Innovative Sensing by Using Deep Learning Framework

Nur Sila Gulgec * Martin Takáč † Shamim N. Pakzad ‡

Abstract

Structures experience large vibrations and stress variations during their life cycles. This causes reduction in the load-carrying capacity which is the main design criteria for many structures. Therefore, it is important to accurately establish the performance of structures after construction that often needs full-field strain or stress measurements. Many traditional inspection methods collect strain measurements by using wired strain gauges. These strain gauges carry a high installation cost and have high power demand. In contrast, this paper introduces a new methodology to replace this high cost with utilizing inexpensive data coming from wireless sensor networks. The study proposes to collect acceleration responses coming from a structure and give them as an input to deep learning framework to estimate the stress or strain responses. The obtained stress or strain time series then can be used in many applications to better understand the conditions of the structures. In this paper, designed deep learning architecture consists of multi-layer neural networks and Long Short-Term Memory (LSTM). The network achieves to learn the relationship between input and output by exploiting the temporal dependencies of them. In the evaluation of the method, a three-story steel building is simulated by using various dynamic wind and earthquake loading scenarios. The acceleration time histories under these loading cases are utilized to predict the stress time series. The learned architecture is tested on an acceleration time series that the structure has never experienced.

Keywords Structural health monitoring • Long Short-Term Memory • Recurrent Neural Networks • Deep Neural Network

1 Introduction

As sensor networks today provide the opportunity to collect an enormous amount of data from any structure, Structural Health Monitoring (SHM) applications start posing a BIGDATA problem [1, 2]. Deep Neural Networks (deep learning or DNN) is an ideal state-of-the-art set of techniques for exploiting the opportunities hidden in BIGDATA [3]. Deep learning algorithms are designed such that they can learn from data. Therefore, deep learning is ideally suited to use large representative training datasets to learn complex features. During this learning process, they build a model which is then used to make data-driven predictions or decisions.

Many traditional SHM and condition assessment methods need full-field strain or stress measurements to be used in remaining fatigue life estimation, assessment of loading conditions, corrosion detection, composite material testing and structural design check. Large scale deployment of wired strain gauges, however, poses a fundamental limitation: they are expensive and laboriously impractical as more spatial information is desired [4]. An important, but relatively new method to measure the strain field is that of indirect monitoring [5, 6, 7]. Indirect sensing approaches first and foremost eliminate the installation costs associated with wiring and also provide a robust way to access critical locations in structures. One of the popular indirect methods for measuring kinematic quantities (displacement, strain) is Digital Image Correlation (DIC) which resolves relative movement using a reference image [8, 9]. The drawback of DIC system is the high cost of the equipment and data storage for a long and continuous dynamic monitoring protocol.

Acceleration measurements are another form of data that SHM applications rely on. Acceleration data can be collected relatively inexpensively by the means of fixed tethered sensors, wireless sensor networks (WSN) or mobile sensing. WSNs have been utilized in a variety of applications that range from low duty-cycle, low-power environmental monitoring applications

*Ph.D. Candidate, Department of Civil and Environmental Engineering, Lehigh University, 117 ATLSS Drive, Imbt Labs, Bethlehem, PA 18015, USA.
†Assistant Professor, Department of Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Harold S. Mohler Laboratory, Bethlehem, PA 18015, USA.
‡Associate Professor, Department of Civil and Environmental Engineering, Lehigh University, 117 ATLSS Drive, Imbt Labs, Bethlehem, PA 18015, USA.

to high-fidelity applications with accurate measurements, high sampling rate, and lossless communication for monitoring of mechanical and structural systems [10]. The advantages of WSNs include the low cost of installation and equipment, as well as their robustness and quality of data.

Addressing these limitations beg for an innovative sensing strategy where data can be integrated from inexpensive data sources. This paper presents a deep learning based approach using inexpensive data sources to predict stress or strain information of structural systems. To achieve that, the study proposes a deep learning framework comprised of multi-layer networks and Long Short-Term Memory (LSTM). Exploiting the multi-layer networks, architecture maps the complex relation between input and output. Furthermore, it captures the temporal dependencies of sensor data by using LSTM which is the state-of-the-art technique for time series prediction [11], language translation [12], and speech recognition [13].

