

MIT Open Access Articles

Deep long short-term memory networks for nonlinear structural seismic response prediction

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Zhang, Ruiyang et al. "Deep long short-term memory networks for nonlinear structural seismic response prediction." Computers and Structures, 220, (August 2019): 55-68 © 2019 The Author(s)

As Published: 10.1016/J.COMPSTRUC.2019.05.006

Publisher: Elsevier BV

Persistent URL: <https://hdl.handle.net/1721.1/126761>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-NonCommercial-NoDerivs License



Deep long short-term memory networks for nonlinear structural seismic response prediction

Ruiyang Zhang^a, Zhao Chen^a, Su Chen^b, Jingwei Zheng^c, Oral Büyüköztürk^d, Hao Sun^{a,d,*}

^a*Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115, USA*

^b*Institute of Geophysics, China Earthquake Administration, Beijing, 100124, China*

^c*Electric Power Planning and Engineering Institute, Beijing, 100120, China*

^d*Department of Civil and Environmental Engineering, MIT, Cambridge, MA 02139, USA*

Abstract

This paper presents a comprehensive study on developing advanced deep learning approaches for nonlinear structural response modeling and prediction. Two schemes of the long short-term memory (LSTM) network are proposed for data-driven structural seismic response modeling. The proposed deep learning model, trained on available datasets, is capable of accurately predicting both elastic and inelastic response of building structures in a data-driven fashion as opposed to the classical physics-based nonlinear time history analysis using numerical methods. In addition, an unsupervised learning algorithm based on a proposed dynamic K-means clustering approach is established to cluster the seismic inputs in order to (1) generate the least but the most informative datasets for training the LSTM and (2) improve the prediction accuracy and robustness of the model trained with limited data. The performance of the proposed approach is successfully demonstrated through three proof-of-concept studies that include a nonlinear hysteretic system, a real-world building with field sensing data, and a steel moment resisting frame. The results show that the proposed LSTM network is a promising, reliable and computationally efficient approach for nonlinear structural response prediction, and offers significant potential in seismic fragility analysis of buildings for reliability assessment.

Keywords: deep learning, long short-term memory, LSTM, nonlinear dynamic analysis, seismic response prediction, time series clustering

1. Introduction

Buildings are vulnerable to natural hazards such as earthquakes, tsunamis and hurricanes. Classical methods for analyzing structural dynamic response subjected to these events typically utilize physics-based models, with linear/nonlinear time history analysis capabilities, based on the principles of mechanics. Numerous studies have been conducted on those methods in last several decades leading to many successful computational implementations in the civil engineering field, among which the finite element method (FEM)

*Corresponding author. Tel: +1 617-373-3888

Email address: h.sun@northeastern.edu (Hao Sun)

is one of the most popular numerical methods for structural dynamic analysis. However, finite element models for dynamic analysis of nonlinear structures usually require small time computational steps and fine element mesh discretization to ensure the solution stability and accuracy. Therefore, it is computationally expensive of using FEM for nonlinear time history analysis, especially when numerous simulations are required to account for stochastic uncertainties of external loads (e.g., Monte Carlo simulations [1–3] or incremental dynamic analysis [4–6] of structural systems for fragility analysis). In addition, model assumptions and parameter uncertainties sometimes lead to significant errors when utilizing FEM to model existing structure systems.

There have tremendous attempts on developing more efficient alternative approaches for surrogate modeling of structural responses based on available input/output data, such as the system identification (SI)-based methods. The basic concept of SI-based methods performs linear/nonlinear mapping of external excitation(s) to the corresponding structural response using state-space or black-box models [7–11]. A comprehensive review of the works on SI and its applications in structural response modeling was given in [12–14]. Alternatively, finite element (FE) model updating has been extensively studied and used to model/predict structural responses [15–18]. For example, Skolnik *et al.* [19] predicted the seismic response of a 15-story steel-frame building using an updated FE model. The model updating process generally requires excessive computational efforts on calibrating the model especially when the model is of high fidelity with a large number of parameters. Though low-fidelity models are more computationally cost-effective, it scarifies accuracy. It is worth mentioning that autoregressive integrated moving average (ARIMA) is one of the most popular linear models for time series modeling [20, 21], which was originated from the autoregressive models (AR) [22], the moving average models (MA) [23] and the auto-regressive moving average models (ARMA) [24]. In general, limitations of the aforementioned methods arise to due to hypotheses such as stationary dynamics and linearity, which makes applying these approaches to model nonlinear responses intractable.

In recent years, artificial neural networks (ANN) has gained considerable attention due to its superior capability in nonlinear functional modeling [25–29]. Researchers have successfully implemented ANN for modeling structural responses under static and dynamic loading during the past decade. For example, the multi-layer perceptron (MLP) network, as a classical ANN, has been very popular with applications to structural system/parameter identification and damage diagnosis [30–33], as well as dynamic response prediction of variety type of structural systems such as slender marine structures [34], bridges [35], buildings [36], and wind turbines [37]. Nevertheless, due to its simple architecture, MLP has distinct limitations in modeling time series of dynamical systems which have highly, complex non-linearity. Recent advances in deep learning such as the the convolutional neural network (CNN) [38] and the recurrent neural network (RNN) [39, 40] offers opportunities to model nonlinear structural responses, given their outstanding capability in deep feature extraction. It is noted that though CNN is known for its capability in classification of data with grid-like topology [41, 42] (e.g., 1D sequences, 2D images), it is also applicable to regression problems despite that very limited studies have been reported in literature (e.g., structural response estimation using 1D CNN [43–45]). Nevertheless, the prediction accuracy of the 1D CNN

dramatically drops when predicting structural response with large plastic deformations. Instead, RNN is specially designed to learn sequential, time-varying, linear/nonlinear patterns for regression problems [39, 40], showing a great potential on structural response modeling. Nevertheless, it is intractable to train RNNs using the gradient-based back propagation for long-range data dependencies due to the gradients vanishing and exploding [46]. This issue makes RNNs incapable of modeling long-range structural responses, especially when complex nonlinearity is present in the responses.

We herein address those aforementioned limitations in nonlinear structural response modeling and prediction through developing a long short-term memory (LSTM) network using limited datasets. To our best knowledge, this paper presents probably the first attempt to employ deep LSTM, with very limited data, for response modeling of highly nonlinear complex dynamic systems (e.g., plastic deformation). Two LSTM schemes are proposed in this paper for data-driven seismic response modeling. In addition, an unsupervised learning algorithm based on K-means clustering is presented for clustering the seismic inputs in order to (1) generate the least but the most informative datasets for training the LSTM and (2) improve the prediction accuracy and robustness of the model trained with limited data. The performance of the proposed LSTM framework is demonstrated through three proof-of-concept studies with both simulated and field-sensing datasets. The contribution and novelty of this paper include (1) the stacked deep LSTM architecture, and (2) the unsupervised learning algorithm for data clustering that enhances the trainability of the proposed deep neural networks.

This paper is organized as follows. Section 2 introduces the proposed LSTM architecture for seismic response modeling and prediction. Section 3 presents the K-means-based unsupervised learning algorithm for data sequence clustering. In Section 4, the proposed methodology is validated both numerically and experimentally through three proof-of-concept studies that include a nonlinear hysteretic system, a real-world building with field sensing data, and a steel moment resisting frame. Finally, Section 5 discusses the limitations of the proposed approach and summarizes conclusions of this paper.

2. Long Short-Term Memory (LSTM) Network

Neural networks have been widely recognized as a powerful machine learning tool for classification and regression problems. Unlike feedforward neural networks, RNNs have feedback loops with recurrent connections between the nodes of the network making them capable of modeling sequences. Noteworthy, long short-term memory (LSTM), as a variation of RNNs, is designed to capture long-range data dependencies for modeling sequential data (e.g., time series), which shows a great potential and promise in modeling nonlinear structural seismic responses.

2.1. The LSTM cell

The standard deep LSTM architecture consists of multiple hidden layers (including the LSTM layers and the fully connected layers) in addition to the input and output layers. The LSTM layer contains a suite of LSTM cells and maps the layer input sequence to the layer

output sequence. Each LSTM cell (see its structure in Figure 1), which is very similar to the neural node in ANN, contains an independent set of weights and biases shared across the entire temporal space within the layer. The LSTM cell consists of four interacting units, including an internal cell, an input gate, a forget gate, and an output gate. The internal cell memorizes the cell state at the previous time step through a self-recurrent connection. The input gate controls the flow of input activation into the internal cell state. The output gate regulates the flow of output activation into the LSTM cell output. The forget gate scales the internal cell state, enabling the LSTM cell to forget or reset the cell's memory adaptively. Let us denote, at the time step t ($t = 1, \dots, n$, where n is the total number of time steps) and within the l th LSTM network layer, the input state to the LSTM cell as $\mathbf{x}_t^{(l)}$, the forget gate as $\mathbf{f}_t^{(l)}$, the input gate as $\mathbf{i}_t^{(l)}$, the output gate as $\mathbf{o}_t^{(l)}$, the cell state memory as $\mathbf{c}_t^{(l)}$, and the hidden state output as $\mathbf{h}_t^{(l)}$. At the previous time step $t - 1$, we denote the cell state memory as $\mathbf{c}_{t-1}^{(l)}$ and the hidden state output as $\mathbf{h}_{t-1}^{(l)}$. The relationship among these defined variables can be described as follows.

$$\mathbf{f}_t^{(l)} = \sigma \left(\mathbf{W}_{xf}^{(l)} \mathbf{x}_t + \mathbf{W}_{hf}^{(l)} \mathbf{h}_{t-1} + \mathbf{b}_f^{(l)} \right) \quad (1)$$

$$\mathbf{i}_t^{(l)} = \sigma \left(\mathbf{W}_{xi}^{(l)} \mathbf{x}_t + \mathbf{W}_{hi}^{(l)} \mathbf{h}_{t-1} + \mathbf{b}_i^{(l)} \right) \quad (2)$$

$$\tilde{\mathbf{c}}_t^{(l)} = \tanh \left(\mathbf{W}_{xc} \mathbf{x}_t + \mathbf{W}_{hc} \mathbf{h}_{t-1} + \mathbf{b}_c^{(l)} \right) \quad (3)$$

$$\mathbf{o}_t^{(l)} = \sigma \left(\mathbf{W}_{xo} \mathbf{x}_t + \mathbf{W}_{ho} \mathbf{h}_{t-1} + \mathbf{b}_o^{(l)} \right) \quad (4)$$

$$\mathbf{c}_t^{(l)} = \mathbf{f}_t^{(l)} \odot \mathbf{c}_{t-1}^{(l)} + \mathbf{i}_t^{(l)} \odot \tilde{\mathbf{c}}_t^{(l)} \quad (5)$$

$$\mathbf{h}_t^{(l)} = \mathbf{o}_t^{(l)} \odot \tanh \left(\mathbf{c}_t^{(l)} \right) \quad (6)$$

where $\mathbf{W}_{\alpha\beta}^{(l)}$ (here $\alpha = \{x, h\}$ and $\beta = \{f, i, c, o\}$) denotes the weight matrices corresponding to different inputs (e.g., $\mathbf{x}_t^{(l)}$ or $\mathbf{h}_t^{(l)}$) within different gates (e.g., input gate, forget gate, tanh layer or output gate as shown in Figure 1), while $\mathbf{b}_\beta^{(l)}$ represents the corresponding bias vectors; l denotes the l th layer of the LSTM network. For example, $\mathbf{W}_{xf}^{(l)}$ and $\mathbf{W}_{hf}^{(l)}$ are the weight matrices corresponding to input vectors \mathbf{x}_t or \mathbf{h}_t , respectively, within the forget gate. Here, $\tilde{\mathbf{c}}_t^{(l)}$ denotes a vector of intermediate candidate values created by a tanh layer shown in Figure 1; σ is the logistic sigmoid function; \tanh is the hyperbolic tangent function; \odot denotes the Hadamard product (element-wise product).

