Multi-class Damage Detection and Localization Using Long Short-Term Memory (LSTM) Networks

Sandeep Sony, PhD

ABSTRACT

In this paper, a novel method is proposed for detecting and localizing structural damage by classifying acceleration responses of a structure using a long short-term memory (LSTM) network. Windows of samples are extracted from acceleration responses in a novel data pre-processing pipeline, and an LSTM network is developed to classify the signals into multiple classes. A predicted classification of a signal by the LSTM network into one of the damage levels indicates a damage detection. Furthermore, multiple signals obtained from the vibration sensors placed on a structure are provided as input to the LSTM model, and the resulting predicted class probabilities are used to identify the locations with high probability of damage. The proposed method is validated on the experimental setup of the Qatar University Grandstand Simulator (QUGS) for binary classification, as well as, full-scale study of the Z24 bridge benchmark data for multi-class damage classification. Experiments show that the proposed LSTM-based method performs on par with 1D convolutional neural networks (1D CNN) on the QUGS dataset, and outperforms the 1D CNN on the Z24 dataset. The novelty of this study lies in the use of recurrent neural network based LSTM for vibration data for multi-class damage identification and localization.

KEYWORDS

Structural health monitoring; LSTM; 1D CNN; damage detection; damage localization; data augmentation.

1. INTRODUCTION

Next-generation civil infrastructure is designed and constructed using state-of-the-art

methodologies and construction materials. However, integrity and condition monitoring of existing ageing structures is still lacking in advancements. Over time, structures deteriorate and lose their load-carrying capacity due to various factors such as environmental, heavy-traffic, and human-induced damages, requiring cutting-edge technology for continuous structural inspection. Recently, the American Society of Civil Engineers (ASCE) reported the current condition of infrastructure in the United States with a low grade of D+. Canadian infrastructure has been ranked along the same lines; it has been reported that it would take 1.1 trillion to replace the crumbling infrastructure of Canada with 141 billion assets in 'deplorable' condition. In can be inferred that, the development of efficient strategies for continuous structural monitoring is of paramount importance for the ageing structures. The conventional approach is to employ a well-trained structural inspector to investigate and inspect the structure, identify defects and implement appropriate maintenance strategies. However, manual assessment of the structures is subjective, error-prone, laborious, and incurs a significant portion of the annualized maintenance budget. Structural Health Monitoring (SHM) provides sensor-driven real-time inspection technology to address these challenges of manual visual inspection. An ideal SHM system consists of two major components: (i) a network of sensors for collecting response measurements, and (ii) advanced data analysis algorithms to evaluate as-is structural conditions (Cawley 2018, Qarib and Adeli 2014).

SHM employs suitable diagnostics algorithms and assists infrastructure owners and decision-makers in maximizing the safety, serviceability, and functionality of critical structures. A continuous SHM will allow efficient and cost-effective disaster management and lead to resilient infrastructure with faster recovery under natural disasters. Vibration-based SHM has emerged as one of the promising fields for condition monitoring and damage diagnosis of civil infrastructure (Gatti 2019, Erazo et al. 2019, Okayasu and Yamasaki 2019) and offers a viable option for tracking time-varying damages in the structures based on the measured data. Existing damage identification techniques involve analysis in time-domain, frequency-

domain, and time-frequency domain (Sirca and Adeli 2012, Sony and Sadhu 2019, Sony and Sadhu 2020, Sony and Sadhu 2021, Sony 2021) along with a combination of artificial intelligence techniques (Ying et al. 2013, Salehi and Burgueno 2018, Rafiei and Adeli 2017, Rafiei and Adeli 2018, Perez-Ramirez et al. 2019).

Artificial intelligence (AI) methods have successfully been applied to solve challenging tasks in several engineering domains and to automate and improve the classification and data mining tasks (Pouyanfar et al. 2019). Likewise, AI techniques provide promising opportunities for detection and localization of damages in civil infrastructure by analyzing various sensor measurements with minimal user intervention, thereby reducing cost and increasing accuracy and reliability. SHM community has adopted established Machine Learning (ML) algorithms to monitor the condition of infrastructure primarily using structural imagery (Sony et al. 2019, Salehi and Burgueno 2018). Recently, infrastructure monitoring using images of damage has garnered significant attention as a straightforward autonomous approach to monitor large scale structures, where Convolutional Neural Networks (CNNs) has gained popularity.

Historically, CNN was first introduced to classify low-resolution images of handwritten characters and was named as LeNet (LeCun et al. 1998). Since then, various CNN models with different architectures are developed. A popular ImageNet CNN model, AlexNet (Krizhevsky et al. 2012), was developed by researchers from the University of Toronto where several layers of convolution and max-pooling were used to train the database. The Visual Geometry Group of Oxford University improved AlexNet, and named VGGNet (Simonyan and Zisserman 2014) and showed how the depth of CNN influences the accuracy of image reconstruction. The development of new deeper CNN architectures introduced a trend towards using more and more (i.e., *deep*) layers. Computing giant *Google* developed a deeper network, GoogleNet (Szegedy et al. 2014), with improved dimensionality reduction and computational efficiencies. ZF-Net provided a considerable improvement in classification error rate over AlexNet. A more deeper network was developed by researchers at *Microsoft*,

and named ResNet (He et al.) with 152 layers, where each layer in the residual block was implemented as a 3*3 convolution.

