

## Deep learning-based sensor fault detection using S-Long Short Term Memory Networks

Lili Li<sup>1,2a</sup>, Gang Liu<sup>\*1,2</sup>, Liangliang Zhang<sup>1,2b</sup> and Qing Li<sup>3c</sup>

<sup>1</sup>School of Civil Engineering, Chongqing University, Chongqing 400045, China

<sup>2</sup>The Key Laboratory of New Technology for Construction of Cities in Mountain Area of the Ministry of Education, Chongqing University, Chongqing 400030, China

<sup>3</sup>College of Computer Science, Chongqing University, Chongqing 400045, China

(Received November 17, 2017, Revised February 8, 2018, Accepted February 17, 2018)

**Abstract.** A number of sensing techniques have been implemented for detecting defects in civil infrastructures instead of onsite human inspections in structural health monitoring. However, the issue of faults in sensors has not received much attention. This issue may lead to incorrect interpretation of data and false alarms. To overcome these challenges, this article presents a deep learning-based method with a new architecture of Stateful Long Short Term Memory Neural Networks (S-LSTM NN) for detecting sensor fault without going into details of the fault features. As LSTMs are capable of learning data features automatically, and the proposed method works without an accurate mathematical model. The detection of four types of sensor faults are studied in this paper. Non-stationary acceleration responses of a three-span continuous bridge when under operational conditions are studied. A deep network model is applied to the measured bridge data with estimation to detect the sensor fault. Another set of sensor output data is used to supervise the network parameters and backpropagation algorithm to fine tune the parameters to establish a deep self-coding network model. The response residuals between the true value and the predicted value of the deep S-LSTM network was statistically analyzed to determine the fault threshold of sensor. Experimental study with a cable-stayed bridge further indicated that the proposed method is robust in the detection of the sensor fault.

**Keywords:** structural health monitoring; sensor fault; Long Short-Term Memory Networks; fault threshold; stochastic gradient decline; deep learning

### 1. Introduction

Study with structural health monitoring (SHM) has gained much attention in recent years in the civil engineering industry with implementation of SHM system in many bridge structures. Performance of the SHM system depends on the quality of sensors and reliability of measurements. However, sensors may inevitably incur various types of faults during their service life. A faulty

---

\*Corresponding author, Professor, E-mail: [gliu@cqu.edu.cn](mailto:gliu@cqu.edu.cn)

<sup>a</sup> Ph.D. Student, Email: [lilili@cqu.edu.cn](mailto:lilili@cqu.edu.cn)

<sup>b</sup> Professor, Email: [mrlzhang@163.com](mailto:mrlzhang@163.com)

<sup>c</sup> Ph.D. Student, Email: [liqing@cqu.edu.cn](mailto:liqing@cqu.edu.cn)

sensor may provide incorrect information for management decision on the bridge. Therefore, it is necessary to detect sensor faults in the early stage of their development. This paper tends to provide an automated method for detecting the faulty sensor in the sensory network of a SHM system.

Detection of instruments are widely studied in the areas of automatic control and mechanical engineering (Mansouri *et al.* 2016). They can be divided into two categories: physical model-based and data-based methods (Gev *et al.* 2013). The data-based methods or the data-driven methods are more popular in recent years because they do not require a physical model of the system which is usually not available. Kalman filter-based approach was applied to the detection of robot fault via sensor output prediction with Kalman filter. The residual between the predicted and actual values is used as an index to determine the most probable fault (Goel *et al.* 2000). Yu (1997) combines the estimation of parameters with the observer method, via the fault and the observer construction for a rapid fault diagnosis. A model-based method was applied to locate the fault. The model-based method is good for sensors in a linear system. However, it is not accurate with non-linear systems (Li and Zhou 2004). This greatly limits the application and development of model-based method in large structural health monitoring systems.

Kerschen *et al.* (2004) presented a data-driven sensor validation approach for SHM systems with the principal-component analysis (PCA) to model the monitoring data, and they used the angle between the principal subspaces as the feature for sensor-fault detection. Isolation of the sensor fault was implemented by removing one sensor at one time, and the faulty sensor was the one removed with the minimum angle. Sharifi *et al.* (2010) proposed a PCA-based sensor fault diagnosis identification (FDI) approach in the residual subspace of the PCA technique rather than in the principal subspace, which is commonly used in monitoring the industrial processes, specifically for smart structures. The detectability of each sensor fault was analyzed using a PCA-based residual generative model, and the probability of fault in each sensor was determined by analyzing these residuals using a Bayesian probabilistic decision process. Smarsly and Law (2014) presented an autonomous and fully decentralized approach toward sensor FDI in a wireless SHM system using analytical redundancy, which is the inherent information in a multivariate redundant measurement system. Each sensor output in this method was predicted using the output of other sensors based on the back propagation neural network embedded in each of the wireless sensor nodes installed in the monitored structure.

The residuals between the real and predicted sensor output values were used to autonomously detect and isolate the bias and drift sensor faults in real time. The above PCA-based methods are not sensitive to small sensor faults, which are relatively difficult to detect from monitored data of healthy structure.

Neural networks, amongst the many data-based methods, have been extensively applied to sensor fault detection in the past. The methods based on deep learning are rarely used in the structural health monitoring. The deep learning methods based on Recurrent Neural Networks (RNNs), especially the Long Short-Term Memory Neural Networks (LSTM NN), have emerged in recent years as one popular architecture to handle sequential data with various applications, including traffic flow prediction, pollution risk prediction, image captioning, speech recognition, genomic analysis and natural language processing (Auli *et al.* 2013). Xiaolei Ma *et al.* (2015) used LSTM NN to predict traffic flow. Comparison with different topologies of dynamic neural networks as well as other prevailing parametric and nonparametric algorithms suggested that the LSTM NN can achieve the best prediction performance in terms of both accuracy and stability. Sak *et al.* (2016) adopted the LSTM NN for pollution risk prediction, but they only classified the

pollution risk ranking without conducting real-value predictions of air pollutant concentrations. Moreover, they made predictions separately for individual cities without considering the spatial correlations between monitoring stations (Hwa-Lung and Chih-Hsin 2010). LSTM NN incorporates representation learning and model training together, which requires no additional domain knowledge. Additionally, this architecture can be used to discover some unseen structures in the data to improve the generalization capability of the model.