It is shown in Figure 1 that the input to the LSTM cell consists of $\{\mathbf{x}_t^{(l)}, \mathbf{c}_{t-1}^{(l)}, \mathbf{h}_{t-1}^{(l)}\}$ while the output includes $\{\mathbf{c}_t^{(l)}, \mathbf{h}_t^{(l)}\}$. This means that the LSTM cell within the l th layer takes the input data at the current time step $\mathbf{x}_t^{(l)}$ (note: the input to the l th layer is essentially

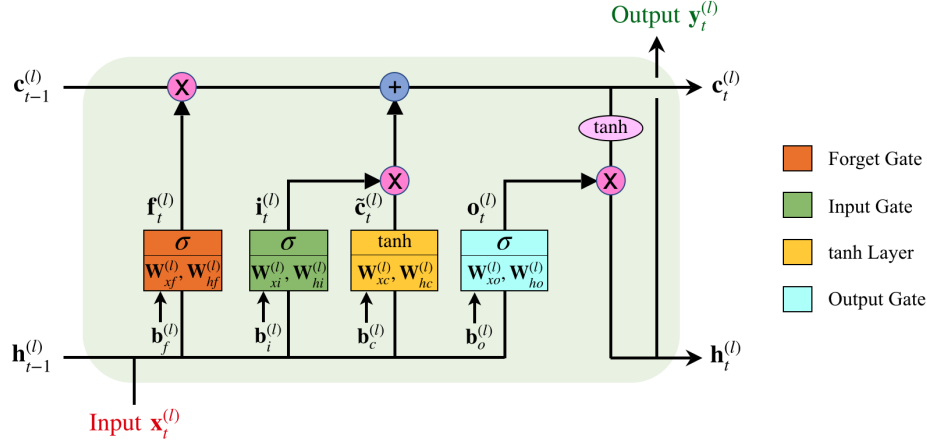


Figure 1: The diagram of an LSTM cell of the l th layer at time t . Note that the output $\mathbf{y}_t^{(l)}$ is equal to the hidden state output $\mathbf{h}_t^{(l)}$.

the output from the $(l-1)$ th layer at time t when $l > 1$, namely, $\mathbf{x}_t^{(l)} = \mathbf{y}_t^{(l-1)}$) as well as the hidden state output $\mathbf{h}_{t-1}^{(l)}$ and the cell state output $\mathbf{c}_{t-1}^{(l)}$ at the previous time step as the input, and outputs the hidden state output $\mathbf{h}_t^{(l)}$ as well as the cell state output $\mathbf{c}_t^{(l)}$ at the current time step. Here, we have several notes on how the memory is operated through the gates for both long and short terms. First, the cell inputs $\mathbf{x}_t^{(l)}$ and $\mathbf{h}_{t-1}^{(l)}$ are passed through the forget gate (sigmoid function) generating a vector with values ranging from 0 to 1 to remove or “forget” information from the cell state ($\mathbf{c}_{t-1}^{(l)}$). Note that a “0” represents completely forgotten and a “1” denotes completely memorized. The next step is to determine what information needs to be added to the cell state ($\mathbf{c}_{t-1}^{(l)}$) through both input gate (sigmoid function) and a hyperbolic tangent function. The cell state will then be updated by adding two resulting components together as given by Eq. (5). The new cell state ($\mathbf{c}_t^{(l)}$) is then scaled within $[-1, 1]$ by a hyperbolic tangent function and controlled by a sigmoid function through the output gate in order to output the hidden state $\mathbf{h}_t^{(l)}$ of the LSTM cell.

2.2. The LSTM network architecture

A typical deep LSTM network for sequence data modeling is composed of multiple LSTM layers and fully-connected (FC) layers as illustrated in Figure 2. The FC layers are used to connect the LSTM layers to the target output layer and thus construct the desired number of output features. Exactly as its name implies, FC layers have full connections to all activation nodes in the previous layer. A FC layer (e.g., the l' th FC layer) multiplies the input by a weight matrix ($\mathbf{W}_{FC}^{(l')}$) and adds a bias vector ($\mathbf{b}_{FC}^{(l')}$). So far, all the unknown weight parameters θ of a multi-layer LSTM-FC network that need to be determined through training can be summarized as follows

$$\theta = \bigcup \left\{ \mathbf{W}_{xf}^{(l)}, \mathbf{W}_{hf}^{(l)}, \mathbf{W}_{xi}^{(l)}, \mathbf{W}_{hi}^{(l)}, \mathbf{W}_{xc}^{(l)}, \mathbf{W}_{hc}^{(l)}, \mathbf{W}_{xo}^{(l)}, \mathbf{W}_{ho}^{(l)}, \mathbf{W}_{FC}^{(l')} \right\}_{l=1,2,\dots,L \text{ and } l'=1,2,\dots,L'} \quad (7)$$

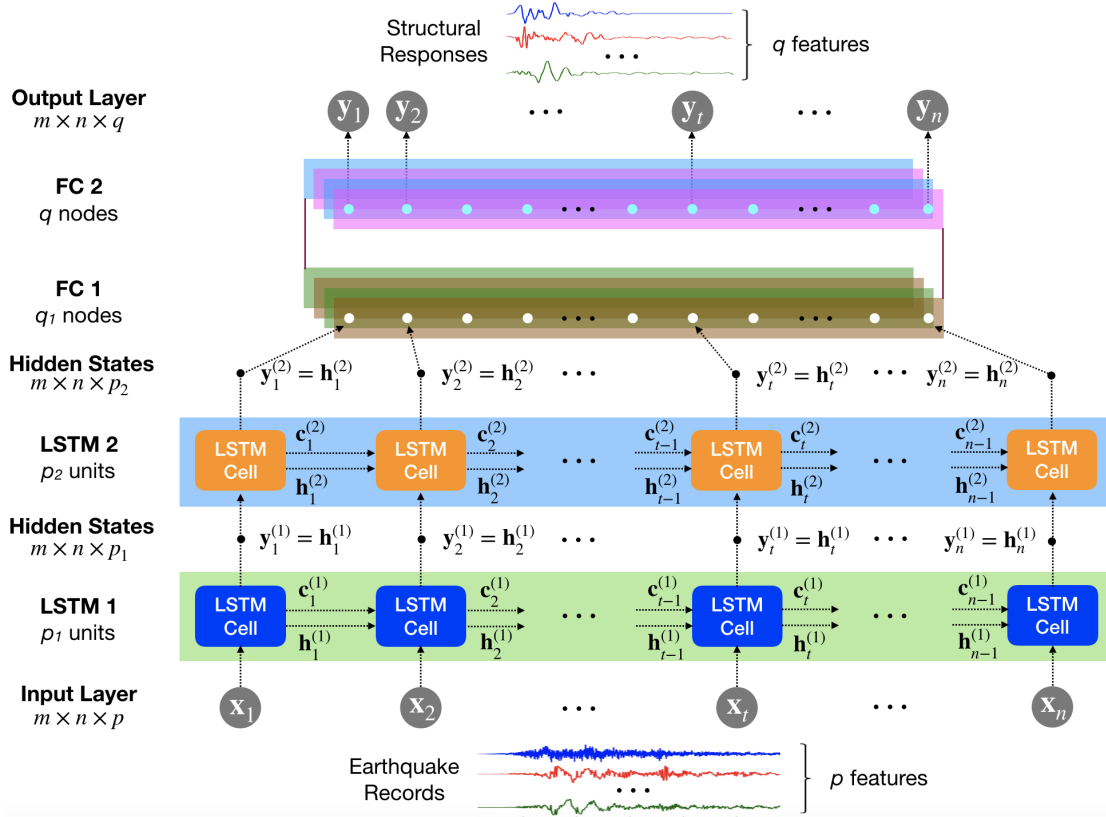


Figure 2: A typical multi-layer LSTM architecture for structural seismic response modeling. This LSTM architecture includes the input layer, two LSTM layers, two fully connected (FC) layers, and the output layer (note that the activation functions are not shown in this figure). Here, p_1 and p_2 denote the number of units (i.e., nodes) for the two LSTM layers, while q_1 and q_2 denote the number of nodes for the two FC layers.

where L denotes the total number of LSTM layers and L' denotes the total number of FC layers. In addition, dropout layers can be added after each layer to reduce overfitting through preventing complex co-adaptations on training data [47, 48] which has remained as a common issue in deep learning. The key idea is to randomly disconnect the connections and drop nodes from the connected layer with a certain dropout rate during training. Dropout layers can also improve the training speed since less parameters will be trained after dropout. Typically, the dropout operation is applied before the FC layer, where exists more learnable parameters and is more likely to cause overfitting. In this study, dropout layers with a rate of 0.2 are applied to the LSTM layers only. In addition, it is noteworthy that the nonlinear activation function Relu is applied within the first L' layers, while the last FC later (e.g., $l' = L'$) takes the linear activation function.

It is noted that, to implement the LSTM architecture with multiple datasets, both the input and output sequences must be formatted as three-dimensional arrays, where the entries are the samples (e.g., independent datasets) in the first dimension (denoted as m), the time

steps in the second dimension (denoted as n), and the input or output features/channels (denoted as p and q , respectively) in the third dimension. The first dimension will be one when using the trained network to evaluate a single set of input sequence. We present two types of input/output format for the LSTM network, including (1) LSTM with full sequence to sequence mapping (LSTM-f), and (2) LSTM with stacked sequence to sequence (LSTM-s) mapping, which are discussed in detail in the context of structural seismic response modeling in Sections 2.3 and 2.4.

2.3. Full sequence to sequence LSTM network (LSTM-f)

In the context of seismic response modeling, let us denote the input ground motion records as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t, \dots, \mathbf{x}_n\}^T \in \mathbb{R}^{n \times p}$ and the output structural response as $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t, \dots, \mathbf{y}_n\}^T \in \mathbb{R}^{n \times q}$ with n time steps, where p and q represent the number of input and output features, respectively. Both \mathbf{X} and \mathbf{Y} are matrices whose rows stand for time steps and columns for features (e.g., ground acceleration/velocity/displacement as the input and story-wise accelerations/velocities/displacements as the output). The LSTM-f network takes the ground motion sequence (\mathbf{X}) as input and maps it to the output of structural response sequence (\mathbf{Y}), where the time steps cover the full duration ($t = 1, 2, \dots, n$) and each cell within a LSTM layer is connected with its two neighbors via $\{t-1, t, t+1\}$. At each time step, all input features $\mathbf{x}_t = \{x_{t1}, x_{t2}, \dots, x_{tp}\}$ at the current time step are fed into the LSTM network that connects to the output features $\mathbf{y}_t = \{y_{t1}, y_{t2}, \dots, y_{tq}\}$ at current time step. This process continues by sending \mathbf{x}_t to the network through repeated LSTM cells across the entire temporal space, forming a chain-like structure to capture the long-short term time dependencies. Figure 2 illustrates the concept of applying the LSTM-f to model the structural response \mathbf{Y} given the seismic excitation \mathbf{X} .

