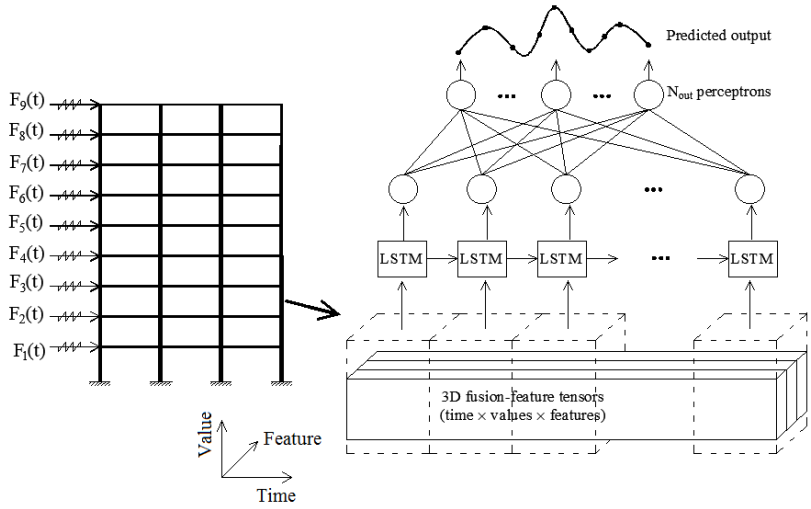

Figure 1. Architecture of the proposed temporal surrogate model for forecasting structures' responses

which the diagonal damping matrix is obtained by linearly combining the mass and stiffness matrices. In this study, the wind load is calculated based on the wind speed according to the procedure described in Eurocode Standard [9]. The wind speed function $U(z,t)$ can be modeled as the sum of the two parts: a mean speed $\bar{U}(z)$ depending on only the height altitude and the time-varying fluctuating part $u(z,t)$ as the following equation:

$$U(z,t) = \bar{U}(z) + u(z,t) \tag{2}$$

The mean speed $\bar{U}(z)$ is calculated by:

$$\bar{U}(z) = c_r(z) \cdot c_0(z) \cdot v_b \tag{3}$$

where v_b is the basic wind velocity depending on the structure location, $v_b = 39$ m/s for structures located in Hanoi [10]; $c_0(z)$ is the orography factor, equal to 1.0 for the ground incline less than 3%; $cr(z)$ is a roughness factor varying with the altitude given by:

$$c_r(z) = k_r \ln\left(\frac{Z}{Z_0}\right) \tag{4}$$

k_r is a terrain factor defined by:

$$k_r = 0.19 \times \left(\frac{Z_0}{Z_{0,II}}\right)^{0.07} \tag{5}$$

where z_0 is the roughness length of terrain category defined in Table 4.1 of Eurocode, $z_{0,II}$ is the roughness length of terrain category II, which is set to 0.05 m. For structures usually encountered in cities, where more than 15% of the surface is covered with buildings with an average height > 15 m, the roughness length z_0 is 1.0 m.

On the other part, the fluctuating part $u(z,t)$ of the wind speed at different stories is modeled by independent and identically distributed Gaussian noise with zero mean and peak value equal to 10% of the peak of the mean wind speed [11], i.e. a standard deviation equal to 3.33% of the latter.

Once the wind speed is obtained, the time-varying wind load F is calculated as follows:

$$F(z,t) = 0.5 \times A \times \rho \times C_p \times U^2(z,t) \tag{6}$$

where ρ is the air density whose default value is 1.25 kg/m^3; C_p is the average wind pressure coefficient which is valued by 1.2 [12] and A is the impact area of the wind load.

Next, Eq. (1) with wind excitation from Eq. (6) can be approximately solved in the time domain by using the mode superposition method which transforms the differential equations of motion into a system of uncoupled differential equations in the modal coordinate. The system of equations is then numerically solved by using a time integration such as Newmark along with the Newton-Raphson algorithm. Such a standard approach has been implemented in various structural analysis softwares such as OpenSees, Abaqus, Sap2000, etc. However, they all require a significantly small time step to ensure convergent results. In contrast, if a large time step is used, the well-known numerical instability problem would likely occur, i.e., the numerical errors would exponentially increase, leading to physically unacceptable results. When a small time step is used, the number of time steps, i.e., number of intermediate calculations will grow significantly, leading to a very high computational cost or even becoming infeasible.