LSTM NN was initially introduced by Hochreiter and Schmidhuber (1997) for efficient learning on tasks that requires sequential short-term memory storage in many time steps during a trial. The LSTM differs from standard recurrent neural networks primarily with memory blocks instead of neurons. The memory blocks are connected through layers. It is very important because the acceleration is not only related to its current position state but also to the past positions (Sheng 2016). Moreover, it is also crucial for the integration of the short-term and long-term characteristics.

This paper presents an improved LSTM NN sensor-fault detection method that uses the deep neural network theory to establish to improve the prediction model in the temporal, and the residuals between the real and predicted sensor output values were used to autonomously detect the constant, gain, bias, and bottom noise sensor faults. The method is based on a hypothesis that the structure is healthy, and only the sensor is fault. Simultaneously, this method is mainly for fault detection of a single sensor in a period of time

The rest of the paper is organized as following. The sensory network and models on different sensor faults are presented in Section 2. The improved S-LSTM NN method is described, and the detection of sensor fault with S-LSTM NN is conducted in Section 3. Numerical and experimental results are presented in Section 4 to validate the proposed method. Finally, concluding remarks are given in Section 5.

## 2. Mathematical models on sensor faults

A sensor usually comprises of several components, i.e. the sensing device, transducer, signal processor and communication circuit, despite the different sensing principles. The sensor may have faults in any of these parts, and it may malfunction when its output deviates from its characterizing performances (Clough and Penzien 2013). There are different types of faults with different sensors. Mathematically, the deviation of signal in faulty instruments may be described in terms of the constant, gain, bias, and bottom noise.

### (1) Constant

Constant fault refers to a case where the sensor gives a constant value with time along with noise regardless of the actual signal changes. The mathematics model can be given as

$$x_o(t) = c + \delta(t) \quad (1)$$

where  $x_o(t)$  describes the sensor reading over time  $t$ ,  $c$  is a constant and  $\delta(t)$  represents the white noise. Most of the constant fault occurs when the sensitive core insulation resistance of sensor drops or damaged. The sensor is therefore insensitive to the measurement, and the output remain constant with time.

### (2) Gain

A sensor is known as in gain fault if the actual value  $x(t)$  of the sensor is associated with an

excessive-variance. The model can be described as

$$x_o(t) = \beta x(t) + \delta(t) \quad (2)$$

where  $\beta$  is a coefficient indicating the gradient of the gain fault. The larger the gain coefficient, the greater the degradation of precision of the sensor. The sensor gain fault is often caused by unstable voltage supply or non-linearity of the sensor. Therefore, coefficient  $\beta$  may change several times compared with the actual values even in the same measurement.

(3) Bias

If the measured reading shifts by a constant value from the true value, it is defined as bias:

$$x_o(t) = x(t) + d + \delta(t) \quad (3)$$

where  $d$  is a constant. Sensor bias fault occurs when the sensor is in creep or the sensor base is loose.

(4) Bottom noise

When the sensor completely fails, the sensor reading consists of bottom noise only as

$$x_o(t) = \delta(t) \quad (4)$$

where  $\delta(t)$  may be a random signal with an unknown mean and covariance, but it is white noise in most cases.

The gain coefficient in the gain fault model is assumed to be a random number between 1.3 and 2.0, with  $\beta = 1.3 + 0.7 \times \eta$ , where  $\eta$  is a random number between null and unity. The bias fault of sensor is small and is difficult to identify. Parameter  $d$  of the bias fault model and parameter  $c$  of the constant fault model are assumed equal with a value equal to the mean of the signal excluding the maximum and minimum values. The noise in signal is assumed white with zero mean and unit variance. The signals from sensors with the above faults are shown in Fig. 1.

### 3. S-LSTMs in sensor fault detection

The Long Short-Term Memory (LSTM) network, is a recurrent neural network that is capable to handle very large architecture using Backpropagation through time and overcoming the problem of exploding or vanishing gradient. This paper will discuss on how to develop the LSTM networks in Python using the Keras deep learning library to address the demonstration of time-series prediction problem. The Python library Keras which uses Tensorflow backend and another Python library enabling the availability of Keras in Pycharm were used.

#### 3.1 S-LSTM architecture and training

LSTM NN is adopted in this study to predict acceleration based on the sensor data. A LSTM NN is composed of one input layer, one recurrent hidden layer and one output layer. The basic unit of the hidden layer is memory block (Ngai *et al.* 2011). The memory block contains memory cells with self-connections memorizing the temporal state, and a pair of adaptive, multiplicative gating units to control information flow in the block. Two additional gates, namely the input gate and output gate, control the input and output activations of the block. The core of memory cell is a

recurrently self-connected linear unit-Constant Error Carousel (CEC), and the activation of the CEC represents the cell state. Due to the presence of the CEC, the multiplicative gates can learn to open and close, and thus LSTM NN can solve the vanishing gradient problem by keeping the network error constant.

To prevent the internal cell values growing without bound when processing continual time series that are not previously segmented, a forget gate was added to the memory block. This treatment enables the memory blocks to reset by itself once the information flow is out of date, and replaces the CEC weight with the multiplicative forget gate activation. The above procedure can be visualized in Fig. 2.