In summary, the LSTM-f network models the pairwise input/output relationship along the temporal space ($t = 1, 2, \dots, n$). However, a major drawback of LSTM-f is that it requires a huge amount of training efforts especially for long sequences that require a large size of computation memory. To address this issue, a sequence stacking method is proposed to reduce the memory size as introduced in Section 2.4.

2.4. Stacked sequence to sequence LSTM network (LSTM-s)

Herein we present the LSTM-s scheme which takes the stacked input of a certain number (e.g., w steps) of time steps, namely, $\{\mathbf{x}_{t-w+1}, \mathbf{x}_{t-w+2}, \dots, \mathbf{x}_t\}^T$, to predict the output \mathbf{y}_t . To better illustrate the concept of LSTM-s, let us assume we only have one feature for both input and output (e.g., $p = q = 1$). Figure 3 shows the schematic of stacked input and output for the proposed LSTM-s scheme. The original sequences of input $\mathbf{X} = \{x_1, x_2, \dots, x_n\}^T \in \mathbb{R}^{n \times 1}$ and output $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}^T \in \mathbb{R}^{n \times 1}$ are divided into multiple stacks, which forms the new input $\tilde{\mathbf{X}} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{\tilde{t}}, \dots, \tilde{x}_s\}^T \in \mathbb{R}^{s \times w}$ and new output $\tilde{\mathbf{Y}} = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{\tilde{t}}, \dots, \tilde{y}_s\}^T \in \mathbb{R}^{s \times 1}$, where n is the total number of time steps and s is the number of stacks. Thus, each stack is considered as one time step and the reduced temporal space is constructed with the total time steps equal to the number of stacks s (e.g., $\tilde{t} = 1, 2, \dots, s$). Note that $sw \leq n$ (zero-padding can be applied if $sw = n$ doesn't hold). Each input stack $\tilde{x}_{\tilde{t}} \in \mathbb{R}^{1 \times w}$ contains a fixed length of the original input sequence, e.g., $\{x_{(\tilde{t}-1)w+1}, x_{(\tilde{t}-1)w+2}, \dots, x_{\tilde{t}w}\}$, and is considered

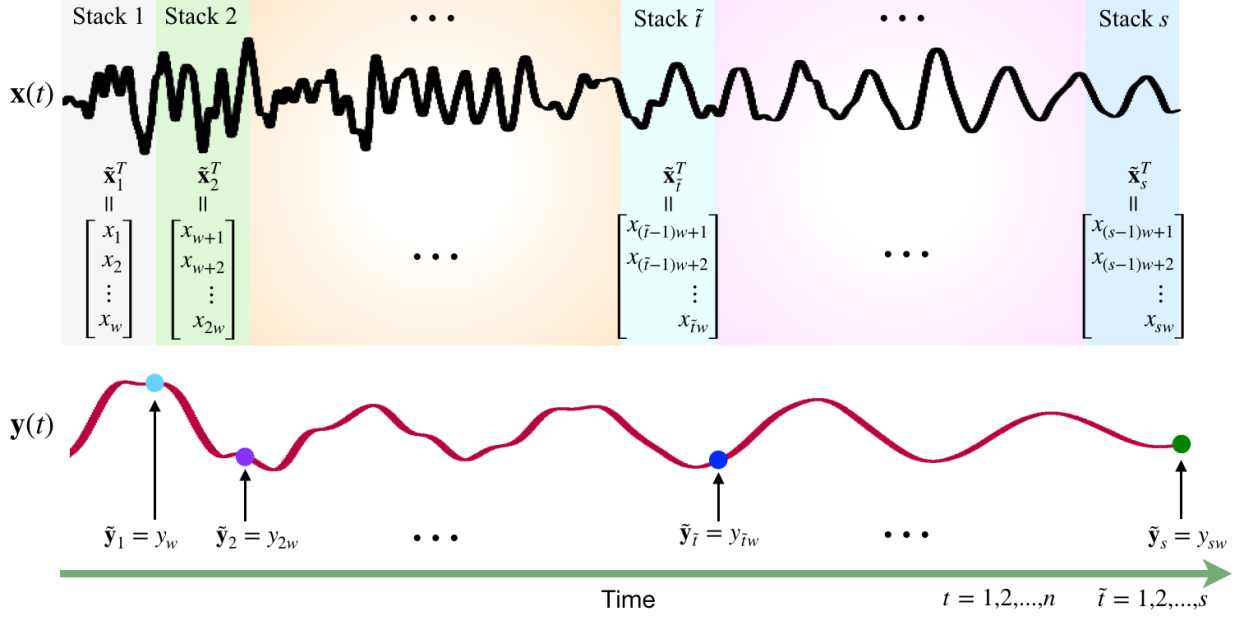


Figure 3: Schematic of stacked input and output for LSTM-s. Note that the width of the stack box shown in the figure does not represent the actual stack size, which is for illustration only. Here, we have $sw \leq n$.

as the new input features for predicting the output feature $\tilde{y}_{\tilde{t}} = y_{\tilde{t}w} \in \mathbb{R}^{1 \times 1}$ at time step \tilde{t} , where w denotes the size of each stack.

Essentially, LSTM-s reduces the temporal dimension by stacking a window of the input sequence and augmenting the corresponding feature space. Since the new output $\tilde{\mathbf{Y}}$ is resampled every w steps from the original output sequence (e.g., $\tilde{\mathbf{Y}} \in \mathbf{Y}$), the stack size w must be carefully selected such that $\tilde{\mathbf{Y}}$ is able to accurately approximate the profile of \mathbf{Y} . Noteworthy, w is an empirical number and depends on the sampling rate, smoothness of the sequence as well as the dominant mode of the dynamic response. For example, if the output \mathbf{Y} is the displacement time history of a building sampled at a typical rate of 100 (or 200) Hz, w can be selected from the range of (2, 10). Now, considering input \mathbf{X} with p features and output \mathbf{Y} with q features, the stacked input and output can be obtained as $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{\tilde{t}}, \dots, \tilde{\mathbf{x}}_s\}^T \in \mathbb{R}^{s \times pw}$ and $\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \dots, \tilde{\mathbf{y}}_{\tilde{t}}, \dots, \tilde{\mathbf{y}}_s\}^T \in \mathbb{R}^{s \times q}$, where $\tilde{\mathbf{x}}_{\tilde{t}} \in \mathbb{R}^{1 \times pw}$ and $\tilde{\mathbf{y}}_{\tilde{t}} \in \mathbb{R}^{1 \times q}$. It is worth mentioning that the LSTM-s scheme can reduce the required computer memory for training the network and thus save the computational cost. In addition, LSTM-s is observed to provide better training and prediction performance as compared to LSTM-f as illustrated in Section 3.

Remark: Another choice of stacking the output $\tilde{\mathbf{y}}_{\tilde{t}}$ is based on the mean value of the sequence $\{\mathbf{y}_{(\tilde{t}-1)w+1}, \mathbf{y}_{(\tilde{t}-1)w+2}, \dots, \mathbf{y}_{\tilde{t}w}\}$, which is given by

$$\tilde{\mathbf{y}}_{\tilde{t}} = \frac{1}{w} \sum_{i=1}^w \mathbf{y}_{(\tilde{t}-1)w+i} \quad (8)$$

3. K-means-based Unsupervised Learning for Data Sequence Clustering

Selecting training/validation datasets plays a significant role in deep learning as it affects the feature extraction from available database. For example, given a limited number of datasets composed of ground motions (GMs) and the corresponding structural responses, one needs to perform the GM selection (e.g., using clustering methods) to determine the optimal partition of training/validation datasets when using surrogate approaches (e.g., deep learning) to model earthquake-induced structural responses. Since the GM input contains different magnitudes and complex frequency contents, clustering can uncover the hidden patterns and characteristics within the GM sequences so as to reduce the number of required training datasets, diversify/maximize the use of information in the available datasets, and improve the prediction performance of the trained deep learning model. Thus, we employ the modified K-means algorithm, namely, TK-means, presented by Huang et al. [49] for sequence clustering.

Similar to the K-means algorithm, the TK-means evaluates the weighted distance between m time series $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m\}^T$ and the centroids of k clusters $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\}$, given by

$$J_0(\mathbf{\Lambda}, \mathbf{W}, \mathbf{C}) = \sum_{p=1}^k \sum_{i=1}^m \sum_{j=1}^n \lambda_{ip} w_{pj} (z_{ij} - c_{pj})^2 \quad (9)$$

Here, both $\mathbf{z}_i = \{z_{i1}, z_{i1}, \dots, z_{in}\}$ and $\mathbf{c}_p = \{c_{p1}, c_{p1}, \dots, c_{pn}\}$ are vectors consisting of sequences with n time steps. Weights are assigned to each time step for every cluster by inducing the weight matrix $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ which contains a set of k vectors. $\mathbf{\Lambda}$ is a $m \times k$ binary matrix, in which the element $\lambda_{ip} = 1$ indicates that the time sequence i is assigned to cluster p ; otherwise, $\lambda_{ip} = 0$. Thus, the clustering problem can be cast as a constrained optimization problem with the regularized objective function defined as

$$J(\mathbf{\Lambda}, \mathbf{W}, \mathbf{C}) = J_0(\mathbf{\Lambda}, \mathbf{W}, \mathbf{C}) + \frac{1}{2}\alpha \sum_{p=1}^k \sum_{j=1}^{n-1} (w_{pj} - w_{p(j+1)})^2 \quad (10)$$

such that

$$\arg \min_{\{\mathbf{\Lambda}, \mathbf{W}, \mathbf{C}\}} J(\mathbf{\Lambda}, \mathbf{W}, \mathbf{C}) \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^m \lambda_{ip} > 0, \lambda_{ip} \in \{0, 1\} \\ \sum_{p=1}^k \lambda_{ip} = 1 \\ \sum_{j=1}^n w_{pj} = 1, 0 \leq w_{pj} \leq 1 \end{cases} \quad (11)$$

Note that the definition of J_0 aims at minimizing the weighted total distance of scatters of all the clusters, while the second term in J shown in Eq. (10) tries to smooth the weights of the adjacent time steps through a smoothing parameter α given by

$$\alpha = \sum_{i=1}^m \sum_{j=1}^n \left(z_{ij} - \frac{\sum_{i=1}^m z_{ij}}{m} \right)^2 \quad (12)$$

The optimization problem can be solved by an iterative sequential procedure. First, \mathbf{W} and \mathbf{C} are randomly initialized, and $\mathbf{\Lambda}$ can be directly solved by

$$\lambda_{ip} = \begin{cases} 1, & \text{if } D_{pj} \leq D_{p'j}, p' \neq p, 1 \leq p' \leq k \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where

$$D_{pj} = \sum_{j=1}^n w_{pj} (z_{ij} - c_{pj})^2 \quad (14)$$

Next, the centroid matrix \mathbf{C} will be updated from \mathbf{W} and $\mathbf{\Lambda}$ using the equation as follows

$$c_{pj} = \frac{\sum_{i=1}^m \lambda_{ip} z_{ij}}{\sum_{i=1}^m \lambda_{ip}} \quad (15)$$

With updated \mathbf{C} and $\mathbf{\Lambda}$, the weight matrix \mathbf{W} can be solved using quadratic programming by minimizing $J(\mathbf{\Lambda}, \mathbf{W}, \mathbf{C})$. The whole process repeats iteratively until the convergence of the objective function is achieved.