The rest of the paper is organized as follows. First, a background information on deep learning is provided in Section 2; then, the proposed methodology with data preparation and training steps are described in Section 3. In Section 4, main findings of this study are discussed. Conclusions and future work are presented in Section 5.

2 Background on Deep Learning

Multi-Layer Neural Networks. Multi-layer neural networks are a subfield of machine learning where they build a deep graph mapped from input data to target [14]. The graphs are organized such that they have multiple linear layers activated by nonlinear transformations (e.g.sigmoid, tanh and others) [15]. The multi-layer networks (fully connected layers or FC layers) are composed of neurons and weight parameters (w) where the value of each neuron (s_i) can be computed by a weighted sum of the values of its input nodes (s'_j) activated by the nonlinear function a:

$$s_i = a\big(\sum_j w_{ij} s'_j\big). \tag{1}$$

Recurrent Neural Networks. Recurrent neural networks (RNN) are a family of deep neural networks for dealing with the sequential data [16]. Unlike multi-layer neural networks, RNNs are able to map target data from the entire history of previous inputs. RNN models capture the dynamics of the sequences with the directed loops in them [17]. A typical RNN architecture is shown in Figure 1, which demonstrates a RNN being unrolled into a full network that has a chain-like structure. At time t, RNN receives the input $x^{(t)}$ and the hidden values from previous state $h^{(t-1)}$. In other words, current decisions are affected by the previous states. Given an input sequence $\{x^{(1)}, x^{(2)}, ..., x^{(T)}\}$, RNN updates the hidden node values $\{h^{(1)}, h^{(2)}, ..., h^{(T)}\}$ by the following equation:

$$h^{(t)} = a(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h), \tag{2}$$

where W^{hx} denotes the weight matrix between the input and hidden layer, W^{hh} and b_h denote the recurrent weight matrix and the bias vector in hidden layer, respectively. The output of the weighted sum is typically passed through a function a such as sigmoid, tanh, ReLU or others. Optionally, output sequence at time t can be found with the formulation $y^{(t)} = (W^{yh}h^{(t)} + b_y)$.

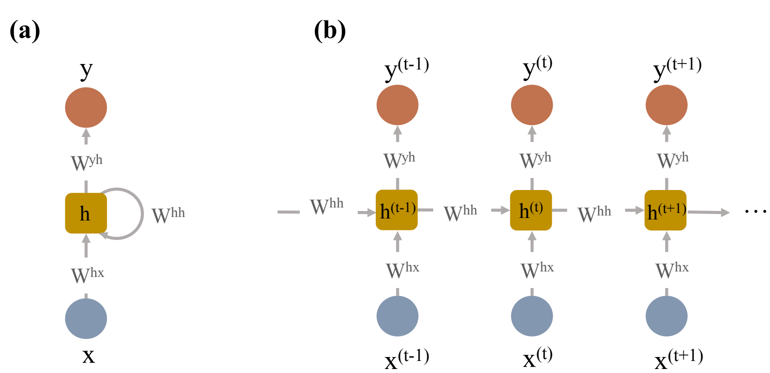

Figure 1: A recurrent neural network (a) rolled, (b) unrolled.