The model input is denoted as  $\mathbf{x}_t = (x_1, x_2, \dots, x_l)$ , and the output sequence is denoted as  $\mathbf{y}_t = (y_1, y_2, \dots, y_l)$ , where  $t$  is the prediction period. In the context of acceleration prediction,  $\mathbf{x}$  can be considered as historical input data, and  $\mathbf{y}$  is the prediction data. The predicted time will be iteratively calculated by the following equations

$$\mathbf{I}_t = \sigma(W_{ix}\mathbf{x}_t + W_{im}\mathbf{m}_{t-1} + W_{ic}\mathbf{c}_{t-1} + b_i) \quad (5)$$

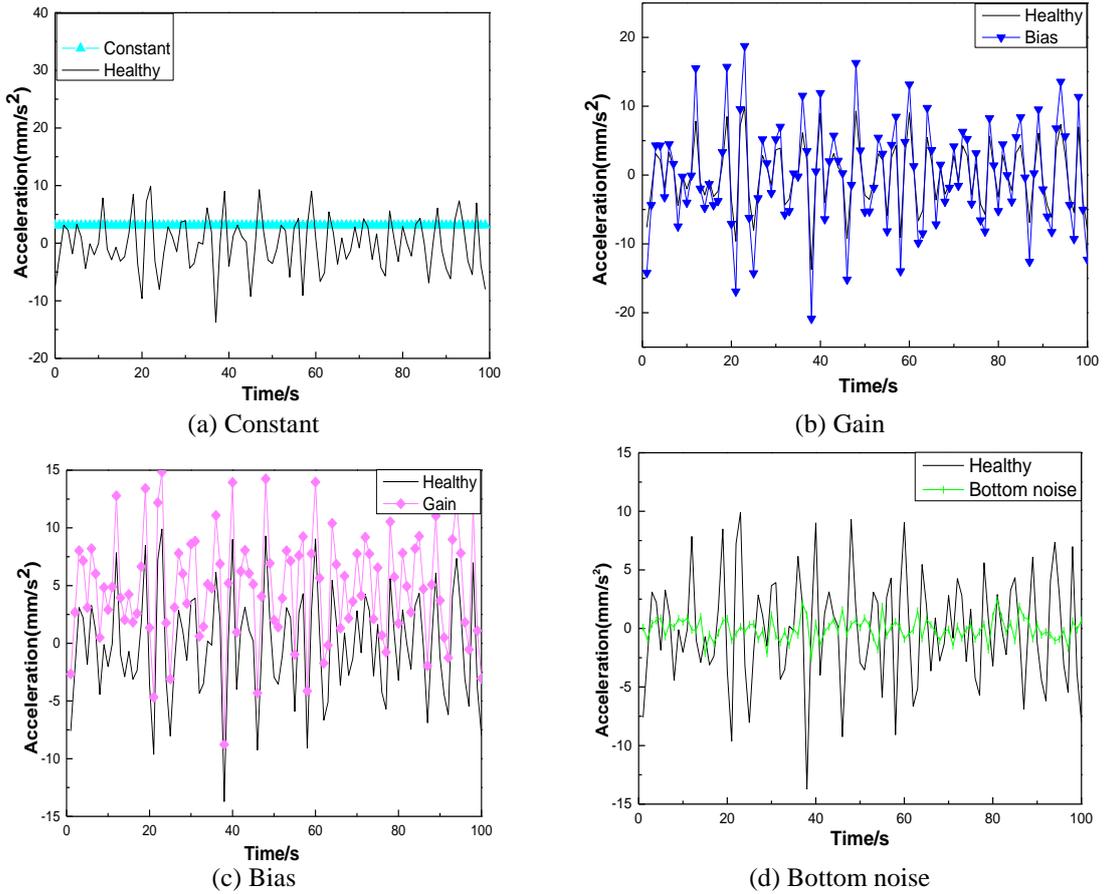


Fig. 1 Healthy signal and signals with sensor fault

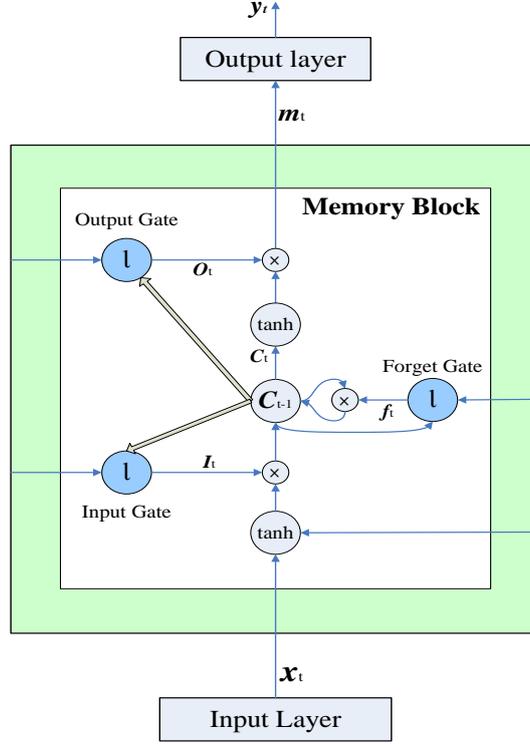


Fig. 2 Memory block architecture

$$\mathbf{f}_t = \sigma(W_{fx} \mathbf{x}_t + W_{fm} \mathbf{m}_{t-1} + W_{fc} \mathbf{c}_{t-1} + \mathbf{b}_f) \quad (6)$$

$$\mathbf{C}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \tanh(W_{cx} \mathbf{x}_t + W_{cm} \mathbf{m}_{t-1} + \mathbf{b}_c) \quad (7)$$