4. Examples: Numerical and Experimental Validation

This section presents the implementation of the proposed LSTM architecture for structural seismic response modeling and prediction. In order to comprehensively demonstrate the robustness and efficiency of the LSTM-f and LSTM-s schemes, three illustrative examples were conducted considering both synthetic data and field data measurements. First, the proposed approach was validated on a numerical Bouc-Wen model with large nonlinearity in comparison with a classical artificial neural network. Next, the proposed LSTM-s architecture was implemented to estimate the seismic response of a 6-story hotel building in California using the field-sensing data. In the last example, our approach was successfully applied to predict the nonlinear responses of a 3-story steel moment resisting frame (MRF) under unseen seismic excitations. It is noted that the input is earthquake ground accelerations and the output is structural displacements or inter-story drifts in all examples.

Training the neural networks is performed in the Python environment using Keras [50] [51]. Keras is a high-level open source deep learning library, built on top of TensorFlow, which offers easy and fast prototyping neural networks. TensorFlow, a symbolic math library for machine learning applications developed by Google Brain Team [52], is served as the backend engine in Keras. It offers flexible data flow architecture enabling high-performance training of various types of neural networks across a variety of platforms (CPUs, GPUs, TPUs). The simulations in this paper are performed on a standard PC with 28 Intel Core

i9-7940X CPUs and 2 NVIDIA GTX 1080Ti GPU cards. The data and codes used in this paper will be publicly available on GitHub at <https://github.com/zhry10/DeepLSTM> after the paper is published.

4.1. Numerical example: Bouc-Wen

A numerical study was conducted using the proposed LSTM architecture to predict the seismic responses of a 5 degrees-of-freedom (DOF) nonlinear hysteretic system [53]. To generate the database for training the deep learning models, we first write the equation of motion of the nonlinear system as follows:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{f}(t) = -\mathbf{M}\Gamma\ddot{\mathbf{x}}_g(t) \quad (16)$$

where \mathbf{M} and \mathbf{C} are the mass and damping matrices; \mathbf{x} , $\dot{\mathbf{x}}$, and $\ddot{\mathbf{x}}$ are the relative displacement, velocity, and acceleration vector to the ground; $\ddot{\mathbf{x}}_g$ represents the ground acceleration; Γ is the force distribution vector; and \mathbf{f} is the nonlinear restoring force vector expressed by the Bouc-Wen model [54, 55], where the i th component of vector \mathbf{f} is presented as [56]:

$$\dot{f}_i = k_i \Delta \dot{x}_i - \alpha_i |\Delta \dot{x}_i| |f_i|^{n_i-1} f_i - \beta_i \Delta \dot{x}_i |f_i|^{n_i} \quad (17)$$

where Δx_i is the relative displacement between $(i-1)$ th and i th DOF, which is denoted as $\Delta x_i = x_i - x_{i-1}$ for $i \geq 2$ and $\Delta x_i = x_1$ if $i = 1$; k_i is the stiffness of the i th DOF; and α_i, β_i , and n_i are the nonlinear parameters of the Bouc-Wen model. The following Bouc-Wen parameters are used to generate datasets to train the LSTM-f and LSTM-s: $m_1 = 100$ kg, $m_2 = m_3 = m_4 = m_5 = 80$ kg, $c_1 = c_3 = c_5 = 0.55$ kNs/m, $c_2 = c_4 = 0.5$ kNs/m, $k_1 = k_3 = k_5 = 30$ kN/m, $k_2 = k_4 = 24$ kN/m, $\alpha_1 = \alpha_3 = \alpha_5 = 1$, $\alpha_2 = \alpha_4 = 2$, $\beta_1 = \beta_3 = \beta_5 = 2$, $\beta_2 = \beta_4 = 1$, $n_1 = n_3 = n_5 = 3$, $n_2 = n_4 = 2$. The natural frequencies of the system are 0.83 Hz, 2.39 Hz, 3.74 Hz, 4.80 Hz, and 5.57 Hz.

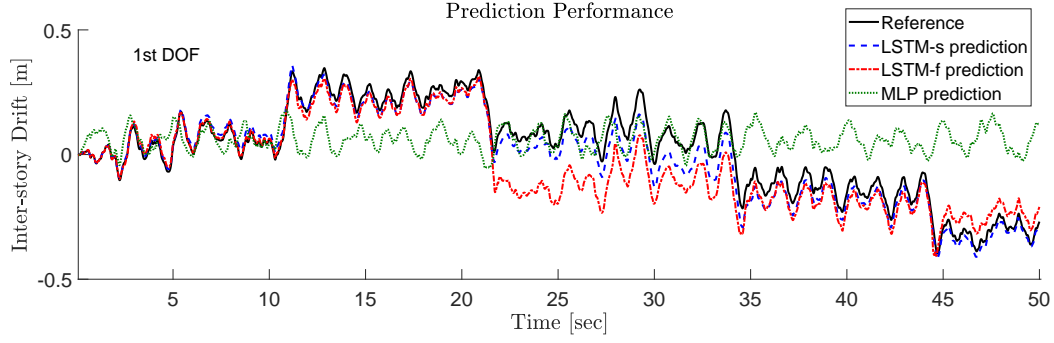
A number of 100 samples (datasets) are generated by the numerical simulation of the 5DOF nonlinear system excited by random band-limited white noise (BLWN) ground motion sequences with different magnitudes. Note that alternatives of generating more realistic earthquake excitations are also applicable herein, which can be found in [57, 58]. Each sample consists of the ground acceleration as input and structural inter-story drifts as output. The entire datasets are divided into three subsets, including the training dataset with 37 samples, the validation dataset with 13 samples, and the prediction dataset with 50 samples which is assumed as unknown dataset for testing the predictability performance of the trained model. All samples are formatted in the required format for LSTM-f and LSTM-s aforementioned. Since a stack size of $w = 20$ is used in LSTM-s, the input and output size of [37, 500, 20] and [37, 500, 5]. In LSTM-f, the input and output formats are [37, 10001, 1] and [37, 10001, 5]. Both LSTM-f and LSTM-s have the same LSTM architecture for comparison purpose, including two LSTM layers and two FC layers in addition to the input and output layer, with the network parameter setting given by: $p = 1$ (LSTM-f) or $p = 20$ (LSTM-s), $p_1 = p_2 = q_1 = 100$, and $q = 5$. Note that the number of nodes for the last FC layer must

be equal to the number of output features which is five (corresponding to 5 DOF) in this study.

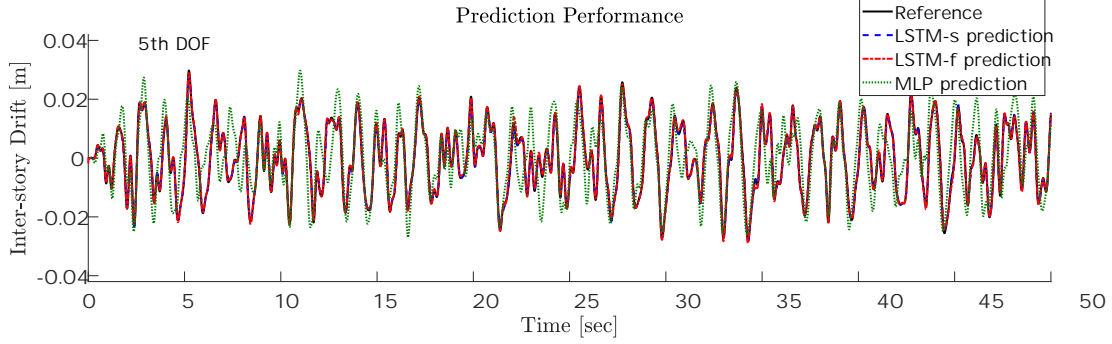
The input and output of the training dataset are fed into the deep LSTM models for training. Due to the small number of samples in the training dataset, a batch size of ten is used which means ten samples of the training dataset are used to evaluate the gradient of the loss function and update the learnable parameters (e.g., weights and biases). The weights of all LSTM and FC layers are randomly initialized using the *glorot* uniform initializer [59], while the biases are initialized as zero. To accelerate the convergence of training such deep LSTM models, Adam (Adaptive Momentum Estimation) is selected as the optimizer, thanks to its low memory requirement, with a learning rate of 0.001 and a decay rate of 0.0001 [60]. Note that the current setup outperforms other options of optimizers (e.g., SGD, RMSprop) and initializers (e.g., *glorot* normal, random uniform distribution) in Keras. The objective of the training process is to minimize the loss function for each batch which is determined as the mean squared error (MSE) for regression problems.

The training phase consists of 50000 epochs to ensure the convergence of the loss function. To improve the training performance, all data are scaled to $[-1, 1]$ using the *MinMaxScaler* in the scikit-learn toolkit [61]. In addition, the training data are shuffled before each epoch which can reduce model variance and speed up the convergence [62]. During the training process, the model is validated through the validation dataset after each epoch with the validation error monitored. The weights and biases are updated iteratively through the back-propagation algorithm to reduce the loss function. Note that the model will not be trained using the validation dataset. Both the training and validation errors decrease during the initial phase of training. However, the validation error rises when overfitting occurs. In this example, both LSTM-f and LSTM-s can lead to very small training and validation loss values, indicating excellent training performance. The LSTM-s performs relatively better than LSTM-f with much smaller training and validation loss values. The trained network with the minimum validation error of 5.1012×10^{-5} for LSTM-f and 6.9661×10^{-6} for LSTM-s are saved as the best models and used to predict structural inter-story drifts under unknown/unseen excitations. Another benefit of LSTM-s is that it is much computationally efficient compared to LSTM-f, as indicted by the total training time of 4.91 hours for LSTM-s *v.s.* 67.11 hours for LSTM-f.

The trained models using both LSTM-s and LSTM-f are used to predict structural inter-story drifts given new ground excitations. Our methods are also compared to the classical multi-layer perceptron (MLP) algorithm [37] that used the same training and prediction datasets. The data format of the MLP takes the incremental form: the ground motion at time steps $\{t-d, \dots, t-1, t\}$ and the inter-story drifts at $\{t-d, \dots, t-1\}$ are considered as the input (e.g., $\{\mathbf{x}_{t-d}, \dots, \mathbf{x}_t, \dots, \mathbf{y}_{t-d}, \dots, \mathbf{y}_{t-1}\}$) to predict the inter-story drift \mathbf{y}_t at the current time step t , where $d \geq 1$. The MLP consists of four hidden layers with the node configuration of [256, 128, 64, 5]. The same Adam optimizer is used for a comparative study. Figure 4(a) and Figure 4(b) show one example of the predicted inter-story drifts using LSTM-s, LSTM-f and MLP for the 1st and 5th DOF (which typically have the largest and smallest inter-story drift for this nonlinear system), respectively. The predicted inter-story drifts using both LSTM-s and LSTM-f reasonably match the reference well for the 1st



(a) The inter-story drift of the 1st DOF



(b) The inter-story drift of the 5th DOF

Figure 4: Predicted responses (inter-story drifts) of the 5DOF nonlinear system for unknown BLWN excitations using LSTM-s, LSTM-f and MLP.

and 5th DOF, while the MLP is incapable of modeling the nonlinear response dominated by plastic deformation. Even for the 5th DOF where slight nonlinearity is present, the MLP prediction is less accurate compared to the LSTM approaches. Overall, the most accurate prediction of plastic deformation is made by LSTM-s as shown in Figure 4(a), demonstrating that the proposed LSTM-s scheme is an effective approach for seismic response modeling and prediction.