Primarily designed for object recognition, 2D CNN algorithms were mostly explored for images in various SHM applications to detect defects and anomalies autonomously. Cha et al. (2017) presented a vision-based methodology for detecting cracks in concrete structures using CNN. The authors used around 40,000 images of damaged and undamaged concrete surfaces collected from various concrete structures to evaluate the accuracy of damage classification using the proposed 2D CNN architecture. Zhang et al. (2017) proposed a pixel-level CNN to detect cracks on 3D pavement surfaces. The proposed CNN, "CrackNet", was made up of two fully connected layers, one convolutional layer, one 1*1 convolution layer, and one output fully-connected layer. The proposed network was more efficient than the traditional CNN architecture because of the absence of pooling layers that downsized the output of previous layers. Zhao et al. (2018) investigated CNN for crack detection in bridges. For bridge damage classification, an AlexNet-based CNN was trained first with around 3800 images of various bridges. For recognition of bridge components, a ZF-Net-based faster regions-CNN was trained with 600 bridge images. To detect cracks, a GoogleNet-based CNN was trained with 60000 cracked and un-cracked images. Accuracies of 96.6% for bridge classification, 90.45% for bridge component classification, and 99.36% for crack detection during testing were achieved. Apart from ageing-related damage identification, image-based damage detection is also of paramount importance for post-disaster reconnaissance. Liang (2019) investigated CNN based deep learning in bridge inspection for system-level, component level, and local damage detection. The neural network was made up of a VGG-16 Transfer Learning-based NN with Bayesian optimization for classification, a faster R-CNN for component detection, and a deep CNN for semantic damage segmentation. Recently, several researchers (Ye et al. 2019, Azimi et al. 2020, Sony et al. 2021, Sun et al. 2020) provided an overview and critical review of various deep learning techniques, especially CNN, for structural damage detection. However, it was concluded that acquiring an extensive database of images of the damage in

a large-scale structure is still an issue.

Similar to 2D CNN, SHM researchers have also explored deep learning methods for effective damage detection using temporal information from sequential signals, such as acceleration measurements. Guo et al. (2014) proposed sparse coding to extract features from unlabeled acceleration measurements. The damage classification was carried out using CNN, and the results were compared with the traditional machine learning methods, such as, logistic regression and decision trees. A three-span bridge was considered to evaluate the efficacy of the proposed method, and it was shown that sparse coding-CNN based method outperforms other methods with a testing accuracy of 98%. Gulgec et al. (2017) conducted a simulation study on a steel gusset plate connection by varying the size and location of the damage. The measurements were also contaminated with 1% and 2% noise to simulate real-world conditions, and CNN was used to classify damage. The proposed method achieved an error of 2% and showed robustness against environmental noise.

One dimensional (1D) CNN (Kiranyaz et al. (2019)) have shown promising results in capturing the temporal information and damage detection and localization. Abdeljaber et al. (2017) introduced 1D CNN for real-time vibration-based damage detection. The 1D CNN configuration used in all experiments has (64,32) neurons on the two hidden convolution layers and (10,10) neurons on the two hidden fully connected layers. The output (MLP) layer size is 2, which was the number of classes. Also, each CNN has a single input neuron, which takes the input signal as the 128 time-domain samples of each frame in the training dataset. The authors trained the neural network on a vibration signal dataset obtained on a 30 joint truss structure, named Qatar Grandstand, by damaging each joint and keeping the other joints undamaged. The proposed model was trained individually on each joint, and near-perfect classification accuracy was proposed. However, the proposed method was not tested in full-scale structures. Ni et al. (2020) showed the applicability of 1D CNN with autoencoders for anomaly detection under data compression. The proposed method was validated on a long-span suspension bridge, and an accuracy of 97.53% was achieved

with a highly compressed dataset and a compression ratio of 0.1. Moreover, Zhang et al. (2019) extended the applicability of 1D CNN to detect changes in stiffness and mass. Various structural components such as a beam and steel girder bridges were used for validating the proposed algorithm, and a mean accuracy of 98% was achieved. Recently. Sharma and Sen (2020) showed the applicability of 1D CNN for damage detection in structural steel frames. Experimental validation was performed on a 2D-steel frame with different damage locations and severity of the damage. The method was shown to identify single as well as multiple damage scenarios. The false-positive rate was also evaluated and found to be well within acceptable limits. Furthermore, Liu et al. (2020) used transmissibility function-based 1D CNN to effectively identify damage at ASCE SHM benchmark structure and compared the performance against time-domain and frequency-domain methods. 1D CNN primarily exhibited showed superior performance over artificial neural networks (ANNs) in the context of computation efficiency and noise imperative for big data.

Recently, Zhang et al. (2019) proposed Long Short-Term Memory (LSTM) model for dam displacement prediction. The authors exploited the long-term dependencies learning capability of LSTM models to predict the displacement of the dam. The external environmental variables such as water pressure, temperature, structural deterioration, and bottom bedrock damage also led to a varied displacement. The study involved optimization of LSTM model to show the effects of the external environment in the resulting displacement. The proposed algorithm was compared with various ML algorithms such as support vector machine, multilayer perceptron, multiple linear regression, and boosted regression tree. It was shown that LSTM performs better than the other methods and also efficiently reflects the delays and makes the variables selection more convenient. Moreover, Yang et al. (2020) presented CNN-LSTM deep learning for computer vision-based modal frequency detection. The spatiotemporal information was extracted by using each pixel as an individual sensor from the images acquired from off-the-shelf camera. The proposed method worked effectively on noisy datasets and achieved an accuracy of 96.6%. To the best of authors knowledge, the

standalone LSTM networks have not been used in the context of structural health monitoring. The main application of a neural network model appears in (Abdeljaber et al. 2017, Abdeljaber et al. 2018), which presents a 1D CNN for binary damage classification on the Qatar Grandstand dataset. While, the 1D CNN captures relevant information in a neighborhood of samples, it lacks the ability to learn the long-term dependencies of the sequential datasets, which is relevant for structural damage identification over a long period of data.

In this study, a LSTM-based structral damage detection and localization method is proposed. The proposed method captures long term patterns in an acceleration signal by feeding a sequence of windows extracted from the signal as input to the LSTM model, allowing it to make predictions on the acceleration data. This makes the LSTM architecture a valuable technique in vibration-based SHM. This study makes four novel contributions. First, it introduces a standalone LSTM-based approach for damage localization using acceleration measurements. Second, the limited dataset is augmented by windowing the acceleration measurements and a novel approach of voting on the prediction class for windowed-dataset is presented to increase the prediction accuracy. Third, a thorough hyperparameter tuning analysis and effect of random initialization of the weights for tuned parameters is conducted and the results are compared with 1D CNN. Fourth, the proposed method is demonstrated for multi-class and multi-level damage identification in a full-scale bridge. To the authors' best knowledge, it is the first time that LSTM-based model has been presented for multi-class damage identification.

