

STRUCTURAL DAMAGE DETECTION USING HYBRID DEEP LEARNING ALGORITHM

Dang Viet Hung^{a,*}, Ha Manh Hung^a, Pham Hoang Anh^a, Nguyen Truong Thang^a

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

Article history:

Received 04/02/2020, Revised 16/3/2020, Accepted 18/3/2020

Abstract

Timely monitoring the large-scale civil structure is a tedious task demanding expert experience and significant economic resources. Towards a smart monitoring system, this study proposes a hybrid deep learning algorithm aiming for structural damage detection tasks, which not only reduces required resources, including computational complexity, data storage but also has the capability to deal with different damage levels. The technique combines the ability to capture local connectivity of Convolution Neural Network and the well-known performance in accounting for long-term dependencies of Long-Short Term Memory network, into a single end-to-end architecture using directly raw acceleration time-series without requiring any signal preprocessing step. The proposed approach is applied to a series of experimentally measured vibration data from a three-story frame and successful in providing accurate damage identification results. Furthermore, parametric studies are carried out to demonstrate the robustness of this hybrid deep learning method when facing data corrupted by random noises, which is unavoidable in reality.

Keywords: structural damage detection; deep learning algorithm; vibration; sensor; signal processing.

[https://doi.org/10.31814/stce.nuce2020-14\(2\)-05](https://doi.org/10.31814/stce.nuce2020-14(2)-05) © 2020 National University of Civil Engineering

1. Introduction

Large-scale civil infrastructures play a critical role in society by facilitating transportation, supporting economic growth, and improving the quality of daily life. Thereby, it is of great importance for ensuring their smooth operations despite various external excitations such as wind loads, vehicular loads, accidental loads, environmental changes, blast loads, fire, earthquakes. To this end, effective and efficient continuous monitoring systems are indispensable. Recently, applying Deep Learning (DL) algorithms to the analysis of the structure's behavior [1, 2] and monitoring the operational condition of infrastructure is an exciting research direction in the engineering community owing to their capacity in dealing with a large amount of measurement data and the rapid development of technology such as high-performance computers and new sensors devices, i.e., wireless sensors, Internet of Thing sensors, etc. The data fed into DL algorithms are collected from a system of sensors embedded across structures. Different types of sensors are helpful, but the measured vibration data are currently the most common.

Formally, using vibration data to detect potential deterioration in structural components is termed Vibration-based Structural health monitoring (VSHM) [3]. Classical methods for VSHM usually require a modal analysis step to extract modal characteristics of the structure such as natural frequencies,

*Corresponding author. E-mail address: hungdv@nuce.edu.vn (Hung, D. V.)

and mode shapes. The deviation between experimentally extracted values with those of intact state is determined then being fed into an optimization method to detect any structural damages. However, for large-scale infrastructure, the modal identification step is challenging because of a vast number of required degrees of freedom and inevitable environmental noise. Besides, low-frequency modal characteristics are insensitive to local damages, while high-frequency ones are arduous to determine. Thus, DL is a promising alternative method because it allows for direct identification of damage from raw sensory data.

Recently, Abdeljaber et al. [4] proposed a one dimensional convolution neural network (1DCNN) to detect changes in structural properties of a steel frame using measured acceleration signals. Li et al. [5] published promising results for structural damage detection of Euler-Bernoulli beams by combining 1DCNN and original waveform signals in lieu of handcrafted features. Avci et al. [6] addressed the loss of connection stiffness of a steel frame structure via a novel structural health monitoring (SHM) method using 1DCNN and wireless sensors networks. Zhang et al. [7] developed a 1DCNN method for VSHM of bridge structures and successfully tested on both a simplified laboratory model and a real steel bridge. Ince [8] demonstrated that the 1DCNN architecture was highly effective in real-time monitoring motor conditions because their model took only 1.0 ms per classification, and the experimental accuracy result was more than 97%. To address the fault diagnosis problem of the wind turbine gearbox, Jiang et al. [9] proposed a 1DCNN-based method with the ability to learn relevant features at multiple time scales in a parallel fashion. Jing et al. [10] showed that the 1DCNN outperformed the popular machine learning methods such as support vector machine, random forest, which utilized classical manual feature extraction in detecting faults of gearboxes.