2.2. Architecture of the deep learning-based temporal surrogate model

In this section, the deep learning (DL)-based forecasting framework is described in detail. Though the DL architecture is one of the most critical parts of a DL-based framework, there are other important components which significantly affect the final performance such as the data preparation, the forecasting strategy, and the training process.

Suppose that we have a time series of interest $\varphi(t) \in R$ denotes the value of the time series at time t. Assume that the historical values of $\varphi(t)$ are known up to time instant t_0, one wants to predict its next τ time steps denoted by $y\,(t_0 + 1 : t_0 + \tau)$ based on its historical values, time-independent input Ω and time-dependent excitations $Y(t)$. In this study, $\varphi(t)$ are floors' displacement time history, $Y(t)$ are time-varying wind excitation, and Ω is a vector of key structural parameters. The input data are then preprocessed and aggregated into a three-dimensional tensor χ_t (time × values × feature).

After that, input data go through a variant of the recurrent neural network (RNN) layer, namely long-short term memory (LSTM), which are described as follows. Generally, a RNN consists of jointly connected repeating cells where previous cells take early data as input, while subsequent cells take later data and output of previous ones as input. More specifically, consider time instant t, the input for the RNN cell corresponding to this instant includes the value of time-series data χ_t and output of the previous RNN cell at time instant $t - 1$. For the first cell, it takes only the value of time-series data χ_0 as input. That is why it is naturally capable of capturing temporal relationships in data.

However, when working with long time-series data, vanishing or exploding gradient problems could occur because the gradient of the loss function with respect to the network's weights propagate backward through multiple RNN cells. To overcome this problem, the LSTM [13, 14] was developed based on the gate mechanisms to control the information flow (Fig. 2).

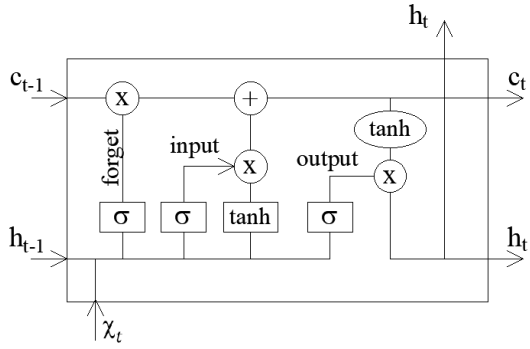

Figure 2. Typical LSTM cell with gate mechanism

It is shown in Fig. 2 that each cell has three gates, namely forget gate, input gate, and output gate for controlling information flow. After each LSTM cell, two components are delivered: the cell output h_t and cell state c_t ranging between $[0, 1]$. A zero value of c_t means that the cell information will be forgotten while one value indicates that the cell information will be retained in long term.

In Fig. 2, the input data for the LSTM cell corresponding to time step t are χ_t, output h_{t-1}, and state c_{t-1} of the LSTM cell at time step $t-1$. Introducing a linear transformation L of the combination of χ_t and h_{t-1}, as follows:

$$L\left(h_{t-1},\chi_t\right) = W \times \left[h_{t-1},\chi_t\right] + b \tag{7}$$

where W and b are the weight matrix and bias vector of the network, respectively.

The three gates inside each cell of LSTM are computed by:

$$\begin{aligned} f_f &= \sigma\left(L_f\left(h_{t-1},\chi_t\right)\right) \\ f_i &= \sigma\left(L_i\left(h_{t-1},\chi_t\right)\right) \\ f_o &= \sigma\left(L_o\left(h_{t-1},\chi_t\right)\right) \end{aligned} \tag{8}$$

in which the subscripts f, i, o indicate the forget gate, input gate and output gate, respectively and σ is the sigmoid activation function.