The paper is structured as follows. A brief introduction of the structural damage identification using deep learning techniques are presented in section 1, followed by the gap areas of the existing research and novelty of the proposed method. Next, section 2 presents the proposed methodology based on LSTM networks along with the data pipeline and performance metrics to identify and localize damage. The results including both binary and multiclass damage localization are illustrated in section 3. This section also highlights the importance of hyperparameter tuning, evaluation of optimal window size along with the sensitivity to

random initialization of weights. The key conclusions of the proposed research are described in section 4.

2. PROPOSED METHODOLOGY

A novel method for damage classification and localization using an LSTM network is proposed. In this section, first, a theoretical explanation of the proposed method is provided and next, performance metrics for evaluating the proposed method are discussed.

2.1 Damage detection using LSTM model

In this section, a method based on a long short-term memory network (LSTM) is proposed to classify the vibration measurement into damage levels. The mathematical illustration is provided below of a typical LSTM network and further information of the proposed method is provided subsequently. The LSTM network (Hochreiter and Schmidhuber 1997) has been successfully used in time series classification and other sequence learning tasks (Lipton et al. 2015).

Given an input acceleration signal, x=(x_1, ... , x_T), a standard recurrent neural network computes hidden vector sequence as h=(h_1, ... , h_T) and the output vector sequence as y=(y_1, ... , y_T) by iterating equations (1 - 2) from time, t =1 to T:

$$h_t = \mathcal{H}(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

where W denotes weight matrices, W_{ih} is the input-hidden weight matrix, W_{hh} is the hidden-hidden weight matrix, b denotes bias vectors, b_h is hidden bias vector, and $\mathcal{H}$ is the hidden layer activation function.

$$y_t = W_{ho}h_t + b_0 \tag{2}$$

where y_t denotes output at time t, W_{ho} is the hidden-output weight matrix, and b_o is output bias vector. $\mathcal{H}$ is usually an element-wise application of a sigmoid function. Fig. 1 illustrates a typical single LSTM memory cell (Chen 2016). The transformations applied

to the input x_t and hidden state from the previous time step h_{t-1} in the LSTM cell are described by Eqs. (3-8) (Chen 2016).

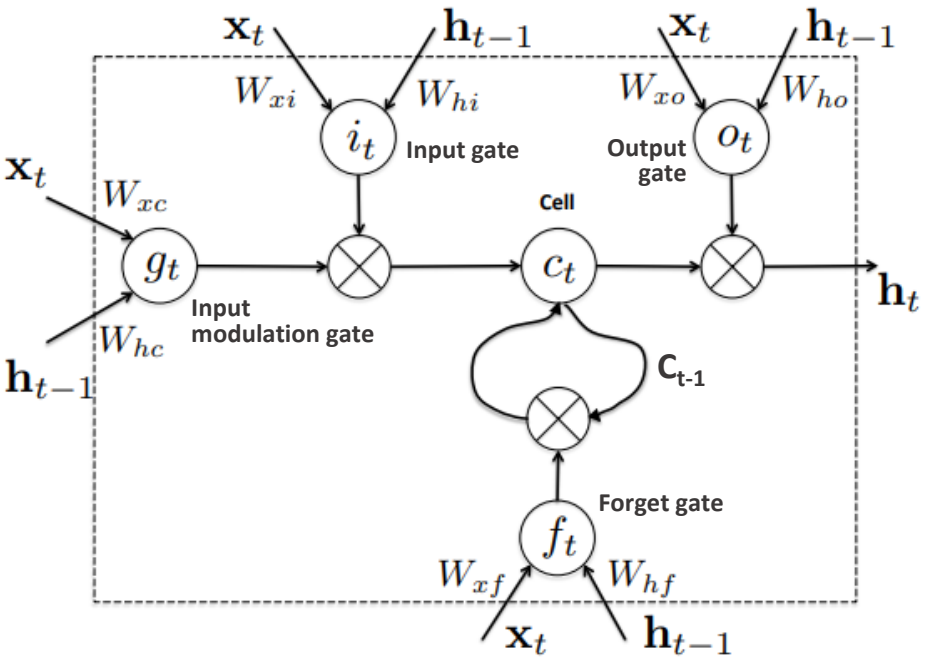

FIG. 1: The typical internal structure of an LSTM cell.

$$i_t = \tanh(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{4}$$

$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{6}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \tag{7}$$

$$h_t = \tanh(c_t) \otimes o_t \tag{8}$$

where σ is the logistic sigmoid function, and i, f, g, o and c are respectively the input gate, forget gate, input modulation gate, output gate and cell activation vectors, all of which

are the same size as the hidden vector h. The weight matrix subscripts have the obvious meaning, for example W_{hi} is the hidden-input gate matrix, W_{xo} is the input-output gate matrix. The bias terms (which are added to i, f, g,and o) have been omitted in Fig. 1 for clarity. The weight matrices represent the learnable parameters of the model and gradient decent algorithm is used to minimize prediction error on a training set.

The cell state c_t encodes the information of the sequence observed up to that time step. The input gate controls the information added to the cell state from the current time step, and the forget gate controls what information needs to be forgotten from the current cell state. For example, if the output vector of the forget gate f_t has a near-zero value in the first dimension, it indicates that the first dimension of the cell state c_t needs to be "forgotten". The forgetting occurs in the element-wise multiplication; i.e. multiplying an element by a near-zero value results in a near-zero element in the output vector. By maintaining cell state in this manner, the LSTM cell is able to capture both long and short term relationships between the input time-series values and the predicted variable (e.g., damage classification). During training, the truncated back propagation through time algorithm is used on truncated sequences to make the process computationally feasible. During prediction, the forward pass can be applied to arbitrarily long sequences (the LSTM cell can be repeatedly applied to any number of input time steps). Any further information about the internal structure and training of LSTM can be found in (Chen 2016).