On the other aspect, the recurrent neural network (RNN) is a special architecture among DL algorithms designed for capturing time-dependent characteristics; thus, RNNs are naturally proposed for feature learning of sensor measurements. However, the sensor data usually consist of long sequential samples; therefore, the vanilla RNN suffers either the gradient exploding or vanishing. To cope with this long-range dependencies, some derived architectures from RNN are developed by scientists such as Long Short Term Memory (LSTM) and its simplified version Gated Recurrent Unit. Zhao et al. [11] developed two LSTM-based methods for structural health monitoring of high-speed CNC machines using sensory data, namely basic LSTMs, and Deep LSTMs. Their results confirmed that the LSTM network could perform better than a number of baseline methods. Yuan et al. [12] investigated the remaining useful life of aero-engine utilizing LSTM under various operation modes and several degradation scenarios. They found that the standard version of LSTM itself has a strong ability to achieve accurate both long term and short term prediction during the degradation process. Lei et al. [13] developed a LSTM-based method for fault diagnosis of wind turbines based on multiple-sensor time-series signals. In their study, LSTM achieved the best performance among deep learning architectures, including the vanilla RNN, the MLP, and the Deep Convolution Neural Network. Qiu et al. [14] addressed the bearing faults diagnosis problem by designing a modified bidirectional LSTM, which could reduce error rates by six times compared to conventional methods.

However, when the length of the time-series becomes larger, the time complexity of the LSTM will intractably increase compared to other counterparts, which hinders the application of LSTM to long-term structural health monitoring. To overcome this drawback, ones propose a hybrid architecture combining the efficiency of 1DCNN in capturing local connectivity with the well-known performance in recognizing long-term dependencies of LSTM network into a single end-to-end architecture. The main contributions of the work are summarized as below:

- This work proposes a hybrid deep learning algorithm for low complexity analysis of structural

damage detection.

- With the use of the proposed approach, relatively high accuracy is achieved for damage identification tasks, including minor damage level which is difficult to visually identify.
- A parametric study is conducted to demonstrate that the present method is robust in handling data corrupted by random environmental noise in practice.

The remainder of this paper is organized as follows: Section 2 introduces in details the components of the architecture of the hybrid Deep Learning algorithm; Section 3 describes the experimental data set and data augmentation techniques; Section 4 presents damage identification results obtained by the mean of the proposed method. Finally, Section 5 draws the conclusion and gives some ideas for future work.

2. Hybrid Deep learning model CNN-LSTM

It is commonly acknowledged that the convolution neural networks (CNNs) can provide outstanding performance on signal classification and pattern recognition because of two folds. On the one hand, its architecture is especially suitable for discovering local relationships in space; on the other hand, it reduces the number of network parameters, thus leading to a lower computational complexity compared to conventional Deep Learning architectures. The hyperparameters of a 1D convolution layer comprise the number of kernels, the kernel length, and the stride value. The formula of one typical convolutional layer is expressed as follows [15]:

$$h_k = \text{conv1D}(w_k, X) + b_k \quad (1)$$

where h_k , w_k and b_k are respectively the output vector, weight vector and bias parameter of the kernel k , X is the input vector and conv1D is the 1D convolution operator whose i^{th} output is calculated by the following formula:

$$\text{conv1D}(w_k, X(i)) = w_k \otimes X(i) = \sum_{j=1}^{N_k} w_{kj} x_{i-j}^{\alpha} \quad (2)$$

where N_k is the length of the kernel k , w_k is the j^{th} element of vector w_k .

On the other aspect, LSTM is a special type of deep neural network, using signal information at multiple previous time steps to perceive insight into the recent time step, referred to as “long-term dependencies”. The fundamental theory of the LSTM can be found in the work of Hochreiter and Schmidhuber [16]. The structure of LSTMs consist of repeating cells jointly connected, each cell has three gates, namely forget gate, input gate, and output gate to control information flow. The output of the LSTM sequences is fed into a fully connected layer with softmax activation function, which further provides the probability for each predicted class.

The mathematical formulas of this model are described as follows. A linear transformation of the combination of input x_t at time step t and output of hidden layer h_{t-1} at time step $t-1$, is expressed by:

$$L(h_{t-1}, x_t) = W[h_{t-1}, x_t] + b \quad (3)$$

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

Formulas of three gates inside each cell of LSTM are written by Olah [17]:

$$\begin{aligned} f_f &= \sigma(L_f(h_{t-1}, x_t)) \\ f_i &= \sigma(L_i(h_{t-1}, x_t)) \\ f_o &= \sigma(L_o(h_{t-1}, x_t)) \end{aligned} \quad (4)$$

The new candidate of information created at time step t is calculated by applying the tanh activation function on a linear transformation of a concatenation $[h_{t-1}; x_t]$:

$$C_t = \tanh(L_c(h_{t-1}, x_t)) \quad (5)$$

Then the flow of information is updated with the new candidate by element-wise operations:

$$s_t = f_f \oplus (f_i \odot C_t) \quad (6)$$

and the output of the cell at time step t is calculated based on the updated information and the output gate:

$$h_t = f_o \odot s_t \quad (7)$$

In summary, the function computing hidden outputs can be expressed as:

$$h_t = F(x_t, h_{t-1}) \quad (8)$$

In these equations, σ is the sigmoid function, $\tanh$ denotes the hyperbolic tangent functions, $\odot$ and $\oplus$ stand for component-wise multiplication and addition of two vectors, respectively.