A candidate information s_t is calculated by applying the tanh activation function on a linear transformation of the combination (h_{t-1},χ_t):

$$s_t = \tanh\left(L_s\left(h_{t-1},\chi_t\right)\right) \tag{9}$$

Note that L_f, L_o, L_i, and L_s have the same mathematical form presented in Eq. (7), but their weight matrices denoted by W_f, W_o, W_i and W_s do not share the same weights. Next, the flow of information from the previous step h_{t-1} is updated with the new cell state C_t using the input gate and forget gate, as below:

$$C_t = \left(f_f \otimes C_{t-1}\right) \oplus \left(f_i \otimes s_t\right) \tag{10}$$

where $\otimes$ and $\oplus$ are element-wise product and element-wise addition, respectively. Next, the output of the cell at a time step t is calculated based on the updated information and the output gate:

$$h_t = f_o \otimes \tanh\left(C_t\right) \tag{11}$$

Next, the output of all LSTM cells will be concatenated and go through a fully-connected layer to predict future values as shown in Fig. 1. After that, these predicted outputs are used as input to compute subsequent values, until the final instant of the calculation time is reached.

The network's weights are determined by minimizing the discrepancy between the predicted values and the actual ones. The number of trainable parameters of each layer will be specified later in the case-study section. In this study, the discrepancy is quantitatively estimated by the root mean square error, also known as L_2 loss function. The adopted optimization algorithm is the first-order algorithm Stochastic Gradient Descent (SDG) [15]. As deep learning application usually works with a considerably large volume of data; thus, using high-order optimization algorithms may exponentially increase the computation time, even making the problem intractable. As its name suggests, the SDG iteratively adjusts the model weights in the opposite direction of the gradient of the loss function (Gradient Descent), i.e., making the loss function smaller. Furthermore, at each iteration of calculation, SDG only employs a random sub-dataset whose size is significantly smaller than that of the original data. By doing so, one can carry out the training process on a conventional computer without requiring expensive GPU devices with high memory capacity. An important configurable hyper-parameter of the SDG is the learning rate which is a small positive value signifying how much the networks' weights are updated each calculation iteration. To avoid a premature convergence problem due to large values of the learning rate, but not excessively extend the training time caused by too small values, the adaptive

learning rate schedule along with early stopping is used. One starts with a relatively large learning rate of 0.01 as it helps accelerate the early stage of the training process where the loss function quickly decreases from a large value to small ones, as shown in the first part of the learning curve in Fig. 4. The learning rate will be divided by half if no improvement is observed, and in the case of multiple no-improvement iterations (up to 10 consecutive steps), a very small learning rate around 1E-5 (which is also a lower bound in this study) would be used. In contrast, if a small learning rate is selected right at the beginning, it will take more iteration steps for the loss function descending the learning curve, i.e., more unnecessary computation time is required.

3. Case study: a 2D nine-story frame structure

In this section, the proposed temporal surrogate model is applied to a 2D nine-story three-bay reinforced concrete (RC) frame structure whose typical story height is 4 m, except for the first floor with a height of 5 m. The bay widths are 6 m. The cross-section size of the beam is 400 × 600 mm, while the section size of the columns is diminished with increasing stories, as illustrated in Fig. 3. The structure is subjected to time-varying wind excitations acting on every floor, these excitations are modeled by using the procedure described in the previous section.

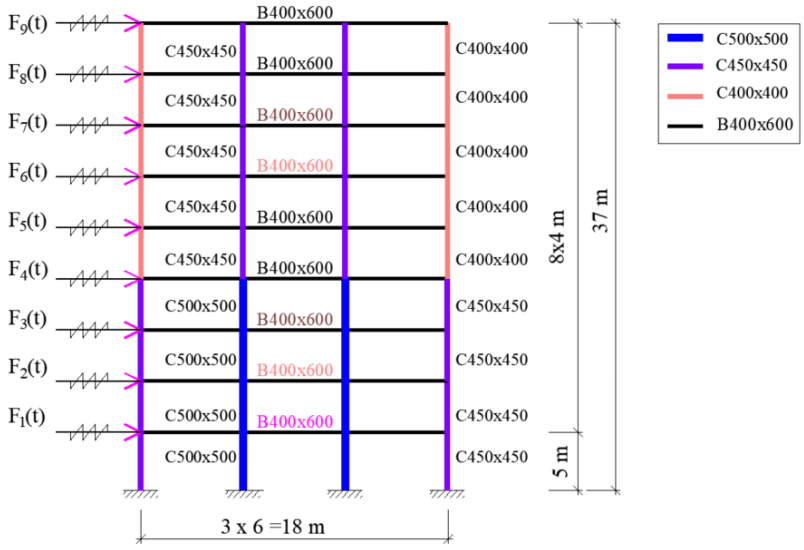

Figure 3. 9-story 3-bay RC frame structure subjected to time-varying wind excitations