The proposed machine learning model for the damage time series level classification is a multi-layer LSTM network architecture, as shown in Fig. 2. The pre-processed sequences of windows are given as input to the model, and the softmax output of the final LSTM time step is considered as the prediction of a sequence (the set of classification probabilities $P(y = c_t)$ to each class c_t. During training, forward and backward passes are performed on the input sequences, and the weight updates are made to minimize the cross-entropy loss on a batch of sequences. The predicted set of classification probabilities $P_p(y_c)$ for a full acceleration measurement is obtained by summing the class probabilities of all the window sequences

in one signal series. The class with the maximum probability is the predicted damage level classification of the series. Note that this is equivalent to voting on the classification probabilities of individual window sequences to arrive at the prediction of the full series. It is observed that the voting process improves the prediction accuracy and other evaluation metrics on the time-series test set.

In the proposed method, the window size w and the sequence length L (no. of windows in a sequence) becomes hyperparameters that are tuned to improve the accuracy of the neural network. The hyperparameters of LSTM network include number of layers and number of nodes in each layer (network architecture), activation function, and batch size (performing weight updates during training). Optimal parameters are found using a random search on a hyperparameter space (Bergstra and Bengio 2012). In each iteration, the search algorithm randomly selects a configuration of values for hyperparameters from a specified set of possible values, and trains three models with those parameters on three splits of training and validation sets (3-fold validation). The hyperparameters configuration that gives the highest mean validation accuracy is selected as the final, tuned set of hyperparameters of the model. A training session is terminated when, either a specified maximum number of epochs is reached, or the validation loss does not decrease for a specified number of epochs (early stopping). The final network weights are taken from the epoch with the smallest validation loss. More details of the experiments and the optimal hyperparameters for each setup are provided in section 4. The purpose of the proposed method is to classify acceleration responses depending on various damage levels and classes. The classification problem is presented as binary (undamaged vs. damaged) or multiclass (undamaged, and damage of more than two levels). Fig. 3 illustrates the proposed data pipeline, which consists of a series of pre-processing and post-processing steps with an LSTM network as the classification model. It should be noted that the solid black lines represent datasets, blue dotted lines represent operation on the datasets, and multiple arrows represent multiple data instances.

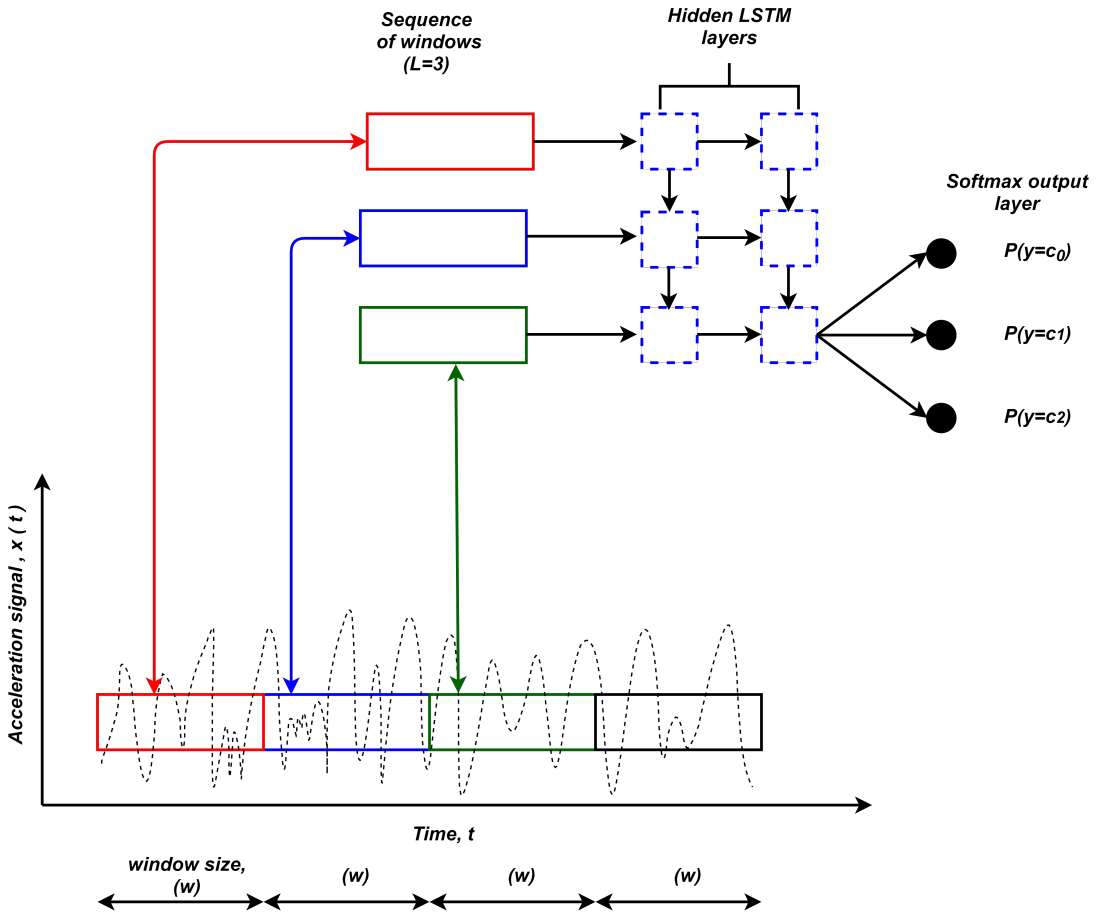

FIG. 2: Extracting sequences of windows from the vibration signals and the LSTM network architecture.