In terms of data processing steps, we need to reshape data into the three-dimensional format accepted by the LSTM. The first dimension is the number of measured cases, which can be up to ten thousands. The second dimension is the number of time steps fed into each LSTM cell, which is of an order of hundreds, and the last dimension is the total number of sensors utilized for a specific structure. In fact, the number of time steps is a hyperparameter, being fine-tuned further to improve the performance of the model.

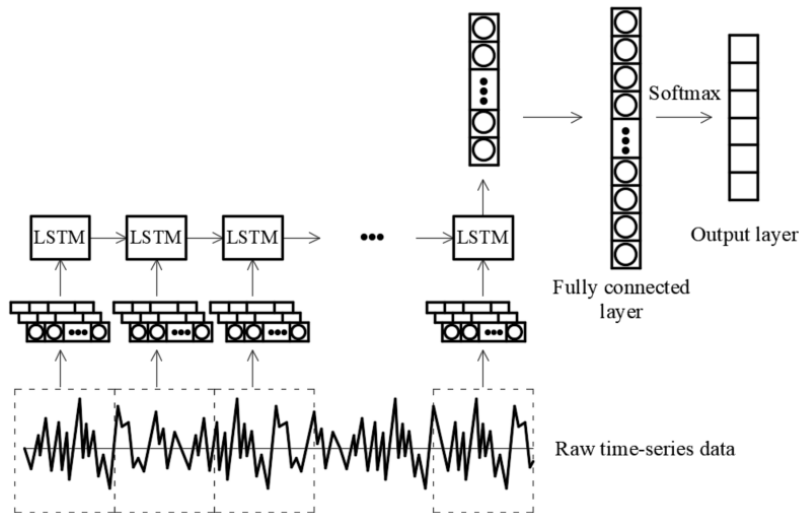


Figure 1. Architecture of the hybrid 1DCNN-LSTM architecture

Having established the convolutional layer and LSTM's memory cell, the hybrid deep learning architecture is schematically illustrated in Fig. 1, whose workflows are described as follows. Once vibration data enter into the network, it is divided into fixed-length segments, then the 1DCNN layer will extract inner relationships between measured points and their higher derivatives before feeding to the memory cell of LSTM where long-term dependencies are identified and retained over time.

The output of the last time instant will be converted into a one dimensional vector, then fed to a fully connected layer where the features are elaborated one more time before being passed to the output layer with the softmax activation function to provide damage identification results.

In this hybrid DL architecture, the essential hyperparameters which need to be determined further are the number of kernels k , the kernel length, the stride value in the convolution layer, and the number of hidden layers in LSTM cell.

3. Structural Health Monitoring Dataset

3.1. Description of laboratory data

In this section, the proposed hybrid deep learning structure is validated through a case study case involving experimentally measured vibration data from a three-story frame structure realized at Los Alamos National Laboratory [18], as shown in Fig. 2. The dataset is selected because of its resemblance to real scenarios, its appropriate number of time series, as well as its validity. The frame consists of columns with 17.7 cm length and $2.5 \times 0.6 \text{ cm}^2$ cross-section, and plates with 2.5 cm thickness and $30.5 \times 30.5 \text{ cm}^2$ area. These structural components are made from aluminum and joined together using bolts. An electrodynamic shaker at the base floor serves to excite the structure randomly, the excitation is band-limited in the range of 20-150 Hz. At the top floor and the third floor, an additional column ($15.0 \times 2.5 \times 2.5 \text{ cm}$) and a bumper are installed, respectively. The contact between these two elements when the frame vibrates will induce non-linearity into the dynamic behavior of the frame. Each floor of the structure is equipped with an accelerometer of 1000 mV/g nominal sensitivity to measure the structure vibration. An acceleration signal is recorded for 25.6 s with a sampling frequency of 320 Hz, resulting in a time-series of 8192 data points. As the maximum excitation frequency is 150 Hz, such sampling frequency is large enough to capture essential information content in the structure response. Fig. 1 shows the setup of the experiment.