In order to prepare adequate training data for the surrogate model, one utilizes the FEM to carry out a series of simulations of the structure. Herein, the FEM program OpenSees [16] by the Pacific Earthquake Engineering Research Center is adopted thanks to its credibility, efficiency, and openness. The structural members including beams and columns are modeled by using elastic elements with corresponding cross-section sizes and material properties. Besides, the Rayleigh damping with a ratio of 5% is set to the first two eigenmodes of the structure. To model the section of RC elements, the fiber approach is utilized because it can take into account both flexure and shear behaviors at the same time. More specifically, the section is divided into a grid of fibers, each fiber has its location and area. In terms of the element type, the concentrated inelasticity elements are adopted, which assumes that if the deformation goes beyond the elastic threshold, the plasticity will occur but concentrate

on a limited length at the span ends while the rest of the element remains elastic. For numerically solving the equation of motion, the Newmark time integration scheme with parameters $\gamma = 0.5$ and $\beta = 0.25$, a time step of $dt = 0.01$ s, a simulation duration of 30 s, and Newton-Raphson algorithm are utilized as mentioned in Section 2. In order to enrich the database, these above structural properties are considered as random values normally distributed with means and coefficients of variance enumerated in Table 1. The random variables in the table include Young moduli of steel and concrete E_s, E_c, the yield strength of steel f_s, the compressive strength of concrete f_c, damping ratio ζ, beam height and width H_b, B_b and column cross section sizes H_{c1}, H_{c2}, and H_{c3}.

Table 1. Structural parameters and their statistical characteristics

	E_s	f_s	E_c	f_c	ζ	H_b	W_b	H_{c1}	H_{c2}	H_{c3}
Mean	210E9	400E6	30E9	25E6	0.05	0.6	0.4	0.4	0.45	0.5
Cov (*)	1/20	1/20	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10
Dist.	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal

(*) Cov: is the coefficient of variation which is the ratio between the standard deviation and the mean value.

In total, 10000 simulations with different wind excitations and structural parameters drawn from Table 1, were performed. Outputs of each simulation are 30 s-duration displacement time histories of each floor.

Having the FEM database in place, the sliding window technique is adopted for data preparation. For example, 500 values from the 1st to the 500th of the data and its 100 subsequent values form an input/output pair. Next, by moving 5 steps forwards, i.e., the 5th to 505th values, and its 100 time-step sequences ahead constitute another pair of input/output, and so forth. It is noted that these lengths are essential parameters, which could affect the efficiency of the framework. Using a large value of input length will reduce the total number of calculation steps, but require more memory for storage. In contrast, short input lengths will increase the number of intermediate steps and likely aggravate cumulative errors. For this study, with a computation device equipped 12 GPU Gb RAM, the length of historical data is set to 500, while that of the future output is 100.

Fig. 4 displays the training curve of the proposed temporal surrogate model on the FEM-generated training database, showing the decreasing trend of loss function values with an increasing number of epochs. Around epoch 250, there is no clear improvement, and more fluctuation is observed. As mentioned before, early stopping was applied, and the configuration corresponds to the lowest loss function value is retrieved as the final surrogate model.

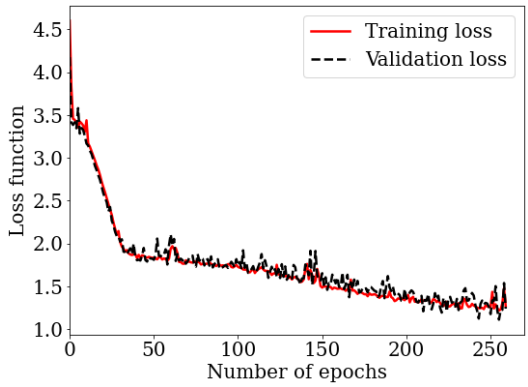

Figure 4. Learning curves over the training process

Next, the trained model is used to infer the structure's responses to random wind loads. Note that, the input of the surrogate model are wind load time histories at each story and structural parameters. Initially, the structure is supposed to be at rest. The inference will be proceeded in an iterative fashion, meaning the predicted results of previous steps will be used as input for the next steps. It is also noteworthy to recall that the outputs of the

surrogate model are displacements at all stories. Fig. 5 demonstrates an example of results obtained by the temporal surrogate model, the ground-truth results obtained by FEM are also plotted for comparison purpose. There are nine subfigures corresponding to nine stories. For clarity, the scales of the y-axis of subfigures are adjusted based on data amplitude, so they are not the same. The vibration amplitude of the higher story is greater than those of lower stories. Apparently, a very good agreement is observed as the lines of the surrogate model closely overlap those of FEM, which means the temporal surrogate model is able to accurately predict the structure's vibration.