To better illustrate the prediction error, both regression analysis and normalized error distribution are performed for both LSTM-s and LSTM-f for all the prediction datasets (see Figure 5). The correlation coefficient r_i is defined as follows

$$r_i = \mathcal{R} \left(\mathbf{y}_i^{true} - \mathbf{y}_i^{pred} \right) \quad (18)$$

where i denotes the i th predicted response; \mathcal{R} represents the regression operation. The probability density function (PDF) of the normalized error distribution, π_i , is defined as

$$\pi_i = \text{PDF} \left\{ \frac{\mathbf{y}_i^{true} - \mathbf{y}_i^{pred}}{\max(|\mathbf{y}_i^{true}|)} \right\} \quad (19)$$

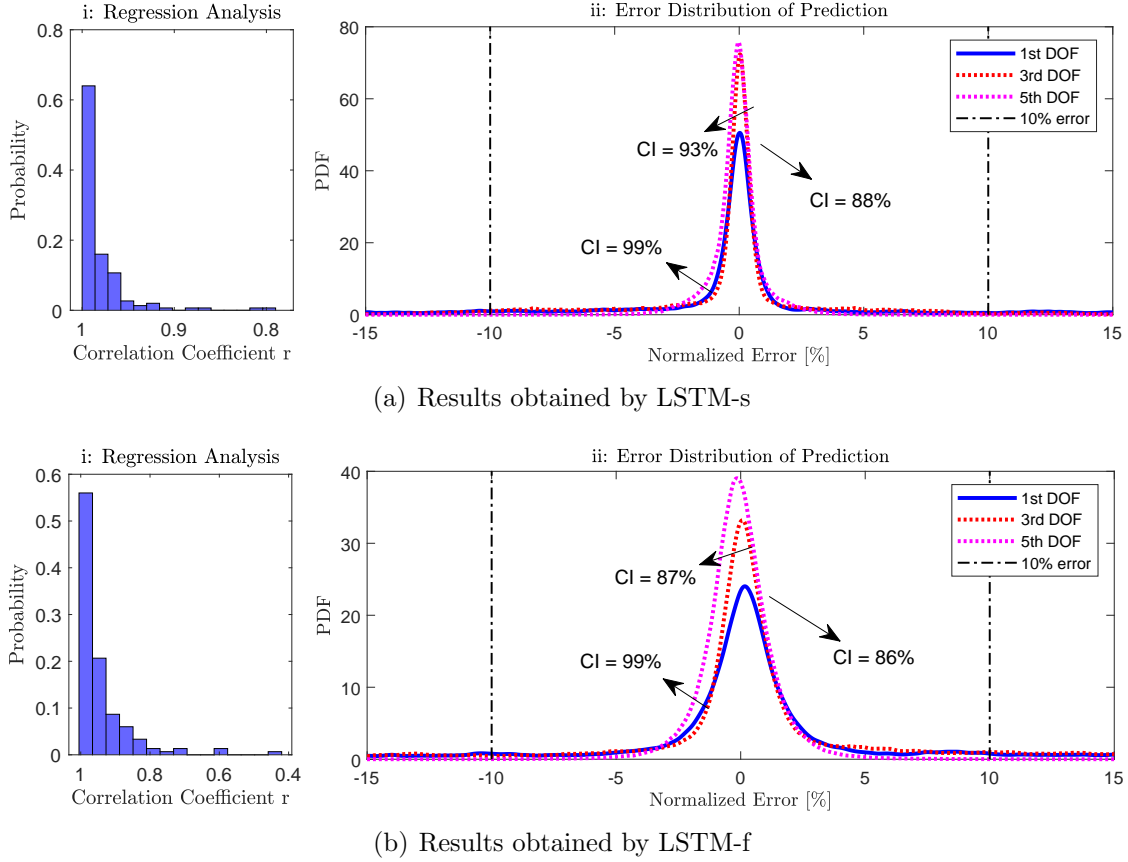


Figure 5: Regression analysis in (i) and Error distribution in (ii) for all prediction datasets using LSTM-s and LSTM-f.

In general, both LSTM-s and LSTM-f provide satisfactory prediction with the correlation coefficient larger than 0.8 for most samples, as shown in Figure 5. Notably, LSTM-s performs stably with the majority of correlation coefficients over 0.9, while the LSTM-f prediction has many regression coefficients below 0.8 (e.g., the coefficients are less than 0.5 in very limited cases). Similar performance is observed in the error distribution as shown in Figure 5. We can observe from the distributions that LSTM-s general produces smaller errors (e.g., the PDFs are sharper and narrower). In addition, it is seen that the LSTM-s prediction errors that are mainly located within $\pm 10\%$ for the 1st, 3rd, and 5th DOF have the confidence intervals (CIs) of 88%, 93%, and 99% respectively, while the LSTM-f achieves a CIs of 86%, 87%, and 99% for the corresponding cases. The reason of resulting in a higher CI for upper stories is that those stories mainly stay in the linear range whose responses are much easier to train and predict compared to the large plastic deformation of the 1st DOF. Considering both computational efficiency and accuracy, the LSTM-s scheme is proven to be the best model (as compared to LSTM-f and MLP) which is used for response modeling in the following two examples.

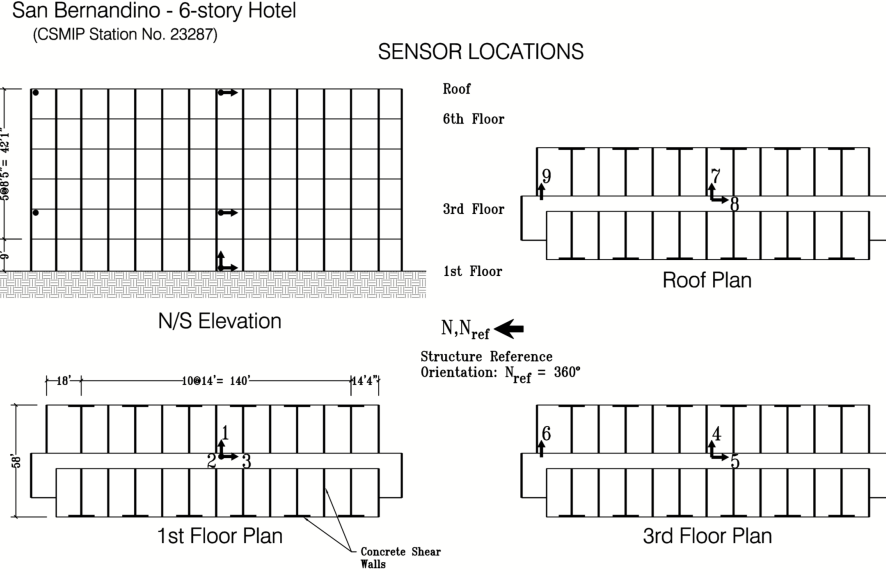


Figure 6: Sensor layout of the 6-story hotel in San Bernardino (Station No. 23287) (<http://www.strongmotioncenter.org/>).

4.2. Field sensing validation: a 6-story instrumented building

The proposed methodology is further demonstrated using field sensing data. A 6-story hotel building in San Bernardino, California adopted from the Center for Engineering Strong Motion Data (CESMD) is investigated [63]. The target structure is a mid-rise concrete building designed in 1970 with a total of nine accelerometers installed on the 1st floor, 3rd floor, and the roof level as shown in Figure 6. The sensors (see the locations in Figure 6) have recorded multiple seismic events from 1987 to 2018. The historically recorded data are used to train the proposed LSTM-s deep learning model for predicting structural displacements subjected to ground motions.

The field data, which contains different sampling rates and high-frequency noise, are first preprocessed using resampling (at 100 Hz) and filtering techniques. A total of 23 available datasets from CESMD are formatted and divided into three subsets, including 11 datasets for training, 4 datasets for validation, and 8 datasets for prediction. A small stack size of $w = 2$ is used in LSTM-s in this example, leading to the input and output size of $[11, 3600, 2]$ and $[11, 3600, 2]$ for the training datasets. More details on the historical data can be found in [45].

The same LSTM architecture (as used in the first example) is adopted here except that only 30 units (nodes) are assigned in each LSTM layer. The training phase consists of 20000 epochs with a batch size of two. The trained model is achieved with the best validation error of 3.703×10^{-7} , and used to predict structural displacements of the 6-story hotel under unseen/unknown/new earthquakes. By simply feeding the seismic input into the trained model, the predicted responses can be obtained as shown in Figure 7 in comparison with the ground truth. Excellent agreement can be observed between the LSTM-s prediction and the reference, for earthquakes with different magnitudes and frequency contents. Moreover,

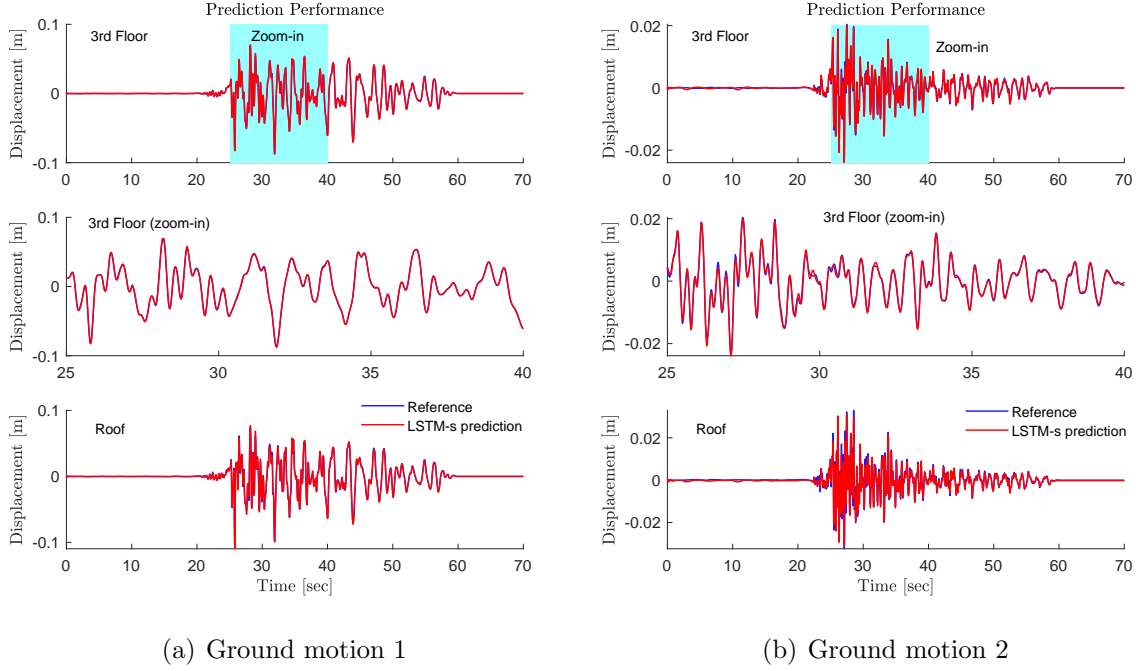


Figure 7: Predicted structural displacements under two example earthquakes using LSTM-s.