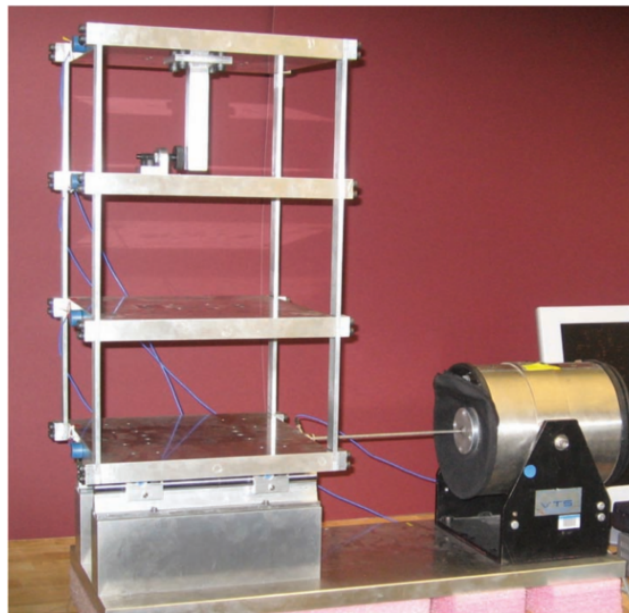


Figure 2. Three-story frame structure experiment [18]

The above default configuration of the structure is considered as the baseline condition. Afterward, a number of modifications are introduced to the structure to generate different structural state conditions. The modifications involve reducing 12.5% stiffness of one or two columns at each story, adding 19% extra floor's mass at the base or the 1st floor, and inducing contact between the suspended column at the top floor with the bumper. As a change in mass or column stiffness does not impose non-linearity in structure's responses, associated structural states numbered from 1 to 9, can be classified as undamaged states. Otherwise, the intermittent contact between the column and the bumper leads to sudden changes in the structure's responses. Therefore corresponding states number from 10 to 17 are treated as damaged conditions. It is noteworthy that by varying frequency of contact between these two elements through their initial distance, one could generate different levels of damage in the structure (minor, medium, or major). Table 1 lists all 17 structural states with detailed descriptions. Each state is measured ten times so that there are in total 170 time series for each accelerometer. Fig. 3 illustrates examples of time-series data measured from the top floor for all 17 structural states. As observed, it is difficult to distinguish damaged structural condition with undamaged ones visually. As such, the proposed hybrid deep learning is used to perform structural damage detection later.

Table 1. Structural state conditions in the three-story frame structure experiment

State	1	2	3	4	5	6
Condition	0	0	0	0	0	0
Description	Baseline	Added mass	Added mass	Column stiffness reduction	Column stiffness reduction	Column stiffness reduction
State	7	8	9	10	11	12
Condition	0	0	0	1 (minor)	1 (medium)	1 (medium)
Description	Column stiffness reduction	Column stiffness reduction	Column stiffness reduction	0.2 mm gap	0.15 mm gap	0.13 mm gap
State	13	14	15	16	17	
Condition	1 (medium)	1 (major)	1 (minor)	1 (minor)	1 (minor)	
Description	0.10 mm gap	0.05 mm gap	0.2 mm gap, added mass	0.2 mm gap, added mass	0.1 mm gap, added mass	

* 0: undamaged condition, 1: damaged condition (major, medium, minor).

3.2. Data augmentation

In this section, the process of generating data for the development of the hybrid deep learning is presented. The vibration of the whole structure is measured at each floor, but the floor close to the non-linear source, i.e., the suspended column and bumper, will be most influenced, thereby, time-series from the top floor will be utilized to generate the required data set. In general, a large and well-balanced database benefit the performance of Deep Learning algorithm, therefore data augmentation techniques are adopted to increase the size of the experimental data. In principle, the data augmentation technique introduces some minor changes in the original data without altering its underlying