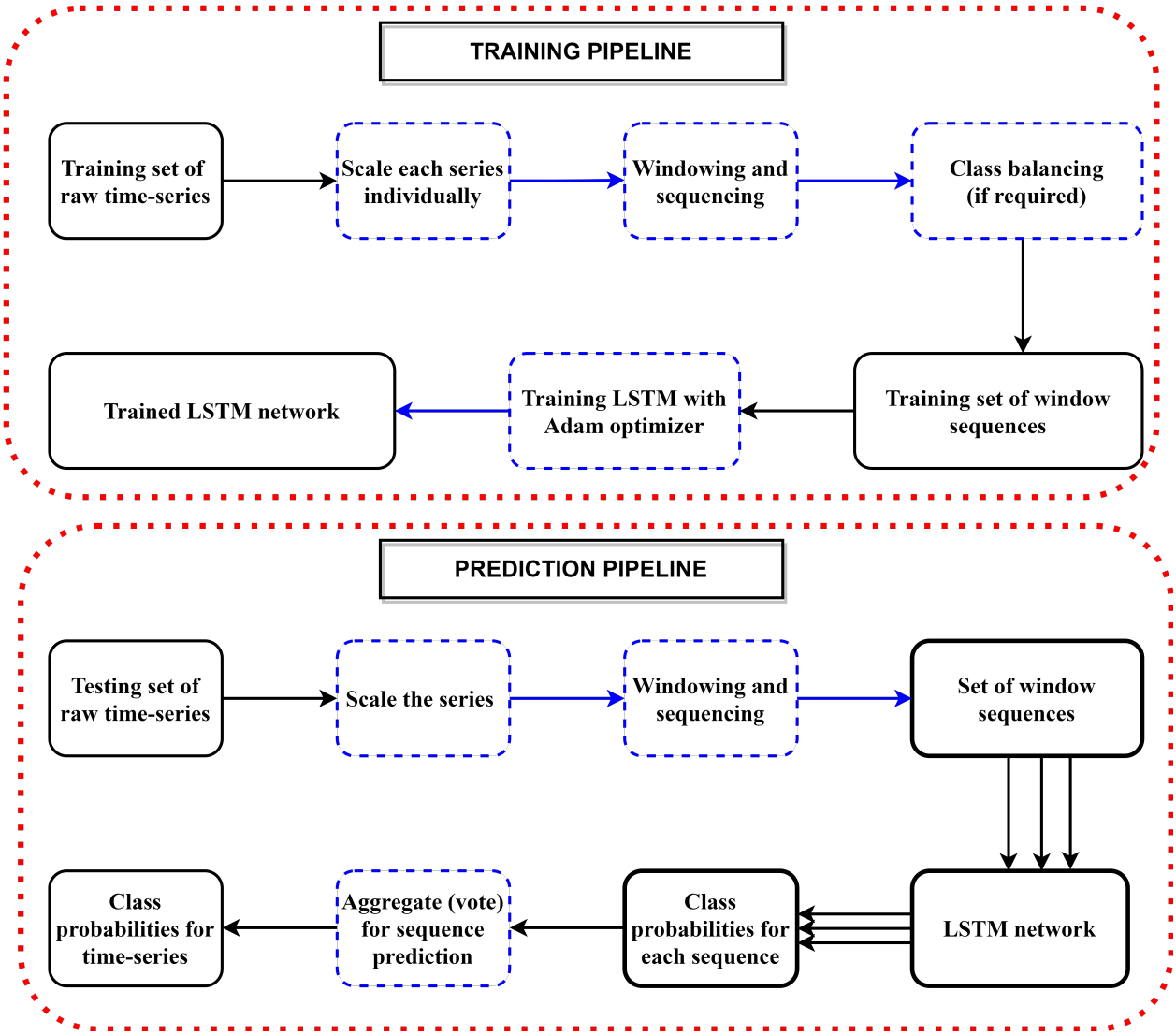

FIG. 3: Data pipelines for training the LSTM network and obtaining predictions for a given acceleration response.

A single acceleration time-series acquired over a number of seconds consists of a large number of samples (for example, a record of long vibration data). In the proposed method, the acceleration is first normalized with respect to its mean and standard deviations. This improves the convergence rate of models trained on the datasets and prevents large value samples from dominating the input (Ioffe and Szegedy 2015). Second, the segment of the scaled time series is fed into a sequence of contiguous windows (window size w), and a sequence of such windows (length L) is arranged to form one input instance to the LSTM

network. Thus, the input of the network is a w-dimensional sequence of length L. Many such sequences can be extracted from a single original accelerometer time series, and each sequence is assigned to a label (damage level) of the original series. The process of extracting sequences of windows from a time series is illustrated in Fig. 2. This technique of transforming the original series into sequences of windows effectively reduces the data dimension, and additionally, it increases the training set size (multiple sequences per time series), which in turn allows training machine learning models with less over-fitting. Imbalanced data can cause problems in model training (Guo et al. 2018). To alleviate this problem, a balanced dataset is created by preparing training set by randomly selecting a number of undamaged sequences equal to the number of damaged sequences.

2.2 Damage localization using LSTM model

The LSTM network described in the previous section produces a set of probabilities $P_p(y_c)$ for a given acceleration signal. These indicate the probability that the signal belongs to each class c. Multiple signals are obtained from accelerometers placed in the critical locations of a structure, and model prediction probabilities are computed for each signal. In this manner, a probability of damage distribution over the space of the structure can be estimated, and locations with high probability of damage can be identified (damage localization). This damage probability distribution over the structure can be visualized (for example, heatmaps of probability values over a 2D structure) to aid an engineer in quickly localizing damages. Algorithm 1 summarizes the proposed approach for damage detection and localization.

Algorithm 1: Damage detection and localization

Input: A set of acceleration signals from a structure.

Output: Predicted damage class (damage detection) and probabilities of damage over the structure (damage localization)

(a) The acceleration signal data is pre-processed, as shown in the training pipeline in Fig. 3.