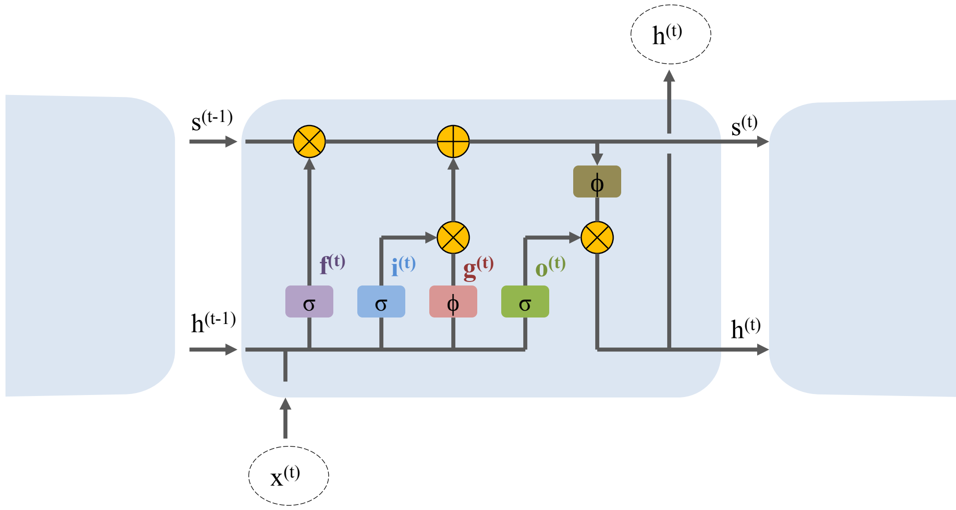

Figure 2: A LSTM memory cell.

Long Short-Term Memory (LSTM). In RNNs, the weight and bias parameters are shared along all the hidden layers [18]. Therefore, they may suffer from long-term time dependency problems where the gradients vanish or explode [19]. In order to address these problems, RNNs are improved over these years. Hochreiter and Schmidhuber [20] proposed long short-term memory (LSTM) which showed groundbreaking performance in time series prediction, language translation and video recognition.

LSTM networks are composed of memory cells which contain a chain of recurrent nodes. These memory cells help the network to control the information by adopting input gate, forget gate and output gate [21]. Diagram illustrated in Figure 2 focuses on single memory cell and shows data flow through it. Each memory cell includes input node, internal state, input, forget and output gates. *Input node*, (g), takes the current input and hidden layer at previous time step, then computes the weighted sum followed by tanh function (ϕ). *Input gate*, (i), controls which input to be passed to the memory cell. For instance, if the value after sigmoid function (σ) is 0, gate cuts off the input otherwise, it allows data to pass through. *Forget gate*, (f) proposed by Gers et al. (1999) [22] flushes the content if it is necessary. *Internal state*, (s), has a self-connected node that updates itself by forgetting or adding new information. *Output gate*, (o), controls what information to pass the next time step. Equations used in LSTM computations are given as follows:

$$g^{(t)} = \phi(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g), \tag{3a}$$
$$i^{(t)} = \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + i_g), \tag{3b}$$
$$f^{(t)} = \sigma(W^{fx}x^{(t)} + W^{fh}f^{(t-1)} + f_g), \tag{3c}$$
$$o^{(t)} = \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + o_g), \tag{3d}$$
$$s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)}, \tag{3e}$$
$$h^{(t)} = \phi(s^{(t)}) \odot o^{(t)}. \tag{3f}$$
$$\tag{3g}$$

3 Proposed Methodology

3.1 Data Preparation

Design of the deep learning architectures critically depends on the training dataset which should be constructed by well-known states [23]. In this study, a three-story steel building is simulated to be used in preparation of the training dataset. The structure is assumed to be six-bay by six-bay office building in seismic region. The primary system consists of eight identical special concentrically braced frames (SCBF) and the gravity load frames where the plan view of the building is shown in 3. The seismic area tributary to one SCBF is defined as one quarter of the total area by exploiting the symmetric layout of the building. The model and section dimensions are designed based on the similar experimental structure model described in Dong et al. (2016) [24]. However, damped braced frame (DBF) and moment resisting frame (DBF) are replaced by SCBF for simplicity. The section properties of the designed model is shown in Figure 3.