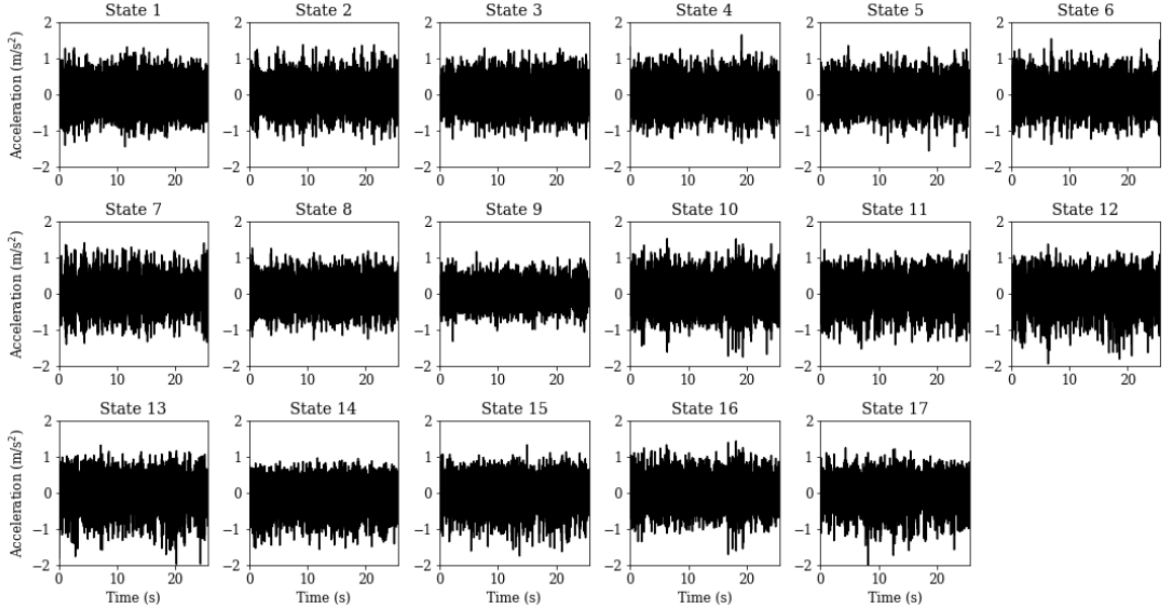


Figure 3. Acceleration-time series measured from top floor for all 17 structural states

pattern. Herein the utilized techniques are flipping (rotation), scaling, and permuting [19]. Flipping inverts the sign of the signal, scaling increases/decreases the magnitude of the raw data slightly by a random ratio from 5 to 10%, and permuting will swap two randomly selected small fractions (2% length) of the signal. Fig. 4 illustrates how data augmentation techniques work. After applying data augmentation techniques, the size of the final database increases up to 1000 time series, which is sufficient for training and validation of the proposed hybrid deep learning model.

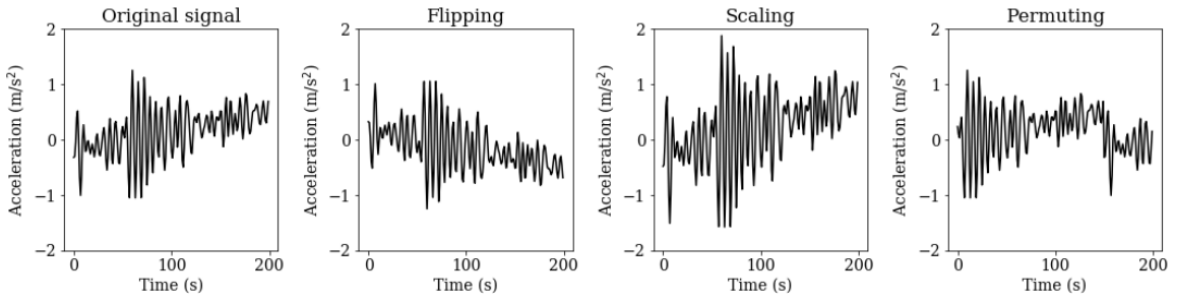


Figure 4. Data augmentation techniques for time-series data

3.3. Data preparation

After applying the data augmentation technique, the obtained database is used to train and evaluate the performance of the hybrid deep learning algorithm. Traditionally, the database is divided into three subsets, namely, training, validation, and testing one with a predefined ratio. However, a single split might not ensure a well-balanced distribution of different structural conditions among sub-dataset. Therefore, the K -fold cross-validation strategy is employed to reduce the bias in the final model. First, the data is broken down into the training and testing subset with a ratio of 90 : 10. Then, the

training dataset is split further into the K equal portions. Here a common value $K = 10$ is selected, meaning the training process will be iterated ten times, each time one different portion is used for validation, whereas the remaining serves for training. The K cross-validation strategy is graphically shown in Fig. 5.

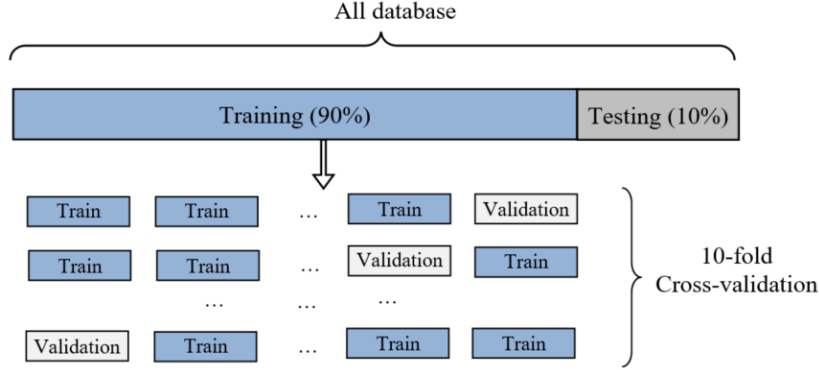


Figure 5. K -Fold cross-validation strategy

4. Computation results

4.1. Training process

In this part, the proposed method is applied to the above acceleration database to determine the structural condition of the frame, i.e., damaged/undamaged. As previously mentioned in Section 3, the hyperparameters of the proposed hybrid architecture are the number of kernels k , the kernel length L_k , in the convolution layer, and the number of hidden layers N_h in LSTM cell. Specifically, k varies in the range $[5, 50]$, L_k in $[10, 100]$, and N_h in $[3, 30]$. Such ranges are predetermined based on the size of the database (1000), the length of one time series (8192), and the number of output classes (2).