(b) A single LSTM model is trained for damage classification.

(c) The predicted class label (undamaged or damage level) for a signal is computed from the model (damage detection for the signal).

(d) Probabilities of classification (to damage level classes) are computed for each signal from the model.

(e) Damage probabilities for all signals form a distribution of damage probabilities over the structure. High probabilities correspond to damaged locations (damage localization).

(f) Visualize the distribution of damage probabilities for visual inspection and automated decision.

2.3 Performance criteria

Several standard metrics can be used to evaluate the performance of a classification model. These metrics measure different aspects of the obtained results. A brief description of the selected metrics is provided below and explained in the context of structural health monitoring of civil infrastructure. The primary form of prediction results is given by the confusion matrix, which is a tabulation of classifications made by a model. It shows the "classification distribution" of a model, and helps identify properties of the model, such as when it is consistently mis-classifying one class as another. The confusion matrix is obtained for both binary and multi-class classifications. Table 1 shows the confusion matrix for the case of binary classification. It should be noted that True Positives is denoted as TP, True Negatives as TN, False Positives as FP, and False Negatives as FN. Multiple metrics can be derived from the confusion matrix as shown in Table 2.

TABLE 1: Confusion matrix for a binary classification problem.

		Predicted class	
		Damage	Healthy
Output class	Damage	TP	FN
	Healthy	FP	TN

TABLE 2: Description of various performance metrics.

Metric	Formula	Remarks
ROC-AUC	Recall Vs FPR	Degree of separability between classes
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$	Less useful for heavily imbalanced data
Precision	$\frac{TP}{TP+FP}$	Positive predicted value
Recall	$\frac{TP}{TP+FN}$	True positive rate or sensitivity
False Positive Rate (FPR)	$\frac{FP}{TN+FP}$	False alarm when there is no damage
False Negative Rate (FNR)	$\frac{FN}{TP+FN}$	No alarm for actual damage
F1 Score	$2 * \frac{precision.recall}{precision+recall}$	The harmonic mean of precision and recall

In this study, two key metrics are used as the performance metrics to evaluate the proposed method, namely, accuracy, and FNR. Accuracy is the primary evaluation metric to understand the ability of the model to correctly classify the inputs. False negative rate is an important metric that has not been commonly used in the literature to evaluate damage detection models. A false negative (type II error) represents a truly damaged series that is classified as undamaged by the model, which could lead to catastrophic consequences in a critical structure. Therefore, we measure and compare the false negative rate in the model evaluation experiments in this work. Two plots highlight the trade-off between metrics as

the decision threshold of the classifier changes: the receiver operating characteristic (ROC) curve shows the trade-off between false positive rate and true positive rate, and the precision-recall (PR) curve shows the trade-off between precision and recall. By analyzing these curves obtained from a test set of model predictions, an engineer can make an informed decision on the balance that is needed between the metrics and choose a decision threshold suitable for the task. The PR curve in particular is suitable for understanding problems where the datasets are highly imbalanced. In this study, ROC-AUC and PR-AUC curves are plotted for visual comparison. The corresponding area-under-the-curve (AUC) of these graphs (ROC-AUC and PR-AUC) represent an aggregate measure of the model ability in terms of the relevant metrics. Therefore, ROC-AUC and PR-AUC reported and compared where appropriate. The damage localization results are evaluated by inspecting the visualization of damage probabilities over the structure, and verifying that high damage probabilities have been assigned to the damaged location, while the other locations are assigned low damage probabilities.

3. PERFORMANCE EVALUATION

The proposed method is evaluated in two stages: an experimental setup, and a full-scale study. For comparison, the 1D CNN model from (Abdeljaber et al. (2017)) is also implemented and evaluated. For performance evaluation, accuracy, FNR, ROC-AUC and PR-AUC are used. Additionally, several experiments are conducted to study the performance change with window size in the model input, and the effect of voting on individual windows to obtain the final prediction.

3.1 Experimental study

Damage detection where the classification is between one of two classes, such as damaged and undamaged, is called a binary classification. Qatar University Grandstand Simulator (QUGS) is used to evaluate the performance of the proposed method (Abdeljaber et al. (2017)). QUGS is constructed to evaluate and develop effective structural damage detection techniques suitable for monitoring of modern stadia, as shown in Fig. 4. The frame was

designed to carry a total of 30 spectators with area dimensions of 4.2 m * 4.2 m. The design considerations used for the experimental test structure was to guarantee its safety and compatibility with the specifications of modern grandstands. In this work, the structure is utilized in its current form (steel frame only) to generate vibration data under several structural damage cases. The signal is applied to the shaker through a SmartAmp 2100E21-400 power amplifier. Two 16-channel data acquisition devices were used to generate the shaker input and collect the acceleration output. A total of 31 vibration scenarios are considered, the first scenario corresponds to the reference (undamaged) case. For scenarios 2 to 31, damage was introduced to the joints 1 to 30, respectively.

TABLE 3: Hyperparameters used in LSTM for tuning by random search algorithm.

Parameter	Values
Window size	64, 128, 160, 256, 512
No. of windows in a sequence	2, 4, 8, 16
No. of hidden layers	1 - 6
No. of hidden nodes	1024, 512, 256, 128, 64, 32
Droput rates	0.2 and 0.5 for each hidden layer
Learning rate	0.0003, 0.001, 0.01
Batch size	64, 256, 512
Cell type	LSTM with tanh activation

A total of 30 accelerometers were mounted on the main girder at each girder and filler beam joint, and a magnetic mounting plate was used to attach the sensors on the frame. A modal shaker was used to excite to the structure. A total of 31 measurement datasets were acquired where, first one was for the undamaged case, and in scenarios 2 to 31, the damage was introduced by loosening the bolts to the joints 1 to 30, respectively, as shown in Fig. 5.