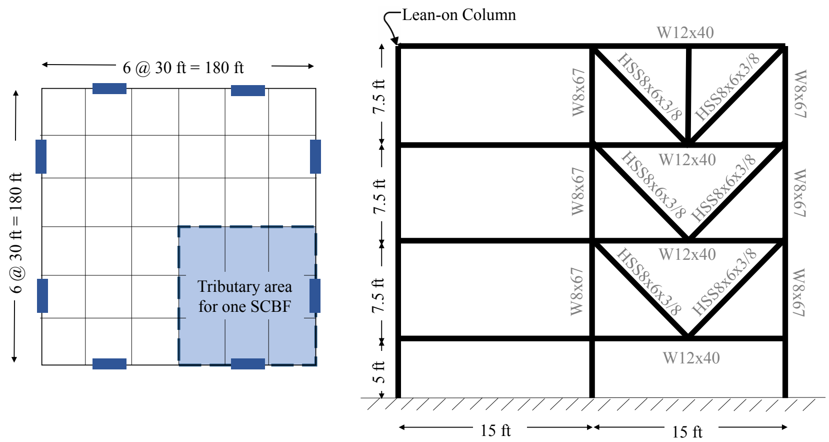

Figure 3: Structure model (a plane view of prototype building) and details of SCBF frames.

The building is loaded with different dynamic loading combinations under the effect of wind and earthquake. Twelve different ground motion records are selected from the PEER NGA online database [25] which have the smallest sum of the squared error (SSE) matching with uniform hazard spectrum (UHS). OpenSha is used to generate hazard spectrum [26].

Wind velocity fluctuations are performed by Monte Carlo Simulation [27]. One-dimensional, uni-variate sample functions are created to match the probabilistic characteristics of the wind load. Wind velocity is defined with the Kaimal's Spectrum [28] which can be formulated by the following equation:

$$S_{xx}(w) = \frac{200zu_*^2}{4\pi U(z)\left[1+\dfrac{50|w|z}{2\pi U(z)}\right]^{5/3}}, \tag{4}$$

where $U(z)$ is mean speed at height z, k is Von karman's constant, w is frequency in rad/s, u_* is shear velocity of the flow defined by $u_* = kU(z)/ln(z/z0)$. In this study, roughness length z_0 is adopted as 0.001266 m and mean wind speed height is taken as 8 m/s to simulate wind vibrations. By using the simulated velocities, wind pressure is calculated and dynamically applied to the structure.

The load combinations through 4a to 4d are utilized with 12 ground motions and 6 wind load scenarios. Dynamic responses are collected to have a total of 1000 points with a sampling time of $\Delta_t = 0.05s$. Total 40 different loading combinations are created. Acceleration $(\ddot{u}(t))$ and stress (σ) time histories are collected from nine locations as shown in Figure 4. The collected responses are distributed to training, validation and testing datasets. Responses are normalized with the overall maximum of sequences.

$$(1.2D + 0.2S_{DS})D + \rho Q_E + L, \tag{5a}$$
$$(0.9D + 0.2S_{DS})D + \rho Q_E, \tag{5b}$$
$$1.2D + 1.6L + 0.8W, \tag{5c}$$
$$1.2D + L + 1.6W, \tag{5d}$$

where D and L are dead and live loads calculated based on ASCE7010 design code [29], W is wind load, Q_E is the effect of horizontal seismic forces, S_{DS} is design, 5 percent damped, spectral response acceleration parameter at short periods which is taken as 1.0 g and ρ is a redundancy factor adopted as 1.

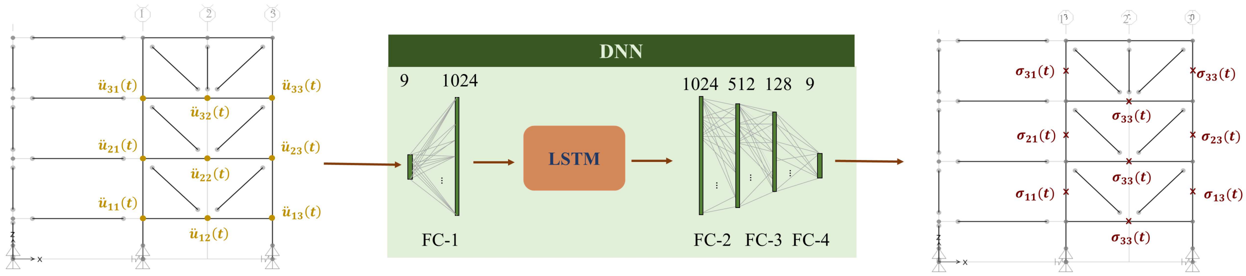

Figure 4: Proposed network topology.