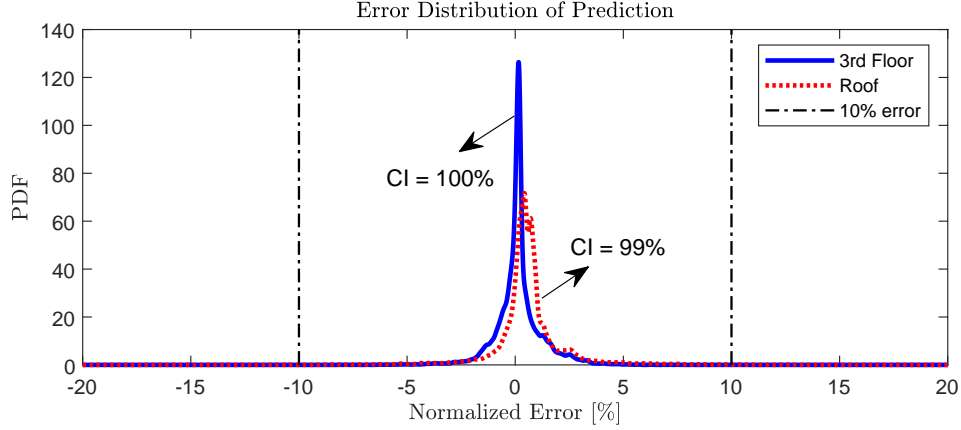


Figure 8: Error distribution of all prediction datasets using the proposed LSTM-s model.

the normalized prediction error distribution is presented in Figure 8, showing CIs of 100% and 99% within $\pm 10\%$ of error for the 3rd floor and roof, respectively. In fact, the majority of prediction errors lie within $\pm 3\%$, demonstrating excellent performance of the proposed LSTM-s approach for structural seismic response prediction.

4.3. 3-story Steel MRF

In this study, we test and validate our approach on a full scale 3-story office building. The prototype building adopted from Dong et al. [50] is assumed to be on a stiff site in

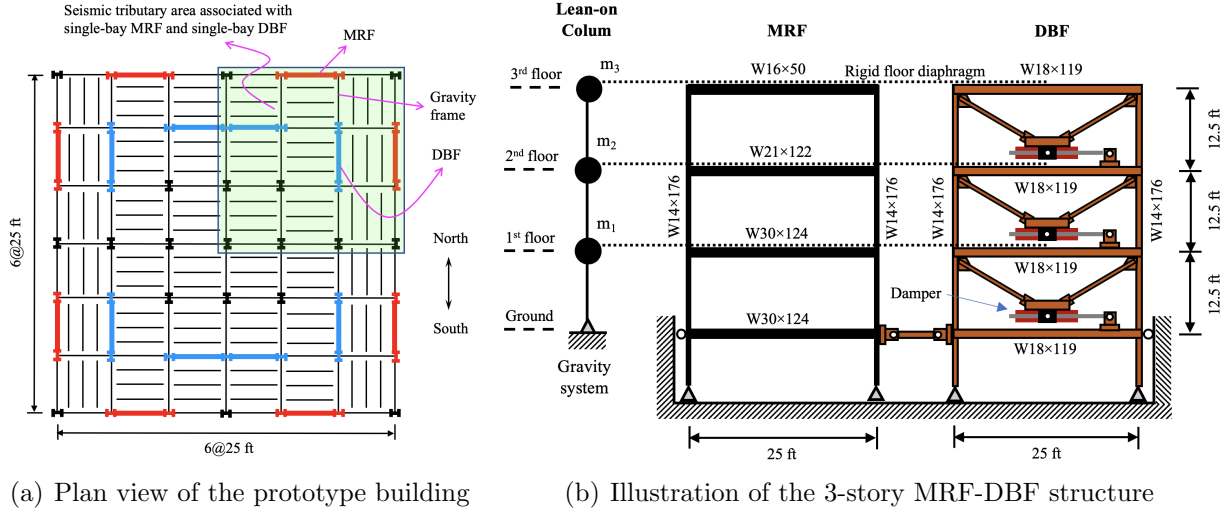


Figure 9: The 3-story steel MRF building.

Pomona, California. Figure 9(a) shows the plan view of the building. The overall dimensions of the prototype structure are 45.7 m (150 ft) by 45.7 m (150 ft) in plan and 11.43 m (37.5 ft) in elevation. The structural system of the building includes a lateral resisting system, a damping system, and a gravity load system. The lateral resisting system consists of 8 identical single-bay moment resisting frames (MRFs). The damping system consists of 8 single-bay frames with nonlinear viscous dampers and associated bracing, termed as damped braced frames (DBFs). The gravity load system includes the uniformly distributed gravity frames in plan. The floor is assumed to be rigid thus the MRFs, DBFs, and the gravity system are assumed to deform together in each horizontal direction. Due to the symmetry of the prototype building, only one quarter of the floor plan within the seismic tributary area (as shown in Figure 9(a)) is considered, forming the prototype structure investigated in this study. The 3-story prototype structure shown in Figure 9(b) consists of a single-bay MRF, an associated single-bay DBF, and the gravity load system with associated seismic mass. The horizontal displacement at the ground level is restrained, and the columns are fixed at the base level. The design details of this structure can be found in the reference [50].

The prototype structure shown in Figure 9(b) is modeled by the nonlinear computational platform, RT-Frame2D, developed in the *Embedded* function under MATLAB/Simulink environment [64, 65]. To preserve stability for nonlinear dynamic analysis, the explicit unconditionally-stable integration scheme developed by Chen and Ricles [66] is adopted. A concentrated plasticity model is employed for the nonlinear beam-column elements in RT-Frame2D, assuming that yielding occurs at the element ends. A bilinear moment-curvature hysteresis material model, with kinematic hardening and a post yielding ratio of 2.5%, is applied. Panel zone elements are used to model the shear deformation and the uniform bending deformation of the MRF panel zones. The element properties include the linear flexural rigidity (EI), axial rigidity (EA), shear rigidity (GA) and yield curvature κ . Mass is

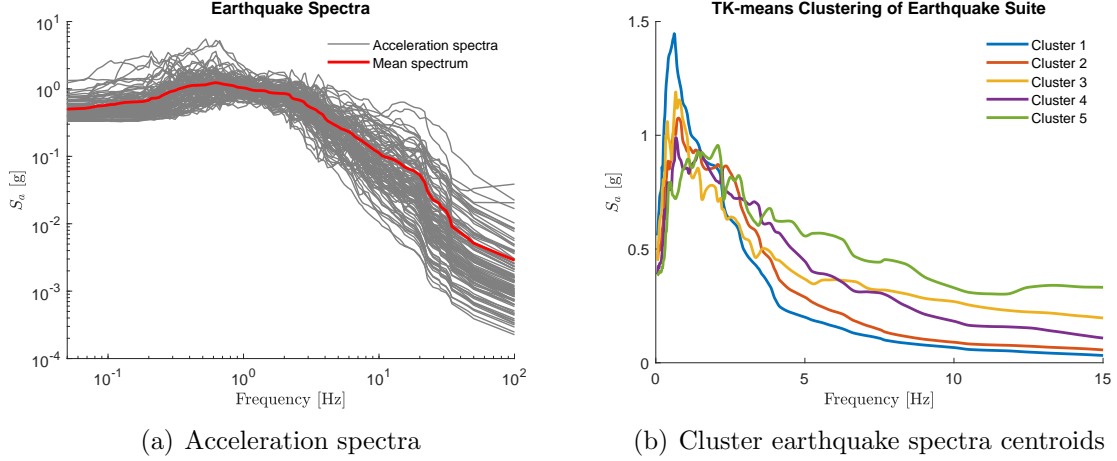


Figure 10: A suite of 100 earthquakes used in this study.

assigned as 4.78×10^5 kg and 5.17×10^5 kg distributed over beam elements at the first/second and third floor respectively for global mass matrix assembling. The gravity load system is represented by the lean-on column, which is modeled with elastic beam-column elements. The seismic mass is lumped and the gravity load is applied at each floor level on the lean-on column so that P-delta effects are included in the nonlinear analysis. The lean-on column is connected to the MRF using a rigid diaphragm. The inherent damping ratios of the first two modes are assigned as 2% using Rayleigh damping. This does not account for energy dissipation from inelastic response of the MRF, which is included directly by the nonlinear elements. The natural frequencies are 1.02 Hz, 3.61 Hz, and 8.32 Hz for the first three modes. More details on the numerical modeling can be found in [64, 65].

A synthetic database is generated to further illustrate the performance of LSTM-s, with a suite of 100 earthquake records selected from the PEER strong motion database [67] in the area of Pomona, California (latitude, longitude = 34.0608° N, 117.7558° W) with a 10% probability of exceedance in 50 years. The selected ground motion records are scaled such that the mean response spectrum matches the design spectrum of the prototype building. Figure 10(a) shows the acceleration spectra of the 100 selected earthquake records. The incremental dynamic analysis (IDA) is conducted for each ground motion record with scaled intensities to simulate different levels of structural damages and nonlinear responses composed of both elastic and plastic deformation, yielding an ensemble of 548 datasets for the prototype structure. Noteworthy, each dataset contains the input ground acceleration and output inter-story drifts. The entire database is divided into training, validation, and prediction datasets (e.g., using TK-means described later). Both training and validation datasets are considered as “known datasets” where both input ground motions and output structural inter-story drifts are fully given for training the LSTM-s model, while the prediction dataset is considered as “unknown ground truth datasets” only for testing purpose.

To show the robustness of the LSTM-s approach, only a limited number of datasets is selected and used for the model training: the suite of 100 ground motion records was

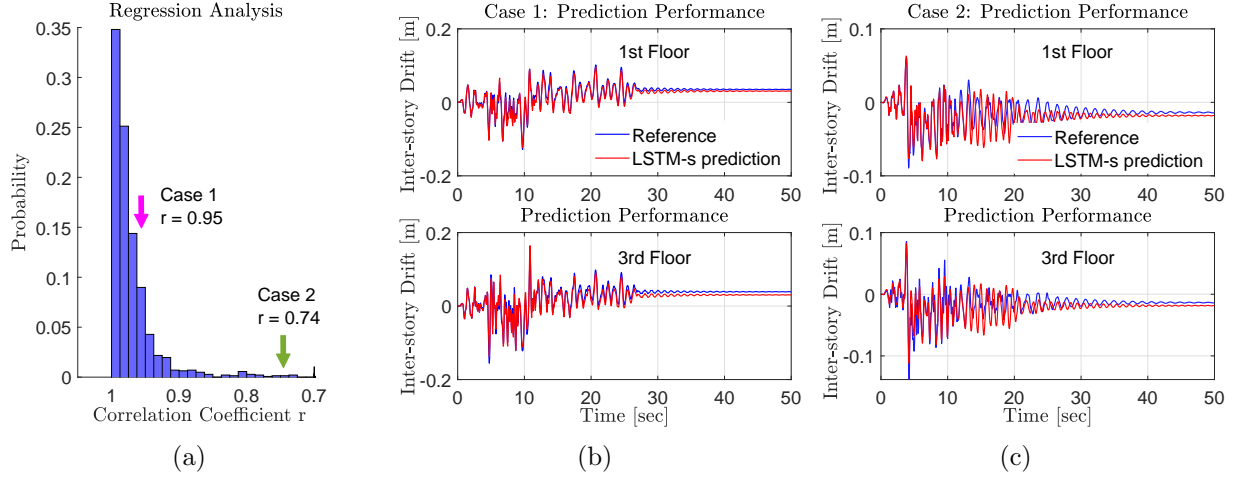


Figure 11: Regression analysis in (a) and two examples of predicted inter-story drifts of the MRF in (b) and (c) for unknown earthquakes using LSTM-s. Note that the cases for the regression correlation coefficients larger than 0.95 are not shown since the predicted response has an excellent match with the ground truth.

first clustered into five clusters using the dynamic TK-means algorithm introduced in Section 3, which groups the earthquake spectra with similar frequency contents. Since IDA is conducted for magnitude effects, the earthquake excitations are clustered based on the normalized spectral accelerations (S_a). Figure 10(b) shows the classified cluster centroids for the suite of 100 earthquakes using the TK-means algorithm. Only four earthquake records are selected from each cluster as known datasets for training/validation, three of which are selected for training (i.e., including the one closest to the cluster centroid and the two spectra farthest from the centroid representing the cluster boundaries). The fourth one is randomly selected from each cluster and regarded as the validation dataset. Therefore, the ground motion selection process, together with IDA, yields 47 training datasets from 15 selected ground motion records and 20 validation datasets from 5 selected ground motion records. A total of 481 datasets from the rest 80 earthquakes are considered as prediction datasets, which is unseen in the training process, to test the LSTM-s predictability.