Table 2. Training and validation accuracy obtained for 10-fold cross-validation

Fold	1	2	3	4	5	6	7	8	9	10	Mean	Std
Train_Acc(%)	98.6	99.0	99.8	98.8	99.2	97.1	99.0	99.8	98.0	97.8	98.7	0.8
Valid_Acc(%)	84.5	93.1	79.3	89.6	82.7	82.7	84.2	87.7	89.4	82.4	85.5	4.0

Let take an example with $k = 100$, $L_k = 100$, and $N_h = 10$, Fig. 6 shows the evolution of training loss and validation accuracy versus the number of epochs. As observed, the training loss curve in blue is decreased steadily, and the accuracy curve in red increases at the same time, for the first 50 epoch. After that, the accuracy improves gradually, before reach to a convergent value of 84.2% around the 160th epoch. Apparently, there is a drop in the final validation accuracy compared to the highest peak around 90%. It can be explained that the model is trapped by a local optimization area, not the desirable global optimized solution. To overcome this problem, one could feed more data for the model, but this solution is not always available. Another solution is using the proposed 10-fold cross-validation strategy, which repeats the training process ten times with different training/validation pair, thus could avoid locally optimized solutions. Afterward, the mean and standard deviation of results are estimated (see Table 2).

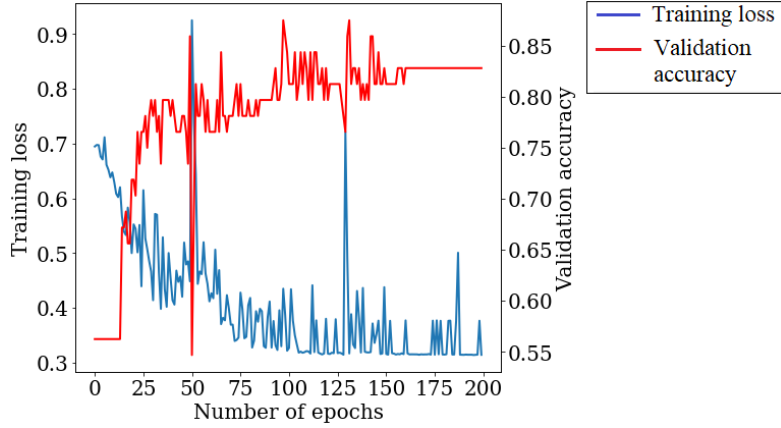


Figure 6. Evolution of validation accuracy and training loss during training process

Actually, there does not exist a common way for the selection of the best parameters, but it depends on specific problems. Thus, one adopts the Grid search technique to test all possible combinations of hyperparameters for the identification of the optimal architecture. It is noted that other parameters are fixed throughout the whole training process. For example, the learning rate is set to $l_r = 0.0001$, the number of epochs $N_{epoch} = 200$, the batch size = 32, the optimizer is Adam [20]. These values are defined by a preliminary study by authors to ensure cover as many details as possible in behaviors of DL model during the training process. In fact, the performance of the final model can be improved further by fine-tuning these training process parameters, but this work mainly focuses on the hybrid deep learning architecture, then only hyper-parameters directly related to the later are investigated. Fig. 7 shows the results of the Grid search technique. It is recorded that the final hybrid deep learning