3.2 Proposed Network Topology and Training

Let a series of observations $\{\ddot{u}_{11}(t), \ddot{u}_{12}(t), \ddot{u}_{12}(t), ..., \ddot{u}_{33}(t)\}$ denotes the acquired acceleration data from the building model. Here the acceleration of the joint of ij sampled at time step t is represented as $\ddot{u}_{ij}(t)$ where i is the number of story and j is the grid number in x direction (as shown in Figure 4). The responses obtained from each joint at the time steps $t = 0, 0.05, ..., 50$ for different loading combinations are prepared as tensors (e.g. in the shape of $[24, 1000, 9]$ for training dataset).

The proposed network topology takes the sequences of acceleration data and learns how to predict stress time histories. At time t, model takes input sequences and passes them through the multi-layer neural network size of $[1024]$. The output of the FC layer is used to feed the LSTM memory cells with size of 1024, which are then followed by three multi-layer network with the sizes of $[1024 - 512 - 128]$. These layers are activated by using tanh() function. The final output of the last DNN layer is used to predict the stress time histories $\{\sigma_{11}(t), \sigma_{12}(t), \sigma_{12}(t), ..., \sigma_{33}(t)\}$. The scheme of the proposed architecture can be found in Figure 4.

The loss function is defined as mean squared error of predictions and true values of acceleration sequences. The networks is trained by ADAM optimizer which is an adaptive learning rate algorithm [30] with batch sizes of $N = 24$. In order to favor the short-term dependencies of data, truncated backpropogation through time (BPTT) approach is adopted [31].

4 Results

This section presents the performance analysis of the proposed deep learning based methodology. The designed architecture is trained for 10,000 epochs until it overfits the training dataset. By overfitting the training dataset, the capability of proposed architecture is observed and predicted results are compared with the true stress time series. The mean squared error for training process is found to be 0.001. As an example, a performance of one earthquake sample sequence that is collected from Joint$_{31}$ is visualized in Figure 5. In figure, normalized acceleration time series are plotted with both normalized target and predicted stress time series. It is observable that predicted sequence perfectly captures the target sequence. Furthermore, figure shows that although acceleration and stress sequences do not have a relationship that easily noticeable, the designed architecture accurately estimates it by exploiting the LSTM cells.

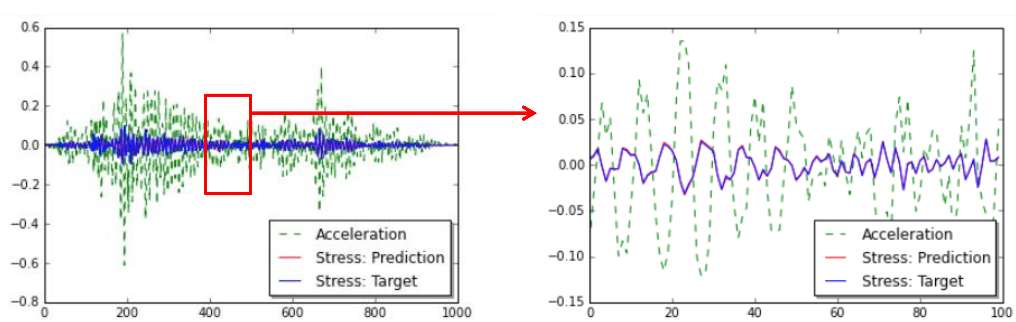

Figure 5: A performance of an example training sample.