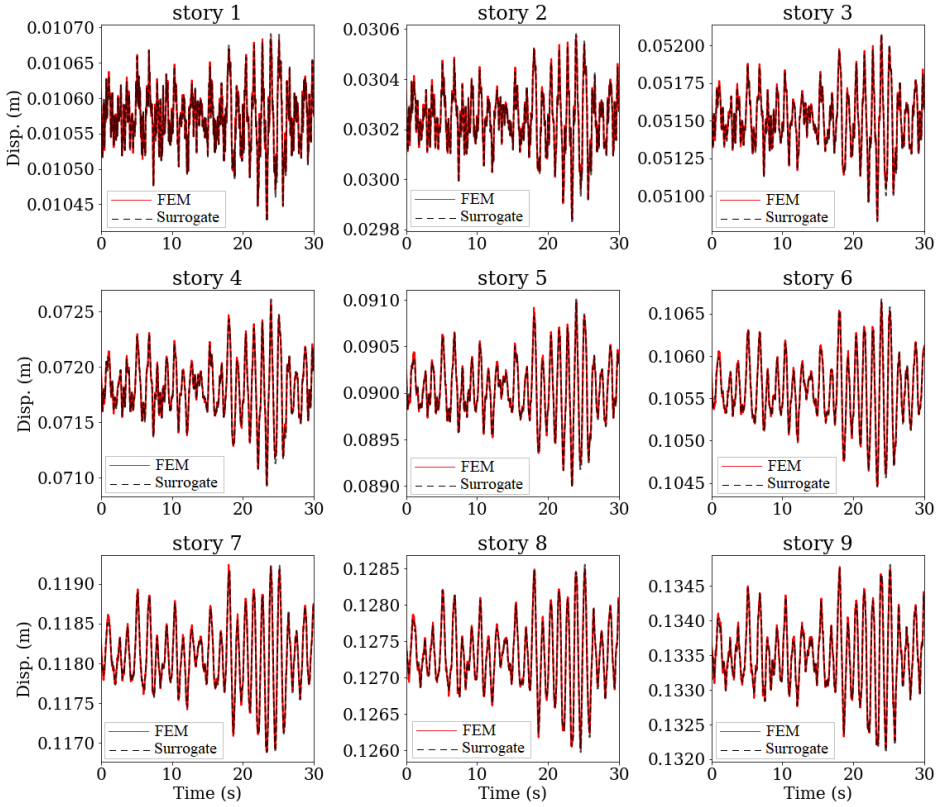

Figure 5. Representative examples of prediction results obtained by the temporal surrogate model and FEM

Fig. 6 presents the stories' maximum displacement provided by the surrogate model on the test data, completely unseen during the training set. Specifically, the maximum displacement is determined by taking the maximum absolute value of displacement time histories. On the figure, the Y-axis is the predicted value, the X-axis is reference values obtained by FEM, the 45 degree-red line is the ideal line where the predicted values perfectly match with reference ones. It can be seen that, scattered points representing prediction results are located near the ideal line, just proving the credibility and reproducibility of the surrogate model.

Furthermore, as the surrogate model can provide an entire displacement time history rather than a few scalar values as some competing methods; thus, one can derive other important criteria such as maximum inter-story drift ratio as shown in Fig. 7. The figure displays the histogram of maximum drift ratios for 1000 samples of the testing data obtained by the surrogate model and those by FEM, showing a reasonable agreement.