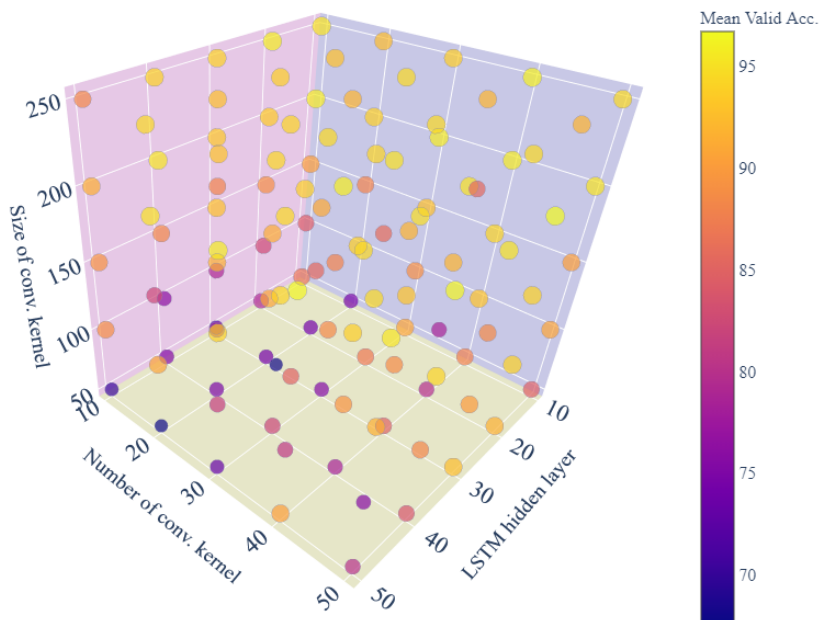


Figure 7. Validation accuracy obtained with different combination of hyperparameters

model with the number of convolutional kernels $k = 50$, the kernel length $L_k = 200$, and the number of LSTM cell's hidden layers $N_h = 20$ provides the highest averaged accuracy of 96.1% and a standard deviation of 1.2% on the validation dataset. When applied to the testing data, it yields an accuracy of 93.0%, unexceptionally. Fig. 8 presents the confusion matrix obtained from the testing data. Moreover, the inference time of the model for one testing time series is only 0.001 s, meaning suitable for a real-time structural health monitoring application. This result confirms the correctness of the proposed method in structural damage identification only using raw measured vibration data, no additional signal pre-processing is required.

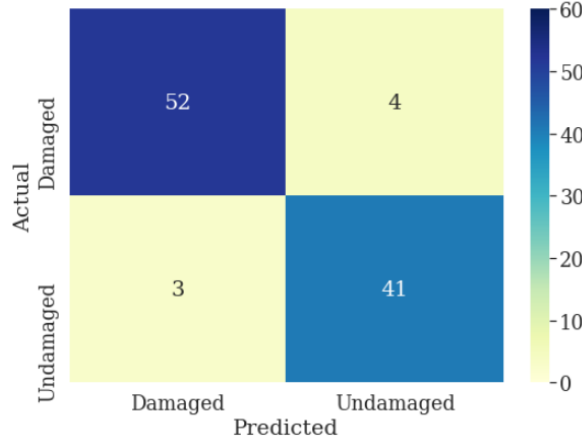


Figure 8. Confusion matrix of detection results on testing data

4.2. Noisy and missing data

In practice, measured vibrational data encompass inevitable noise caused by device instability, modeling assumption, or human errors. Therefore, it is of great importance to quantify the robustness of a SHM application in dealing with noisy data before applying to the real-world structure. For this purpose, one incorporates white-noise of different levels to the above testing data, then estimate the corresponding detection accuracy of the optimized hybrid DL model for these noisy data. The white-noise is defined by the following equation:

$$X_{noise}(t) = X(t) + \alpha \cdot \eta(t) \quad (9)$$

in which $X(t)$ and $X_{noise}(t)$ are original and added-noise time series, respectively, $\eta(t)$ is the white noise time series with zero mean and unit variance, α is the noise amplitudes based on the root mean squared value of $X(t)$. The value of α varies in the range of 0%–20%. Because the noise is random, ten run are carried out for each noise level, then mean and standard deviation values are calculated.

Fig. 9 depicts how the damage detection accuracy evolves with respect to the noise amplitude. It can be seen that with low level of noise, the proposed method still provides highly accurate results, i.e. more than 90% with a noise level up to 6%. The accuracies remain reasonable with noise level up to 12%, i.e., an accuracy of around 85%. However, when the noise became excessively high (from 15%), the model performance degrades sharply below 80%. At the same time, the variance of the prediction results increases with the increasing noise level. These results confirm that the proposed hybrid DL algorithm can be applied for real data contaminated by environmental noise whose amplitude is fairly small compared to that of vibration data.

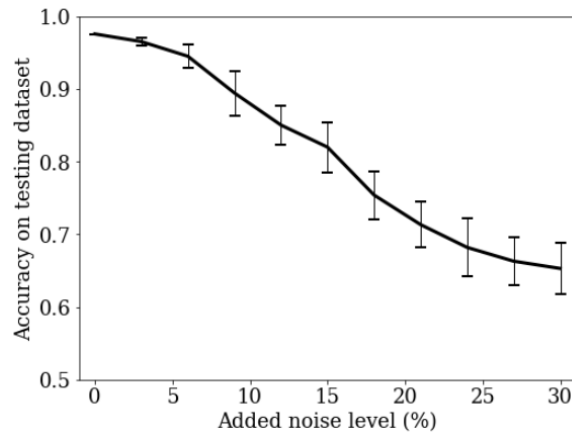


Figure 9. Evolution of prediction accuracy on testing dataset versus added-noise level