$$\mathbf{o}_t = \sigma(W_{ox} \mathbf{x}_t + W_{om} \mathbf{m}_{t-1} + W_{oc} \mathbf{c}_{t-1} + \mathbf{b}_o) \quad (8)$$

$$\mathbf{m}_t = \mathbf{o}_t \otimes \tanh(\mathbf{c}_t) \quad (9)$$

$$\mathbf{y}_t = W_{ym} \mathbf{m}_t + \mathbf{b}_y \quad (10)$$

where  $\otimes$  represents the scalar product of two vectors. The  $W$  terms denote weight matrices (e.g.,  $W_{ix}$  is the weights of matrix from the input gate to the input.  $W_{fx}$ ,  $W_{cx}$ ,  $W_{ox}$  are similarly defined).  $W_{ic}$ ,  $W_{fc}$ ,  $W_{oc}$  are diagonal weight matrices for peephole connections. These peephole connections make a gate to know the real state of the memory cell before it is handled by the output gate.  $\mathbf{I}_t$ ,  $\mathbf{f}_t$ ,  $\mathbf{o}_t$  and  $\mathbf{C}_t$  are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are of the same size as the cell output activation vector  $\mathbf{m}_t$ , the  $\mathbf{b}$  terms denote the bias vectors (e.g.,  $\mathbf{b}_i$  is the input gate bias vector).  $\tanh(\cdot)$  is the cell input and cell output activation

function, and  $\sigma(\cdot)$  denotes the standard logistics sigmoid activation function defined by

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{11}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

The LSTM network has memory, which is capable of remembering over long sequences. However, the current LSTM NN build state over the batch with relatively poor robustness. This paper improves the robustness by adding a state over the epoch. Normally, the state within the network is reset after each training batch and fitting the model, as well as each call to model prediction or model evaluation. Finer control can be obtained when the internal state of the LSTM network is cleared in Keras by making the LSTM layer “stateful”. In the improved model, when the LSTM layer is constructed, the stateful parameter must be set true. Instead of specifying the input dimensions, the number of samples in a batch, number of time steps in a sample and number of features in a time step must be coded by setting the inherent parameter (batch\_input\_shape). Simultaneously, the same batch size must be used later when evaluating the model and making predictions. These all mean that the new model can build state over the entire training sequence and even maintain that state if needed to make predictions. This novel model is named S-LSTM NN, and its robustness in prediction is compared with that from the original model (LSTM) with numerical studies.

The S-LSTM NN architecture is shown in Fig. 3. The number of nodes in the input layer and output layer of the neural network is determined by the characteristics of the task and sample. During the forecasting process, the first five steps of acceleration information were used as input, with the next step as output. There are input layers with 3 input neurons each and one feed-forward hidden layer with 128 neurons taking in the features of the original information and project them to a high dimensional space. Features in the temporal correlated data with long time dependencies were automatically extracted from the S-LSTM layer (256 memory blocks). The last two feed-forward hidden layers (128 and 128 neurons respectively) add additional nonlinear transformations before being finally fed to the output layer with 1 output neurons. Neurons between neighboring layers of the neural network are fully connected. The activation function of each neuron in a feed-forward hidden layer is *tanh* function.

The training of the S-LSTM NN is based on truncated Back Propagation Through Time (BPTT) and a modified version of Real Time Recurrent Learning using the gradient descent optimization method (Boden 2001) was adopted. The common objective function is to minimize the sum-of-squares-errors. The errors are truncated when they arrive at a memory cell output. When they enter the memory cell’s linear CEC, the errors can flow back forever, making error flow outside the cell tends to decay exponentially (Gers *et al.* 2000). Detailed execution steps are not covered in this section, and interested readers may refer to Gers’s work for more information (Gers 2001). To evaluate the effectiveness of the proposed S-LSTM NN, the root-mean-squares-error (RMSE) on the estimation was used in our experiments with

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [y_i - \hat{y}_i]^2} \tag{13}$$

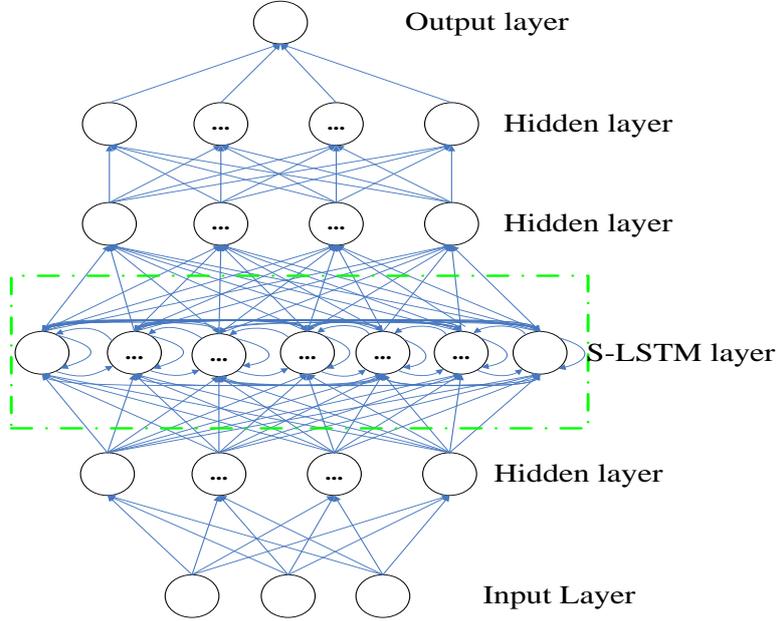


Fig. 3 The S-LSTM RNN architecture

where  $y_i$  is the observed acceleration,  $\hat{y}_i$  is the predicted acceleration, and  $n$  is the number of test samples.