The acceleration data was sampled at 1024 Hz and collected for 256 seconds under a white noise shaker for each joint. Therefore, each signal contains 262144 samples. The complete data set contains 31 distinct acceleration measurement datasets, where the first one is for undamaged case and the rest are for one damaged scenario each. Two sets of datasets under a similar vibration environment are acquired for training and testing of a model, respectively.

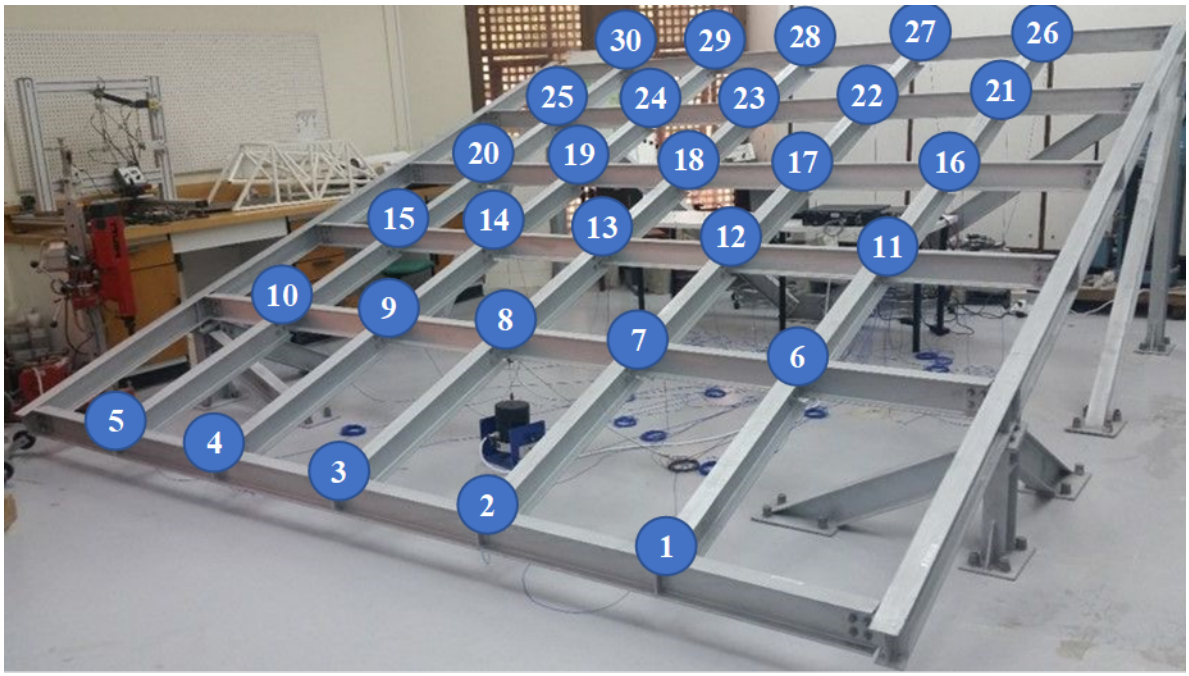

FIG. 4: QUGS testbed, where the joints are numbered from 1 to 30 (Abdeljaber et al. 2017).

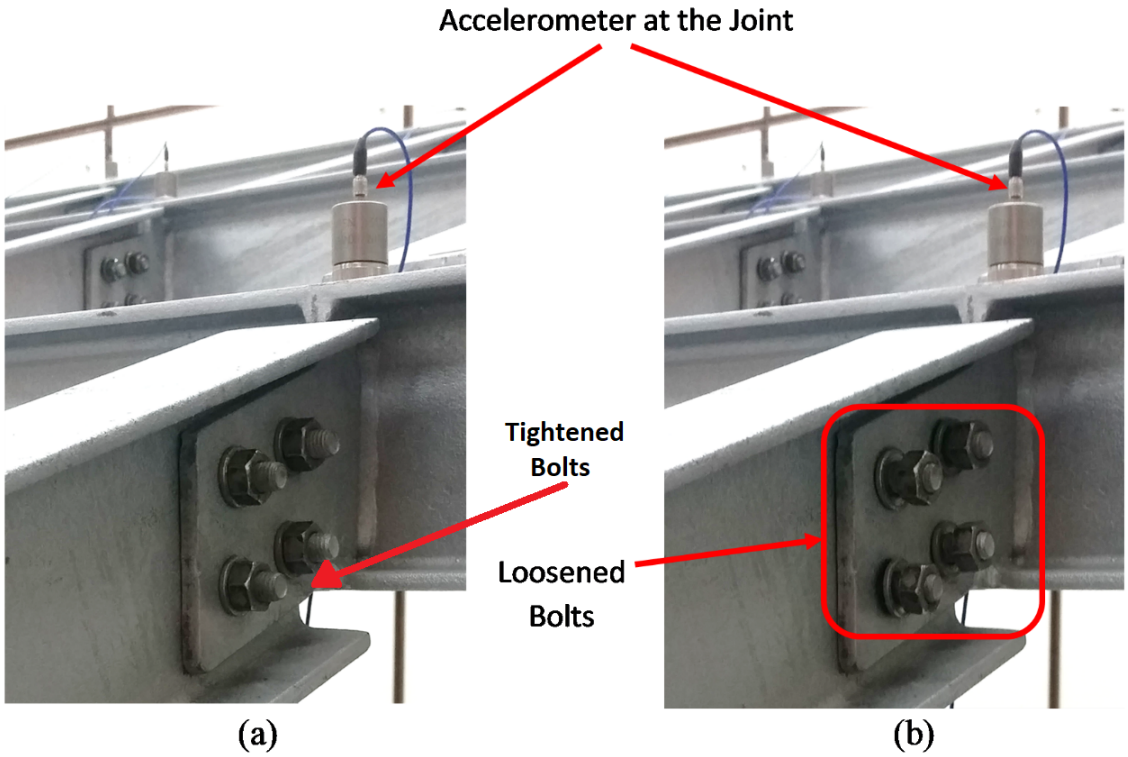

FIG. 5: Undamaged and damaged state of the girder and beam joint in QUGS (Abdeljaber et al. 2017).