Firstly, all the training/validation datasets are reshaped into the input/output format required by LSTM-s aforementioned, e.g., the training input and output sizes are [47, 1000, 10] and [47, 1000, 3] while the validation has the corresponding dimensions of [15, 1000, 10] and [15, 1000, 3], given $w = 10$. The LSTM architecture shown in Figure 2 is employed, with the network parameter setting given by: $p = 10$, $p_1 = p_2 = q_1 = 100$, and $q = 3$.

The training phase consists of 50000 epochs to fully capture the elastic and plastic behaviors of the MRF. The proposed LSTM-s is able to achieve a small validation error of 7.3244×10^{-6} even with limited training datasets. The trained model is then used to predict structural inter-story drifts given new ground motion records. Figure 11(a) shows the regression analysis for the 481 prediction datasets. It is seen that satisfactory correlation coefficients over 0.9 (namely, $r \geq 0.9$) are achieved for the majority of prediction samples. To fully illustrate the prediction performance of our LSTM-s model, the time histories

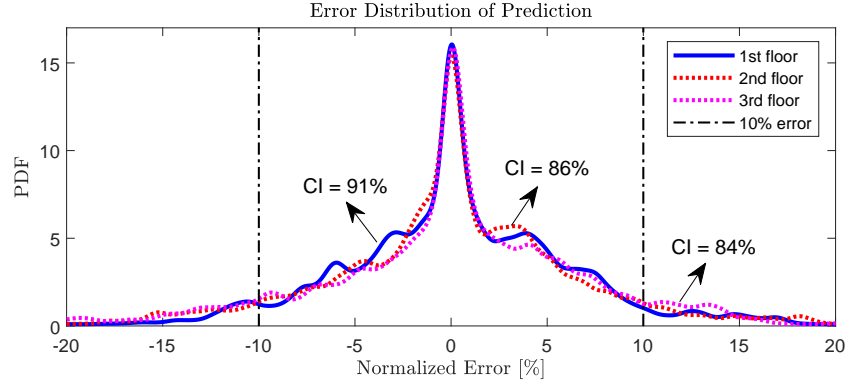


Figure 12: Error distribution of all prediction datasets using the proposed LSTM-s model.

of predicted inter-story drifts are presented in Figure 11(b) and (c) corresponding to the correlation coefficient of 0.95 and 0.74 (the lowest r) respectively. The LSTM-s prediction in Figure 11(b), where $r = 0.95$, matches the reference well in both magnitudes and phases. Note that the prediction results for $r \geq 0.95$ are not shown since the predicted inter-story drifts have an excellent match with the ground truth. Even for the case of less satisfactory prediction shown in Figure 11(c) where $r = 0.74$, our LSTM-s approach is still able to reasonably well predict the inter-story drift time histories, especially retaining an accurate prediction of the peak response and the residual inter-story drift which are typically the most important parameters in performance-based evaluation of structures. This clearly shows the bottom limit of our proposed LSTM-s approach.

The overall accuracy and robustness of the trained LSTM-s model can be reflected from the normalized error distribution defined in Eq.19 for all 481 prediction datasets (see Figure 12). It is seen that the prediction errors are mainly located within $\pm 10\%$ with CIs of 91%, 86%, and 84%, respectively, for the three stories. In addition, the predicted responses for IDA under same ground motion but with different intensities are also investigated and presented in Figure 13. Although the input earthquakes are scaled linearly, the trained model is capable of capturing and distinguishing the nonlinear structural responses. This clearly illustrates that our proposed LSTM-s approach can accurately model/estimate both elastic and plastic dynamic behaviors, demonstrating promising potential in surrogate modeling of structural systems for seismic response prediction. It is worth mentioning that the trained LSTM-s model can be used to develop seismic fragility curves (e.g., within a Monte Carlo simulation framework) for reliability assessment of nonlinear structural systems for seismic protection.

5. Conclusions

This paper presents a deep learning approach, based on a long short-term memory (LSTM) recurrent neural network, for response modeling and prediction of structural seismic response. The proposed LSTM architecture consists of several LSTM layers and fully-connected layers to model the time dependency and causality of the input and output se-

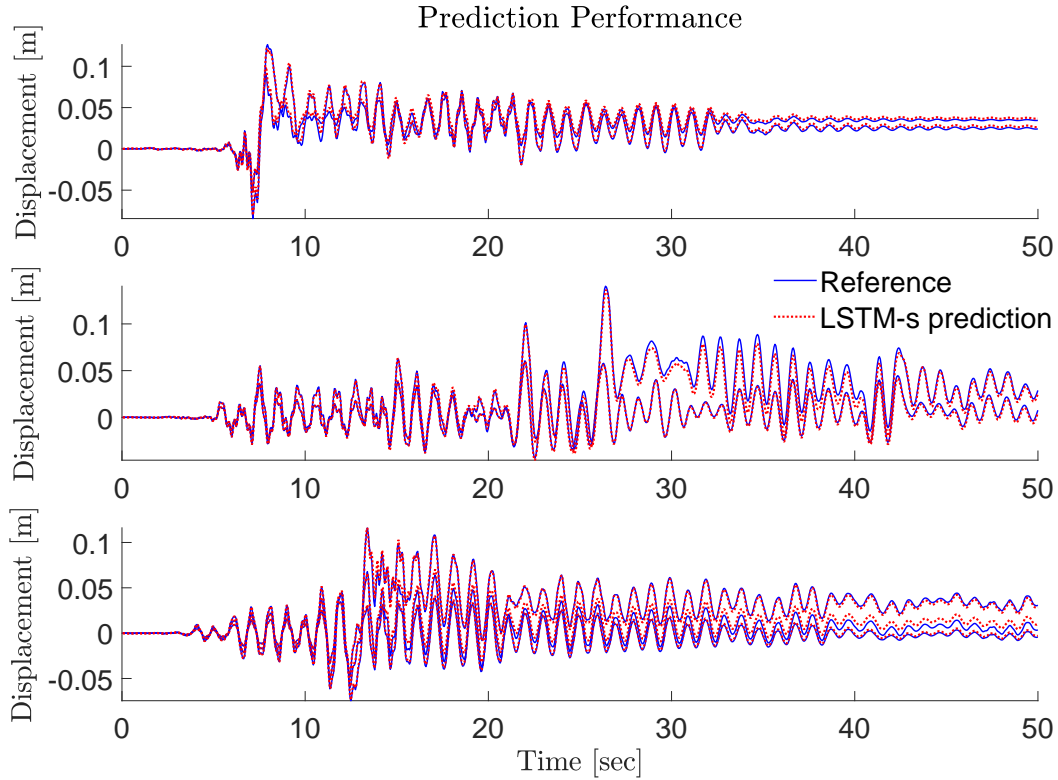


Figure 13: LSTM-s predicted 1st story drifts under three example unknown earthquakes with multi-intensities.

quences. Two schemes of LSTM are developed, namely, the LSTM-f and LSTM-s with different input/output formats. The performance of these two schemes are illustrated via three examples with both synthetic data and field sensing measurements. First, the proposed approach was tested on a 5DOF nonlinear hysteretic system with large nonlinearity described by a Bouc-Wen model. LSTM-s was proven to be the most reliable and computationally cost-effective approach for nonlinear structural seismic response modeling and prediction, in comparison with LSTM-f and the classical MLP approach. The satisfactory performance of the proposed LSTM-s model was further illustrated through an instrumented building (e.g., a 6-story hotel building in San Bernardino, California) with field sensing recordings. Noteworthy, very high prediction accuracy was observed when the building exhibits a close-to-linear behavior without plastic deformations. In the third example, the LSTM-s approach was applied to estimate the seismic response of a 3-story nonlinear steel moment resisting frame (SMRF). A synthetic database was generated through incremental dynamic analysis with a suite of 100 ground motions. Large residual deformation can be observed for the structure under intensive earthquake excitations. To improve the robustness of the LSTM-s model trained on limited data, an unsupervised learning algorithm (i.e., TK-means) was proposed for ground motion selection and thus for effectively clustering the training/validation/prediction datasets from limited data. The highly nonlinear response

of the SMRF structure can be well estimated by the trained LSTM-s model. Overall, the results demonstrate that the proposed LSTM-s approach is a promising, reliable, robust and scalable approach for nonlinear structural response modeling under ground motions with larger intensities. Finally, it is noteworthy that the proposed algorithm is fundamental in nature which is scalable to model the dynamic response of other types of material and structural systems, and offers significant potential in seismic fragility analysis of buildings for reliability assessment.

Acknowledgement

The authors acknowledge the partial support provided by Royal Dutch Shell through the BeeView 2 Project. We also would like to thank chief scientist Dr. Dirk Smit and project managers Dr. Lorna Ortiz-Soto and Dr. Xander Campman for their oversight of this work. The data and codes used in this paper will be publicly available on GitHub at <https://github.com/zhry10/DeepLSTM> after the paper is published.