The training of the S-LSTM NN is in the following steps:

- (1) Importing all of the functions and classes. This assumes a working Pycharm environment with the Keras deep learning library installed.
- (2) Loading the dataset to match the format of Pandas dataframe.
- (3) Since to the LSTMs are sensitive to the scale of the input data, specifically when the  $\sigma(\cdot)$  and  $\tanh$  activation functions are used, it can be a good practice to normalized the data in the range from null to unity.
- (4) Splitting the ordered dataset into training and test datasets. In this paper, 68% of the dataset that was used to train the S-LSTM NN leaving the remaining 32% for testing.
- (5) Conduct the training of the network and estimation on the sensor data.

### 3.2 Fault threshold setting

The established S-LSTM NN was used to calculate the prediction value, and the residual  $\mathbf{Re}$  is taken as an index to detect sensor fault.  $\mathbf{Re}$  is calculated from the prediction value and the observed value, as shown in Eq. (14).

$$\mathbf{Re} = \|\hat{\mathbf{y}} - \mathbf{y}\|^2 \quad (14)$$

According to the confidence interval of parameters in statistics, the mean and variance of  $\mathbf{Re}$  are calculated via Eqs. (15) and (16) as

$$\mu(\mathbf{R}_e) = \frac{1}{n} \sum_{i=1}^n \mathbf{R}_{ei} \tag{15}$$

$$\sigma^2(\mathbf{R}_e) = \frac{1}{n-1} \sum_{i=1}^n [\mathbf{R}_{ei} - \mu(\mathbf{R}_e)]^2 \tag{16}$$

where  $\mathbf{R}_{ei}$  is reconstruction error at the  $i$ th time instant.

The confidence interval of the mean with confidence level  $(1-\alpha)$  can be expressed as:

$$P\left(\mu - Z_{\frac{\alpha}{2}}\sigma, \mu + Z_{\frac{\alpha}{2}}\sigma\right) = 1 - \alpha \tag{17}$$

where  $\alpha$  is the confidence level;  $Z$  is the confidence level related coefficient. In this paper, the confidence level is 99.74%, and  $Z$  is 3. Thus, the threshold  $\lambda$  can be obtained as

$$\lambda = \mu(\mathbf{R}_e) + 3\sigma^2(\mathbf{R}_e) \tag{18}$$

This threshold can be selected for determining the sensor fault. For example, when  $\mathbf{R}_{ei} > \lambda$ , the sensor fault will be acceptable and vice versa.

### 3.3 Procedures on sensor fault detection

The algorithm on the detection of sensor fault is summarized as follows:

- (1) Obtain the acceleration data when the sensor is normal.
- (2) Data pre-processing and splitting the data into training and test datasets.
- (3) Construct the S-LSTM NN for data prediction.
- (4) Calculate the residual between the predicted value and the observed value of the training dataset.
- (5) Compute the fault threshold ( $\lambda$ ) of each sensor by Eq. (18).
- (6) Compute the residual of all sensors by Eq. (14).
- (7) The residuals are plotted to compare with the fault threshold of exceedance.
- (8) If an unacceptable number of exceedance of the threshold occurs, it indicates a sensor fault.

## 4. Numerical and experimental examples

### 4.1 Numerical verification

The structure is a 40 m long three-span continuous beam with a 0.25 m×0.6 m uniform cross-section. The elastic modulus and density of material are respectively  $3 \times 10^7$  kN/m<sup>2</sup> and 2500 kg/m<sup>3</sup>. The poisson ratio is 0.3. The beam is divided into 200 0.2 m long elements. It was excited with nonstationary excitation at the supports. The sampling rate is 100 Hz and the response time duration studied is 400 seconds. Newmark- $\beta$  method is used to calculate the acceleration response of the beam (Mehranbod *et al.* 2003). Ten sensors (S1~S10) were mounted on the beam at 3 m, 6 m, 9 m, 15 m, 18 m, 21 m, 25 m, 33 m, 35 m and 38 m from the left end as shown in Fig. 4.

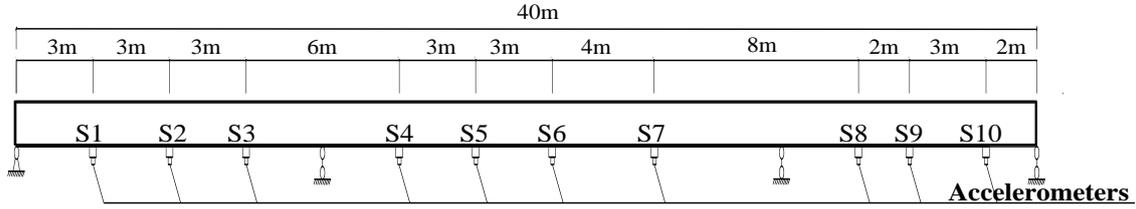


Fig. 4 Simply supported beam model

Table 1 The fault in Sensor S1 for different scenarios

Scenarios	1	2	3	4
Fault type	Constant	Gain	Bias	Bottom noise
Parameters	$c, \delta$	$\beta, \delta$	$d, \delta$	<i>noise</i>

Table 2 Fault thresholds

Sensor	Fitted Mean $\mu(\mathbf{R}_e)$	Fitted Variance $\sigma^2(\mathbf{R}_e)$	Threshold $\lambda$
S1	0.0180	0.0092	0.0455
S6	0.0607	0.0693	0.2685
S8	0.0339	0.0331	0.1310

The four different types of sensor faults discussed above are studied. Sensor S1 is simulated to have only one type of fault at one time, and the sensor faults are described in Table 1. In this paper, one representative sensor was randomly selected from each span for illustration. The fault thresholds of the three selected sensors obtained from Eq. (18) are shown in Table 2.