After training of the network, the model is tested with the earthquake loading scenarios that structure has never been exposed. The mean square error is found to be 0.12. Similar plots are also generated for example earthquake loading cases. The earthquake loads for Joint_{31} and Joint_{33} are presented in Figure 6. These examples shows that even the model is overfitted to training process and unseen loading earthquake cases are used, the model predicts the target sequences almost perfectly. This performance shows that introduced approach is promising for described innovative sensing strategy. The small deviance in true and estimated sequences for Joint_{33} can be reduced by adopting multiple LSTMs in the model or applying a fine tuning.

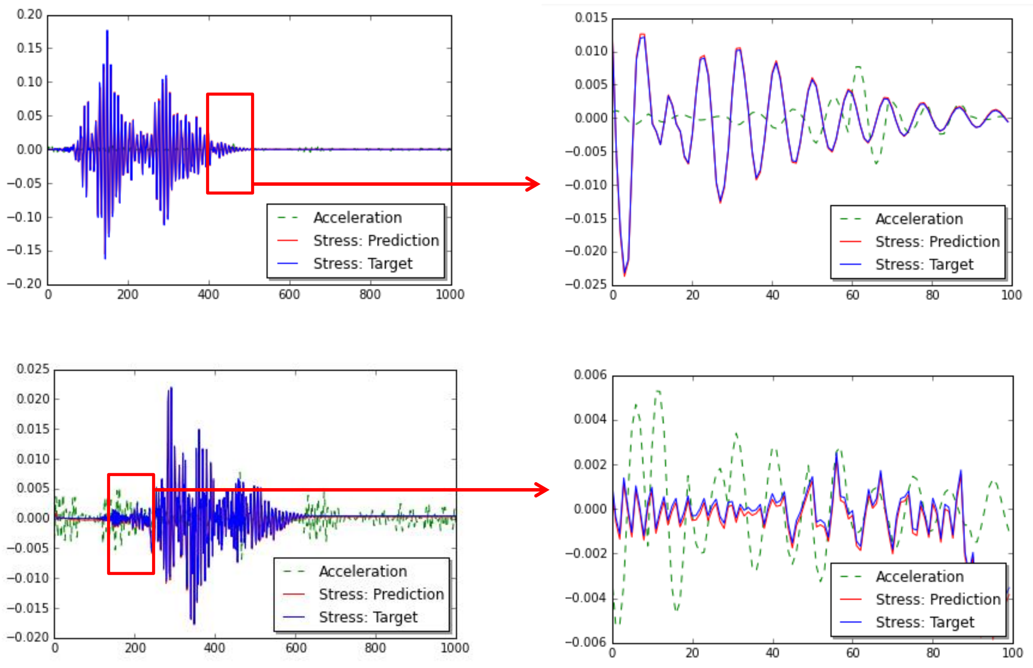

Figure 6: A performance of example testing samples.

5 Conclusion

This paper introduces a deep learning based platform so the data obtained by using wireless sensor networks can be used to obtain stress or strain information which is necessary for many potential applications including: damage diagnosis, remaining fatigue life estimation, accurate assessment of loading conditions, corrosion detection, composite material testing, and structural design check. The proposed network exploits the temporal modeling of LSTM and nonlinear mapping of FC layers to be able discover temporal dependencies and complex relationships between input and output sequences. Based on the findings of the approach, accurate estimation of stress time series is possible with acceleration acquired from inexpensive sensing system. Results show that stream of prediction values are matching quite well for training samples. The performance of the network needs a little improvement for the load scenarios that structure has never been experienced.

To discover more of abilities of deep neural networks, further research steps are important. The future work aims to extend this work by (i) designing more complex network architecture which considers both temporal and spatial dependencies of data, (ii) using experimental data, (iii) trying different type of data source conversions specific to other structures.

Acknowledgment. Research funding is partially provided by the National Science Foundation through Grant No. CMMI-1351537 by Hazard Mitigation and Structural Engineering program, and by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA). Martin Takáč was supported by National Science Foundation grant CCF-1618717 and CMMI-1663256.