References

- [1] Zielichowski-Haber W. Sensitivity analysis of uncertain structures subject to stochastic load in the framework of SFEM. In: PAMM: Proceedings in Applied Mathematics and Mechanics, vol. 5: :731–732Wiley Online Library; 2005.
- [2] Di Paola M, Navarra G. Stochastic seismic analysis of MDOF structures with nonlinear viscous dampers. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*. 2009;16(3):303–318.
- [3] Zhang Jian, Wan Chunfeng, Sato Tadanobu. Advanced Markov chain Monte Carlo approach for finite element calibration under uncertainty. *Computer-Aided Civil and Infrastructure Engineering*. 2013;28(7):522–530.
- [4] Vamvatsikos Dimitrios, Cornell C Allin. Incremental dynamic analysis. *Earthquake Engineering & Structural Dynamics*. 2002;31(3):491–514.
- [5] Vamvatsikos Dimitrios, Fragiadakis Michalis. Incremental dynamic analysis for estimating seismic performance sensitivity and uncertainty. *Earthquake engineering & structural dynamics*. 2010;39(2):141–163.
- [6] Tirca Lucia, Serban Ovidiu, Lin Lan, Wang Mingzheng, Lin Nenghui. Improving the seismic resilience of existing braced-frame office buildings. *Journal of Structural Engineering*. 2015;142(8):C4015003.
- [7] Sjöberg Jonas, Zhang Qinghua, Ljung Lennart, et al. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*. 1995;31(12):1691–1724.
- [8] Braun James E, Chaturvedi Nitin. An inverse gray-box model for transient building load prediction. *HVAC&R Research*. 2002;8(1):73–99.
- [9] Belleri Andrea, Moaveni Babak, Restrepo José I. Damage assessment through structural identification of a three-story large-scale precast concrete structure. *Earthquake Engineering & Structural Dynamics*. 2014;43(1):61–76.
- [10] Yousefianmoghadam Seyedsina, Behmanesh Iman, Stavridis Andreas, Moaveni Babak, Nozari Amin, Sacco Andrea. System identification and modeling of a dynamically tested and gradually damaged 10-story reinforced concrete building. *Earthquake Engineering & Structural Dynamics*. 2018;47(1):25–47.
- [11] Yi Ting-Hua, Yao Xiao-Jun, Qu Chun-Xu, Li Hong-Nan. Clustering Number Determination for Sparse Component Analysis during Output-Only Modal Identification. *Journal of Engineering Mechanics*. 2018;145(1):04018122.

- [12] Sohn Hoon, Farrar Charles R, Hemez Francois M, et al. A review of structural health monitoring literature: 1996–2001. *Los Alamos National Laboratory, USA*. 2003;.
- [13] Ching Jianye, Beck James L, Porter Keith A, Shaikhutdinov Rustem. Real-time Bayesian state estimation of uncertain dynamical systems. 2004;.
- [14] Kerschen Gaetan, Worden Keith, Vakakis Alexander F, Golinval Jean-Claude. Past, present and future of nonlinear system identification in structural dynamics. *Mechanical systems and signal processing*. 2006;20(3):505–592.
- [15] Brownjohn James MW, Xia Pin-Qi. Dynamic assessment of curved cable-stayed bridge by model updating. *Journal of structural engineering*. 2000;126(2):252–260.
- [16] Moaveni Babak, Conte Joel P, Hemez François M. Uncertainty and sensitivity analysis of damage identification results obtained using finite element model updating. *Computer-Aided Civil and Infrastructure Engineering*. 2009;24(5):320–334.
- [17] Friswell Michael, Mottershead John E. *Finite element model updating in structural dynamics*. Springer Science & Business Media; 2013.
- [18] Sun Hao, Betti Raimondo. A hybrid optimization algorithm with Bayesian inference for probabilistic model updating. *Computer-Aided Civil and Infrastructure Engineering*. 2015;30(8):602–619.
- [19] Skolnik Derek, Lei Ying, Yu Eunjong, Wallace John W. Identification, model updating, and response prediction of an instrumented 15-story steel-frame building. *Earthquake Spectra*. 2006;22(3):781–802.
- [20] Fishwick Paul A. Neural network models in simulation: a comparison with traditional modeling approaches. In: Proceedings of the 21st conference on Winter simulation:702–709ACM; 1989.
- [21] Ediger Volkan Ş, Akar Sertac. ARIMA forecasting of primary energy demand by fuel in Turkey. *Energy Policy*. 2007;35(3):1701–1708.
- [22] Akaike Hirotugu. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*. 1969;21(1):243–247.
- [23] Durbin James. Efficient estimation of parameters in moving-average models. *Biometrika*. 1959;46(3/4):306–316.
- [24] Said Said E, Dickey David A. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*. 1984;71(3):599–607.
- [25] Irie Bunpei, Miyake Sei. Capabilities of three-layered perceptrons. In: IEEE International Conference on Neural Networks:218; 1988.
- [26] Hornik Kurt. Approximation capabilities of multilayer feedforward networks. *Neural networks*. 1991;4(2):251–257.
- [27] Chen SABS, Billings SA. Neural networks for nonlinear dynamic system modelling and identification. *International journal of control*. 1992;56(2):319–346.
- [28] Tianping Chen, Hong Chen. Approximations of continuous functions by neural networks with application to dynamic system. *IEEE Transaction Neural Networks*. 1993;4(6):910–918.
- [29] Chen Tianping, Chen Hong. Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks. *IEEE Transactions on Neural Networks*. 1995;6(4):904–910.
- [30] Lightbody Gordon, Irwin George W. Multi-layer perceptron based modelling of nonlinear systems. *Fuzzy sets and systems*. 1996;79(1):93–112.
- [31] Huang CS, Hung SL, Wen CM, Tu TT. A neural network approach for structural identification and diagnosis of a building from seismic response data. *Earthquake engineering & structural dynamics*. 2003;32(2):187–206.
- [32] Yi Ting-Hua, Li Hong-Nan, Sun Hong-Min. Multi-stage structural damage diagnosis method based on. *Smart Structures and Systems*. 2013;12(3.4):345–361.
- [33] Comerford Liam, Kougioumtzoglou Ioannis A, Beer Michael. An artificial neural network approach for stochastic process power spectrum estimation subject to missing data. *Structural Safety*. 2015;52:150–160.
- [34] Guarize R, Matos NAF, Sagrilo LVS, Lima ECP. Neural networks in the dynamic response analysis of slender marine structures. *Applied Ocean Research*. 2007;29(4):191–198.

- [35] Ying Wang, Chong Wang, Hui Li, Renda Zhao. Artificial Neural Network Prediction for Seismic Response of Bridge Structure. In: Artificial Intelligence and Computational Intelligence, 2009. AICT'09. International Conference on, vol. 2: :503–506IEEE; 2009.
- [36] Lagaros Nikos D, Papadrakakis Manolis. Neural network based prediction schemes of the non-linear seismic response of 3D buildings. *Advances in Engineering Software*. 2012;44(1):92–115.
- [37] Christiansen Niels H, Høgsberg Jan, Winther Ole. Artificial neural networks for nonlinear dynamic response simulation in mechanical systems. In: Nordic Seminar on Computational Mechanics,(24):77–80; 2011.
- [38] LeCun Yann, Bengio Yoshua, others . Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*. 1995;3361(10):1995.
- [39] Medsker Larry, Jain Lakhmi C. *Recurrent neural networks: design and applications*. CRC press; 1999.
- [40] Mandic Danilo P, Chambers Jonathon. *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc.; 2001.
- [41] Lawrence Steve, Giles C Lee, Tsoi Ah Chung, Back Andrew D. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*. 1997;8(1):98–113.
- [42] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems:1097–1105; 2012.
- [43] Sun Shan-Bin, He Yuan-Yuan, Zhou Si-Da, Yue Zhen-Jiang. A Data-Driven Response Virtual Sensor Technique with Partial Vibration Measurements Using Convolutional Neural Network. *Sensors*. 2017;17(12):2888.
- [44] Wu Rih-Teng, Jahanshahi Mohammad R. Deep Convolutional Neural Network for Structural Dynamic Response Estimation and System Identification. *Journal of Engineering Mechanics*. 2018;145(1):04018125.
- [45] Zhang Ruiyang, Chen Zhao, Zheng Jingwei, Sun Hao. Temporal Convolutional Neural Network for Data-driven Seismic Response Modeling and Serviceability Assessment of Buildings. *ASCE Journal of Structural Engineering*. 2019;:submitted.
- [46] Bengio Yoshua, Simard Patrice, Frasconi Paolo. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*. 1994;5(2):157–166.
- [47] Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, Sutskever Ilya, Salakhutdinov Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014;15(1):1929–1958.
- [48] Gal Yarin, Ghahramani Zoubin. A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems:1019–1027; 2016.
- [49] Huang Xiaohui, Ye Yunming, Xiong Liyan, Lau Raymond YK, Jiang Nan, Wang Shaokai. Time series k-means: A new k-means type smooth subspace clustering for time series data. *Information Sciences*. 2016;367:1–13.
- [50] Dong Baiping, Sause Richard, Ricles James M. Seismic Response and Damage of Reduced-Strength Steel MRF Structures with Nonlinear Viscous Dampers. *Journal of Structural Engineering*. 2018;144(12):04018221.
- [51] Chollet François, others . Keras <https://keras.io>2015.
- [52] Abadi Martín, Agarwal Ashish, Barham Paul, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org; 2015.
- [53] Sun Hao, Luş Hilmi, Betti Raimondo. Identification of structural models using a modified Artificial Bee Colony algorithm. *Computers & Structures*. 2013;116:59–74.
- [54] Wen Yi-Kwei. Method for random vibration of hysteretic systems. *Journal of the engineering mechanics division*. 1976;102(2):249–263.
- [55] Zhang Ruiyang, Phillips Brian M, Taniguchi Shun, Ikenaga Masahiro, Ikago Kohju. Shake table real-time hybrid simulation techniques for the performance evaluation of buildings with inter-story isolation. *Structural Control and Health Monitoring*. 2017;24(10):e1971.
- [56] Sato Tadanobu, Qi Kai. Adaptive H_{∞} filter: its application to structural identification. *Journal of Engineering Mechanics*. 1998;124(11):1233–1240.

- [57] Liang Jianwen, Chaudhuri Samit Ray, Shinozuka Masanobu. Simulation of nonstationary stochastic processes by spectral representation. *Journal of Engineering Mechanics*. 2007;133(6):616–627.
- [58] Vlachos Christos, Papakonstantinou Konstantinos G, Deodatis George. A multi-modal analytical non-stationary spectral model for characterization and stochastic simulation of earthquake ground motions. *Soil Dynamics and Earthquake Engineering*. 2016;80:177–191.
- [59] Glorot Xavier, Bengio Yoshua. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics:249–256; 2010.
- [60] Kingma Diederik P, Ba Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014;.
- [61] Pedregosa Fabian, Varoquaux Gaël, Gramfort Alexandre, et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*. 2011;12(Oct):2825–2830.
- [62] Meng Qi, Chen Wei, Wang Yue, Ma Zhi-Ming, Liu Tie-Yan. Convergence Analysis of Distributed Stochastic Gradient Descent with Shuffling. *arXiv preprint arXiv:1709.10432*. 2017;.
- [63] Haddadi Hamid, Shakal A, Stephens C, et al. Center for Engineering Strong-Motion Data (CESMD). In: Proceedings of the 14th World Conference on Earthquake Engineering; 2008.
- [64] Castaneda Aguilar Nestor E. Development and validation of a real-time computational framework for hybrid simulation of dynamically-excited steel frame structures. 2012;.
- [65] Castaneda Nestor, Gao Xiuyu, Dyke Shirley J. Computational tool for real-time hybrid simulation of seismically excited steel frame structures. *Journal of Computing in Civil Engineering*. 2013;29(3):04014049.
- [66] Chen Cheng, Ricles James M. Development of direct integration algorithms for structural dynamics using discrete control theory. *Journal of Engineering Mechanics*. 2008;134(8):676–683.
- [67] Chiou Brian, Darragh Robert, Gregor Nick, Silva Walter. NGA project strong-motion database. *Earthquake Spectra*. 2008;24(1):23–44.