A total of 5 scenarios were studied including the 4 scenarios with faulty sensor and scenario 5 without fault. Each sensor response sample consists of 40000 data. The S-LSTM and original LSTM networks are used to predict the acceleration data of S1, S6 and S8. The predicted response of sensor S1 is shown in Figure 5. The enlarged part of the figure shows that the traditional LSTM under-estimates the acceleration response. The RMSE of S-LSTM is 0.03, and that from the traditional LSTM is up to 0.23. This is probably due to the insufficient learning capability of past events in the traditional LSTM.

The threshold value and the  $\mathbf{R}_e$  from previous sections are shown in Figs. 6 and 7. Since only sensor S1 has fault in scenarios 1 to 5, the  $\mathbf{R}_e$  pattern for sensor S1 may varies but those for other sensors remain the same as for the healthy scenario as shown in Figs. 6 and 7.

The signal from faulty sensor S1 is noted to have different detection result as shown in Figs. 5 and 6 compared to those from the healthy sensor (S6 and S8). All the  $\mathbf{R}_e$  values are much larger than the threshold value in Table 2 (0.0455) for the healthy sensor. It is noted that the most  $\mathbf{R}_e$  values of sensors S6 and S8 are less than their threshold. Even if there are individual values exceeding the threshold, they do not exceeding the threshold by more than  $(1-99.74\%)=0.26\%$  probability. This provides clear evidence that a comparison of the residual between scenarios with and without fault could distinguish the faulty sensor.