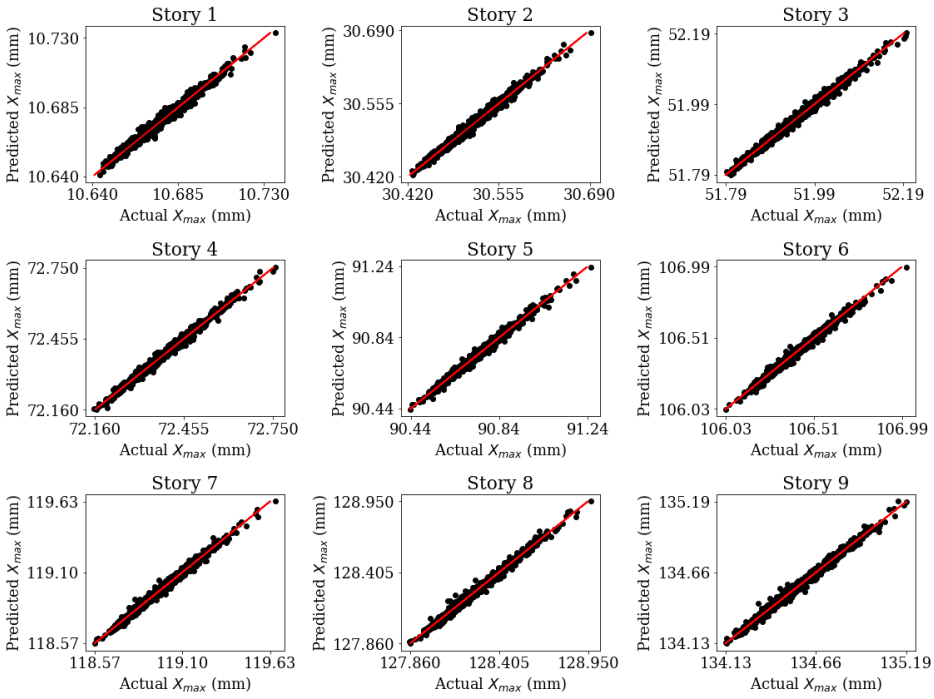

Figure 6. Comparison results of the maximum story displacement between the surrogate model and FEM

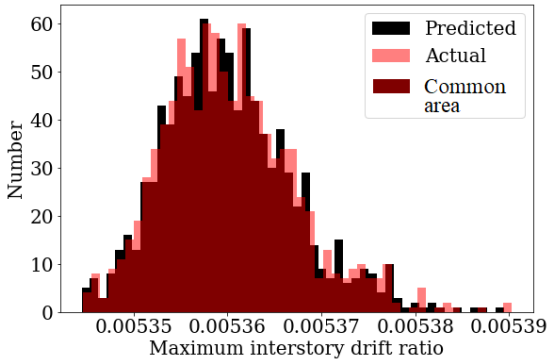

Figure 7. Histogram of maximum inter-story drift ratio on the test data

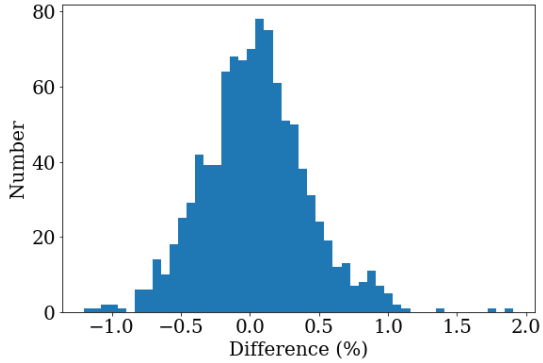

Figure 8. Difference of maximum inter-story drift ratio obtained by FEM and the surrogate model

Moreover, Fig. 8 presents the sample-by-sample difference of maximum inter-story drift ratio obtained by FEM and the surrogate model, showing that almost difference lies within the range $[-1.0\%, 1.0\%]$ which reaffirm the accuracy of the proposed model.

On the other hand, the time complexity of the surrogate model for all steps including data generation, training process and inference time are enumerated in Table 2.

It can be seen that a large amount of time spends on data generation using FEM, it takes about 2900 minutes for generating 10000 samples, i.e., 17.4 s for one calculation sample. Meanwhile, at inference time, it takes only 0.21 s for one calculation sample ($3.6 \times 60/1000$ s), i.e., the surrogate model could reduce the time complexity by approximately three orders of magnitude. Therefore, the proposed surrogate model is an alternative but complementary and very efficient approach to the FEM

for applications requiring a large number of calculations such as parametric, reliability or optimization studies.