References

[1] Thomas J Matarazzo, S Golnaz Shahidi, Minwoo Chang, and Shamim N Pakzad. Are today's shm procedures suitable for tomorrow's bigdata? In *Structural Health Monitoring and Damage Detection, Volume 7*, pages 59–65. Springer, 2015.

[2] Nur Sila Gulgec, S Golnaz Shahidi, and Shamim N Pakzad. A comparative study of compressive sensing approaches for a structural damage diagnosis. In *Geotechnical and Structural Engineering Congress 2016*, pages 1910–1919, 2016.

[3] Nur Sila Gulgec, Golnaz S Shahidi, Thomas J Matarazzo, and Shamim N Pakzad. Current challenges with bigdata analytics in structural health monitoring. In *Structural Health Monitoring & Damage Detection, Volume 7*, pages 79–84. Springer, 2017.

[4] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 254–263. ACM, 2007.

[5] Diego F Giraldo, Shirley J Dyke, and Juan M Caicedo. Damage detection accommodating varying environmental conditions. *Structural Health Monitoring*, 5(2):155–172, 2006.

[6] Arturo González, E Covián, and J Madera. Determination of bridge natural frequencies using a moving vehicle instrumented with accelerometers and gps. In *Proceedings of the Ninth International Conference on Computational Structures Technology, CST2008, Athens, Greece, 2-5 September, 2008*. Civil-Comp Press, 2008.

[7] G Lederman, Z Wang, J Bielak, H Noh, JH Garrett, S Chen, J Kovacevic, F Cerda, and P Rizzo. Damage quantification and localization algorithms for indirect shm of bridges. In *Proc. Int. Conf. Bridge Maint., Safety Manag., Shanghai, China*, 2014.

[8] Satoru Yoneyama, A Kitagawa, S Iwata, K Tani, and H Kikuta. Bridge deflection measurement using digital image correlation. *Experimental Techniques*, 31(1):34–40, 2007.

[9] Francois Hild and Stéphane Roux. *Digital image correlation*. Wiley-VCH, Weinheim, 2012.

[10] Shamim N Pakzad, Gregory L Fenves, Sukun Kim, and David E Culler. Design and implementation of scalable wireless sensor network for structural monitoring. *Journal of infrastructure systems*, 14(1):89–101, 2008.

[11] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.

[12] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[13] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772, 2014.

[14] Nur Sila Gulgec, Martin Takáč, and Shamim N Pakzad. Structural damage detection using convolutional neural networks. In *Model Validation and Uncertainty Quantification, Volume 3*, pages 331–337. Springer, 2017.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[16] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[17] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[18] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[19] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[21] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[22] Felix A GERS, Jürgen SCHMIDHUBER, and Fred CUMMINS. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[23] MF Elkordy, KC Chang, and GC Lee. Neural networks trained by analytically simulated damage states. *Journal of Computing in Civil Engineering*, 7(2):130–145, 1993.

[24] Baiping Dong, Richard Sause, and James M Ricles. Seismic response and performance of a steel mrf building with nonlinear viscous dampers under dbe and mce. *Journal of Structural Engineering*, 142(6):04016023, 2016.

[25] PEER NGA. Pacific engineering research database, 2008.

[26] Edward H Field, Thomas H Jordan, and C Allin Cornell. Opensha: A developing community-modeling environment for seismic hazard analysis. *Seismological Research Letters*, 74(4):406–419, 2003.

[27] Luigi Carassale and Giovanni Solari. Monte carlo simulation of wind velocity fields on complex structures. *Journal of Wind Engineering and Industrial Aerodynamics*, 94(5):323–339, 2006.

[28] JC Kaimal, JC Wyngaard, Y Izumi, and OR Cote. Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, 98(417):563–589, 1972.

[29] null null. *Minimum Design Loads for Buildings and Other Structures*. American Society of Civil Engineers, asce/sei 7-10 edition, 2013.

[30] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.