5. Conclusions

In this study, one developed a hybrid deep learning framework leveraging the superiority in capturing local connectivity of 1DCNN and the capability in retaining long-term dependency of LSTM to structural damage detection. The present framework is able to deliver highly accurate damage detection results directly using raw time-series data without any additional signal-preprocessing. The efficacy and efficiency of the proposed method are proved through a case study with experimental vibrational data obtained from a three-story frame. Moreover, a parametric study is carried out to provide insight into the influence of random noise on output results. It is desirable that a method applicable in reality needs to keep their performance at the acceptable level when the measured data is lightly lost or contaminated by noise. As expected, the robustness of the hybrid architecture is clearly shown by the parametric study results.

For the next step of this study, the proposed approach could be extended to allow much larger and longer input data because the real-world structures usually contain many sensors across their physical bodies. Moreover, other kinds of monitoring data should be incorporated into datasets, such as strain, rotation measurement, camera, and computer-vision based data for the purpose of detecting different types of structural anomalies such as concrete crack, steel corrosion, and fatigues.

References

- [1] Hung, T. V., Viet, V. Q., Thuat, D. V. (2019). [A deep learning-based procedure for estimation of ultimate load carrying of steel trusses using advanced analysis](#). *Journal of Science and Technology in Civil Engineering (STCE)-NUCE*, 13(3):113–123.
- [2] Nguyen, T. T., Dinh, K. (2019). [Prediction of bridge deck condition rating based on artificial neural networks](#). *Journal of Science and Technology in Civil Engineering (STCE)-NUCE*, 13(3):15–25.
- [3] Reynders, E., De Roeck, G. (2014). [Vibration-based damage identification: The z24 bridge benchmark](#). *Encyclopedia of Earthquake Engineering*, 482:1–8.
- [4] Abdeljaber, O., Avci, O., Kiranyaz, M. S., Boashash, B., Sodano, H., Inman, D. J. (2018). [1-D CNNs for structural damage detection: verification on a structural health monitoring benchmark data](#). *Neurocomputing*, 275:1308–1317.
- [5] Li, J., Dackermann, U., Xu, Y.-L., Samali, B. (2011). [Damage identification in civil engineering structures utilizing PCA-compressed residual frequency response functions and neural network ensembles](#). *Structural Control and Health Monitoring*, 18(2):207–226.

- [6] Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Inman, D. J. (2018). [Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks](#). *Journal of Sound and Vibration*, 424:158–172.
- [7] Zhang, Y., Miyamori, Y., Mikami, S., Saito, T. (2019). [Vibration-based structural state identification by a 1-dimensional convolutional neural network](#). *Computer-Aided Civil and Infrastructure Engineering*, 34 (9):822–839.
- [8] Ince, T. (2019). [Real-time broken rotor bar fault detection and classification by shallow 1D convolutional neural networks](#). *Electrical Engineering*, 101(2):599–608.
- [9] Jiang, G., He, H., Yan, J., Xie, P. (2018). [Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox](#). *IEEE Transactions on Industrial Electronics*, 66(4):3196–3207.
- [10] Jing, L., Zhao, M., Li, P., Xu, X. (2017). [A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox](#). *Measurement*, 111:1–10.
- [11] Zhao, R., Wang, J., Yan, R., Mao, K. (2016). Machine health monitoring with LSTM networks. In *10th International Conference on Sensing Technology (ICST)*, IEEE, 1–6.
- [12] Yuan, M., Wu, Y., Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In *International Conference on Aircraft Utility Systems (AUS)*, IEEE, 135–140.
- [13] Lei, J., Liu, C., Jiang, D. (2019). [Fault diagnosis of wind turbine based on Long Short-term memory networks](#). *Renewable Energy*, 133:422–432.
- [14] Qiu, D., Liu, Z., Zhou, Y., Shi, J. (2019). Modified Bi-Directional LSTM Neural Networks for Rolling Bearing Fault Diagnosis. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 1–6.
- [15] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D. J. (2019). 1D convolutional neural networks and applications: A survey. *arXiv preprint arXiv*, 1905.03554.
- [16] Hochreiter, S., Schmidhuber, J. (1997). [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- [17] Yan, S. (2018). [Understanding LSTM networks](#).
- [18] Figueiredo, E., Park, G., Figueiras, J., Farrar, C., Worden, K. (2009). *Structural health monitoring algorithm comparisons using standard data sets*. Los Alamos National Lab, Los Alamos, United States.
- [19] Le Guennec, A., Malinowski, S., Tavenard, R. (2016). Data augmentation for time series classification using convolutional neural networks. In *Workshop on Advanced Analytics and Learning on Temporal Data*.
- [20] Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*, 1412.6980.