Table 2. Time complexity of the proposed temporal surrogate model

Steps	Size	Computational time (minutes)
Data generation	10000 samples	~ 2900 (using FEM)
Training process	8000 samples, ~ 250 epochs	32.1
Inference	1000 samples	3.6

4. Conclusions

This article develops a temporal deep learning-based method for forecasting dynamic responses of structures prone to wind excitation with reduced time complexity while still maintaining high accuracy. The theoretical background of wind loads and deep learning architecture and implementation details, including key hyperparameters, and training curve of the proposed method, were described throughout the article.

Moreover, its viability was quantitatively demonstrated through a case study of a 9-story RC frame. The results showed that the maximum top floor displacement obtained by the proposed approach was highly similar to those calculated by the FEM. Meanwhile, the time complexity was reduced up to three orders of magnitude at inference time. Such an efficient and effective method is remarkably practical for various real scenarios where a large number of calculations are needed to carry out, such as reliability, sensitivity, or parametric analysis.

In the future, it is desirable to extend the proposed approach for other structures with different geometrical configurations. This can be done by combining the Deep Learning-based model with the graph theory, forming a hybrid spatio-temporal model. Another interesting direction is to extend the model for more complex structural problems, for example soil-structure interaction, structures subjected to both earthquake and wind loads, etc., where highly non-linearity may have a critical impact on the model performance.

References

[1] Dang, V. H., Perret-Liaudet, J., Scheibert, J., Bot, A. L. (2013). Direct numerical simulation of the dynamics of sliding rough surfaces. *Computational Mechanics*, 52(5):1169–1183.

[2] Wan, H.-P., Ni, Y.-Q. (2018). Bayesian Modeling Approach for Forecast of Structural Stress Response Using Structural Health Monitoring Data. *Journal of Structural Engineering*, 144(9):04018130.

[3] Hung, D. V., Hung, H. M., Anh, P. H., Thang, N. T. (2020). Structural damage detection using hybrid deep learning algorithm. *Journal of Science and Technology in Civil Engineering (STCE) - NUCE*, 14 (2):53–64.

[4] Dang, H. V., Trestian, R., Bui-Tien, T., Nguyen, H. X. (2021). Probabilistic method for time-varying reliability analysis of structure via variational bayesian neural network. *Structures*, 34:3703–3715.

[5] Hung, T. V., Viet, V. Q., Thuat, D. V. (2019). A deep learning-based procedure for estimation of ultimate load carrying of steel trusses using advanced analysis. *Journal of Science and Technology in Civil Engineering (STCE) - NUCE*, 13(3):113–123.

[6] Koeppe, A., Bamer, F., Padilla, C. A. H., Markert, B. (2017). Neural network representation of a phase-field model for brittle fracture. *PAMM*, 17(1):253–254.

[7] Bui, K.-T. T., Torres, J. F., Gutiérrez-Avilés, D., Nhu, V.-H., Bui, D. T., Martínez-Álvarez, F. (2022). Deformation forecasting of a hydropower dam by hybridizing a long short-term memory deep learning network with the coronavirus optimization algorithm. *Computer-Aided Civil and Infrastructure Engineering*.
[8] Chopra, A. K. (2007). *Dynamics of structures*. Pearson Education India.
[9] British Standards Institution (2005). *Eurocode 1: Actions on structures*. London: BSI.
[10] Vietnam Standard TCVN 2737:2020. *Loads and Effects - Design standard*.
[11] Bai, L., Zhang, Y. (2013). Nonlinear dynamic behavior of steel framed roof structure with self-centering members under extreme transient wind load. *Engineering Structures*, 49:819–830.
[12] Liu, Z., Liu, Z. (2018). Random function representation of stationary stochastic vector processes for probability density evolution analysis of wind-induced structures. *Mechanical Systems and Signal Processing*, 106:511–525.
[13] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
[14] Houdt, G. V., Mosquera, C., Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8):5929–5955.
[15] Lydia, A., Francis, S. (2019). Adagrad—an optimizer for stochastic gradient descent. *International Journal Of Information And Computing Science*, 6(5):566–568.
[16] McKenna, F. (2011). OpenSees: A Framework for Earthquake Engineering Simulation. *Computing in Science & Engineering*, 13(4):58–66.