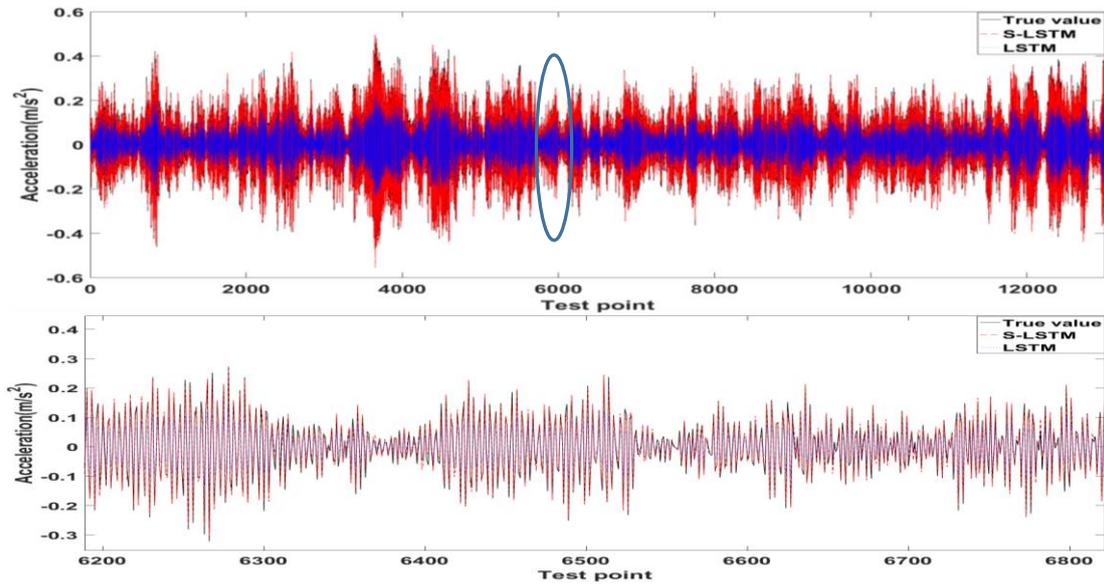


Fig. 5 Acceleration prediction performance comparison

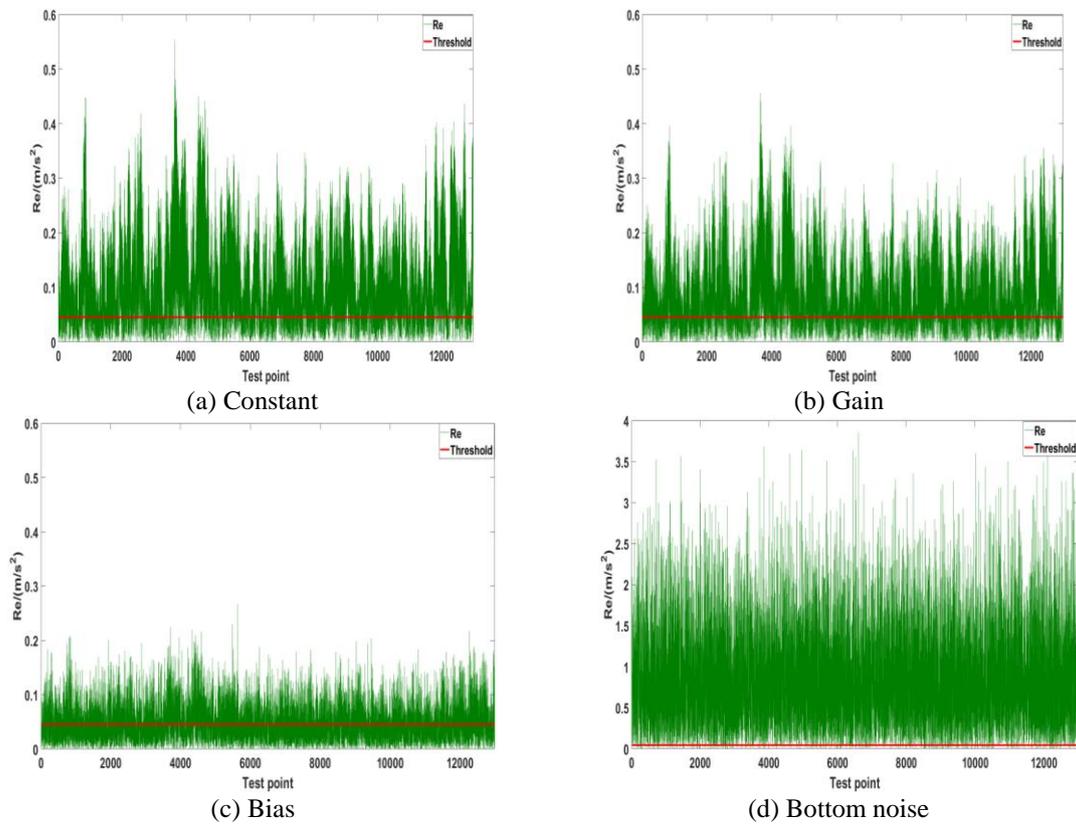


Fig. 6 The residual of Sensor S1 (scenarios 1 to 4)

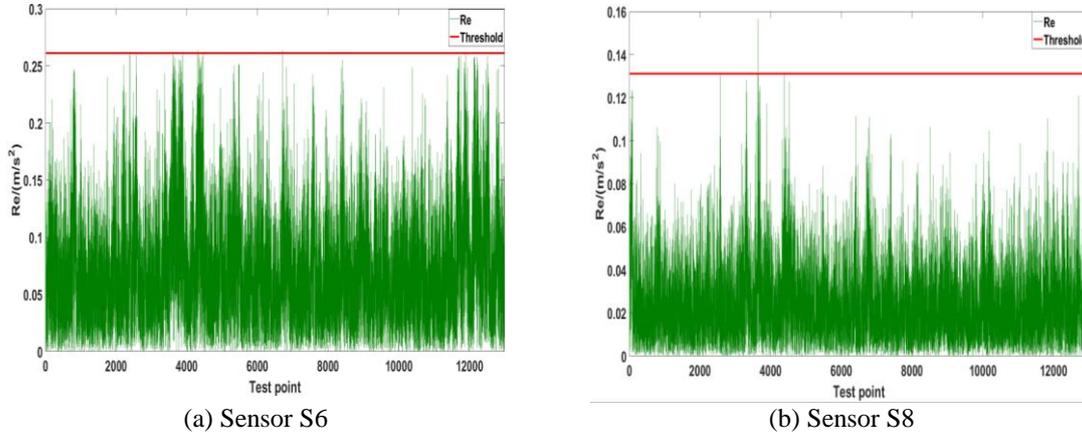


Fig. 7 The residual of Sensor S6 and S8

#### 4.2 Experimental study

An experiment was conducted on the Dong Shui Men Yangtze River Bridge which is a light-railway and highway cable-stayed bridge as shown in Fig. 8. The cable-stayed bridge has a double tower and single cable plane with 222.5 m+ 445 m + 190.5 m spans and a total length of 858 m. The left and right towers are 172.6 m and 162.5 m high respectively.

Thirty-one dynamic sensors were mounted on the bridge with fourteen accelerometers on the main deck to measure the vertical responses. Detailed layout of the measured deck sections and the sensor locations are shown in Fig. 9.

The monitoring system of the bridge collected data since October 2014. The acceleration response of accelerometer ZL51-S1 on the bridge deck is shown in Fig. 10.

Similar to the above, four real bridge sensors (ZL11-S1, ZL31-S1, ZL51-S1 and ZL81-S1) are randomly selected for algorithm validation. The results are plotted in Fig. 11. It is noted that no sensor has the  $Re$  values exceeding the threshold by more than 0.26% probability. Therefore, it may be concluded that the accelerometers of the bridge deck are in healthy condition, which is consistent with results from manual inspection, it also shows that the proposed deep learning method can correctly diagnose that whether the sensor has a fault.



Fig. 8 Dong Shui Men Yangtze River Bridge

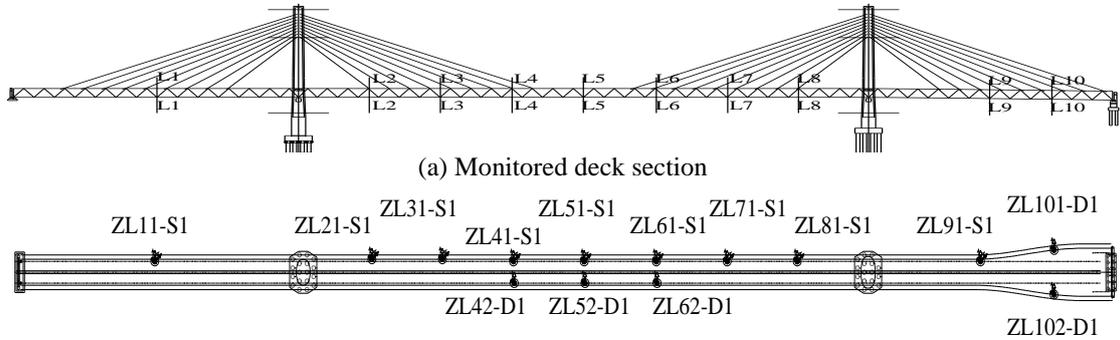


Fig. 9 Arrangement of accelerometers

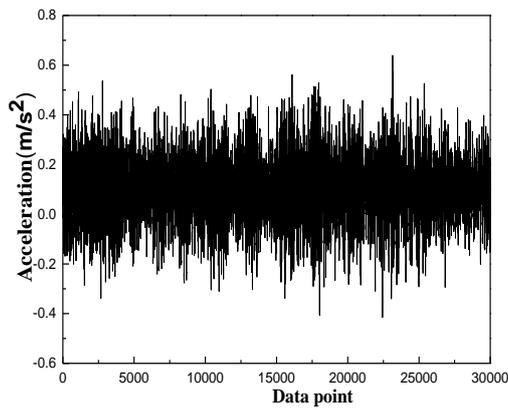


Fig. 10 Acceleration signal of real bridge

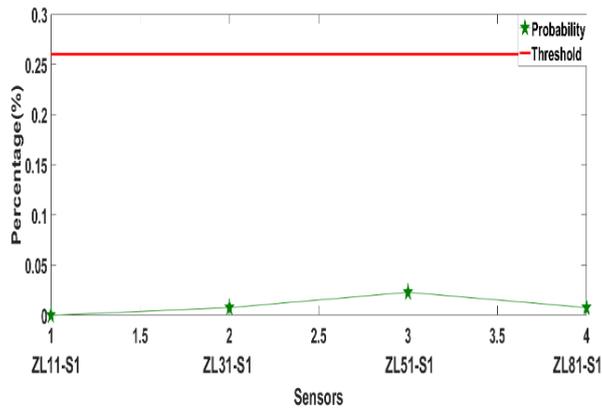


Fig. 11 Checking diagram of acceleration sensor

## 5. Conclusions

Sensors in the SHM system collect information of the structure for different types of managerial decision on the operation and performances of the infrastructure. Incorrect data from a faulty sensor is likely to mislead further structural safety evaluation and causes alarms. The proposed strategy with an improved LSTM NN, the S-LSTM NN, was used to predict the signal of the sensor. This strategy is applied to numerical and real cable-stay bridge for verification, and a fault threshold based on the residual is proposed. For the healthy sensor, it is noted that the most residual values of sensors are less than their threshold. Even if there are individual values exceeding the threshold, they do not exceeding the threshold by more than 0.26% probability, Conversely, all the residual values are much larger than the threshold value when the sensors are fault. Numerical and experimental results from a cable-stay bridge show that an automated method to detect faulty sensors based on S-LSTM is successful to quickly detect the type of fault in sensors.

## Acknowledgments

This research work was supported by the National Natural Science Foundation of China (Grant Nos. 51578095, 51778093). The comments and work by Prof. S.S. Law in improving the English standard of this paper is fully acknowledged.

## References

- Auli, M., Galley, M., Quirk, C. and Zweig, G. (2013), "Joint language and translation modeling with recurrent neural networks", *Emnlp*, **13**(10) 1044-1054.
- Boden, M. (2001), "A guide to recurrent neural networks and backpropagation", *Electr. Eng.*, **1**(2), 1-10.
- Clough, R. W., & Penzien, J. (2013). Dynamics of Structures. Dynamics of Structures, Berkeley, California, USA.
- Ge, Z., Song, Z. and Gao, F. (2013), "Review of recent research on data-based process monitoring", *Ind. Eng. Chem. Res.*, **52**(10), 3543-3562.
- Gers, F. (2001), "Long short-term memory in recurrent neural networks", *Lausanne*, **23**(66), 102-109.
- Gers, F.A., Schmidhuber, J. and Cummins, F. (2000), "Learning to forget: Continual prediction with LSTM. neural computation", *Neural Computation*, **12**(10), 2451-2471.
- Goel, P., Dedeoglu, G., Roumeliotis, S.I. and Sukhatme, G.S. (2000), "Fault detection and identification in a mobile robot using multiple model estimation and neural network", *Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, **3**(April), 2302-2309.
- Hochreiter, S. and Urgan Schmidhuber, J. (1997), "Long short term memory", *Neural Computation*, **9**(8), 1735-1780.
- Hwa-Lung, Y. and Chih-Hsin, W. (2010), "Retrospective prediction of intraurban spatiotemporal distribution of PM2.5 in Taipei", *Atmosph. Environ.*, **44**(25), 3053-3065.
- Kerschen, G., Boe, P. De, Golinval, J. and Worden, K. (2004), "Sensor validation using principal component analysis", *Smart Mater. Struct.*, **14**(1), 36-42.
- Li, L. and Zhou, DH.(2004), "Robust fault diagnosis for nonlinear systems based on analytical model", *Inform. Control*, **33**(4), 451-456.
- Ma, X., Tao, Z., Wang, Y., Yu, H. and Wang, Y. (2015), "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data", *T. Res. Part C: Emerging Technologies*,

- 54(4), 187-197.
- Mansouri, M., Nounou, M., Nounou, H. and Karim, N. (2016), "Kernel PCA-based GLRT for nonlinear fault detection of chemical processes", *J. Loss Prevent. Proc.*, **40**(1), 334-347.
- Mehranbod, N., Soroush, M., Piovosio, M. and Ogunnaike, B.A. (2003), "Probabilistic model for sensor fault detection and identification", *AICHE J.*, **49**(7), 1787-1802.
- Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y. and Sun, X. (2011), "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature", *Decis. Support Syst.*, **50**(3), 559-569.
- Sak, H., Yang, G., Li, B. and Li, W. (2016), "Modeling dependence dynamics of air pollution: Pollution risk simulation and prediction of PM 2.5 levels", *Atmos. Environ.*, **42**(1), 4567-4588
- Sheng, C. (2016), Nonlinear Damage Identification of Respiratory Crack in Vehicle - bridge Coupling System, Master. Dissertation, Chongqing University, ChongQing.
- Sharifi, R., Kim, Y. and Langari, R. (2010), "Sensor fault isolation and detection of smart structures", *Smart Mater. Struct.*, **19**(10), 55-67.
- Smarsly, K. and Law, K.H. (2014), "Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy", *Adv. Eng. Softw.*, **73**(1), 1-10.
- Yu, D. (1997), "Fault diagnosis for a hydraulic drive system using a parameter-estimation method", *Control Eng. Pract.*, **5**(9), 1283-1291.