TABLE 4: Optimal configuration of LSTM hyperparameters for QUGS experiment.

Parameter	Values
Window size	64
No. of windows in a sequence	8
No. of hidden layers	3
Architecture	[64, 128, 64, 32, 1]
Droput rates	0.2
Learning rate	0.001
Batch size	256
Training epochs	100 with early stopping

A range of values for hyperparameters is used for tuning the LSTM model using a random search, as shown in Table 3. The window size as an external parameter is varied between 64 and 512 samples. Various values for other hyperparameters such as the number of hidden layers (range of 1-6), and nodes in hidden layers (range of 32-1024) are considered to achieve optimal performance of the proposed model. Consequently, the random search algorithm explores shallow, wide and deep LSTM architectures. The optimal hyperparameter configuration obtained using random search algorithm for QUGS experiment is presented in Table 4. It can be observed that the highest performing window is 64, and 3 required at least three hidden layers for best accuracy. A comparison is drawn between the window-size (w) and the performance metric (P_m) to understand model performance with changing w. Three different metrics are used, namely, ROC-AUC, accuracy, and FNR as these three metrics broadly cover the efficacy and any shortcomings of the classification model. The result is shown in Fig. 6. It can be observed that w=64 and w=128 yield the best performance metric values (high accuracy and ROC-AUC with low FNR), and performance metrics consistently

degrade with increasing window size. This behavior is attributed to a reduction in the data samples (sequences of windows) with increasing w, causing the LSTM network to overfit when trained on smaller datasets.

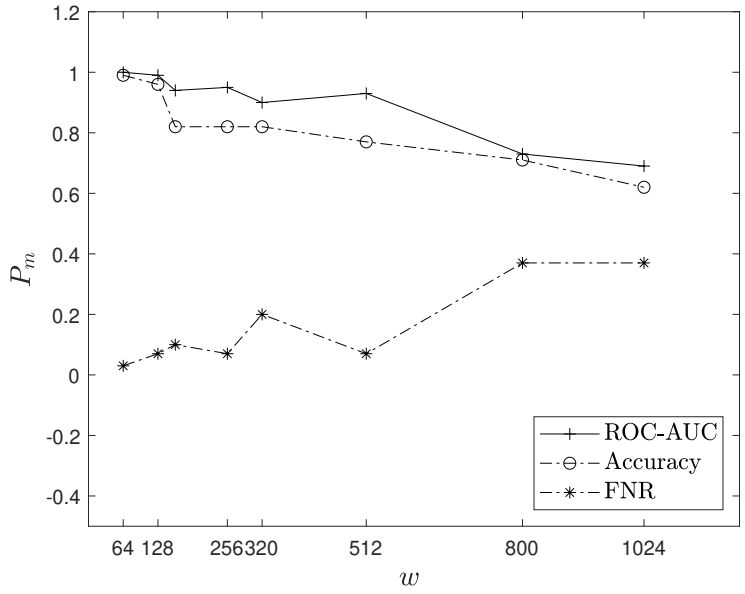

FIG. 6: Performance evaluation of LSTM based on window size for QUGS.

To compare the proposed LSTM approach with the 1D CNN method introduced by Abdeljaber et al. (2017), a 1D CNN is trained with the same hyperparameter tuning process as outlined before. The optimal 1D CNN network has 4 hidden layers with nodes 256, 128, 64, 32 in each layer and a kernel size of 64. After acquiring the optimal tuned parameters, a variability study is conducted to understand variance in the metrics by training the LSTM and 1D models with random initialization of network weights 5 times. The result is shown in Fig. 7. It was found that both models perform well consistently with accuracy at 1.0 and False negative rate at 0.0 with negligible variability.

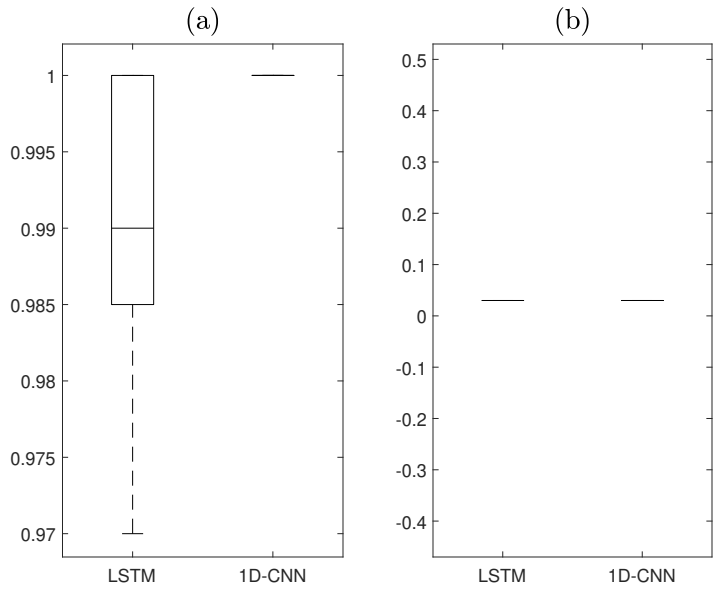

FIG. 7: Performance evaluation for random weight initialization for QUGS (a) Accuracy, and (b) FNR.

One possible explanation for the perfect performance behavior of the machine learning models is the based on the nature of the acceleration signals. The damages in the joints of the QUGS structure are highly localized, and the sensors are placed at the exact locations of damage. Furthermore, the controlled experimental laboratory setup makes the acquired signals be of high quality and noise free. These properties lead to acceleration signals that have distinct, discriminating patterns between the damaged versus undamaged cases. The 1D CNN's capability to learn local structure in the time signals, and the LSTM's capability to learn long-term irregular dependencies lead to both models learning patterns that give excellent test set performance.

In the proposed method, the damage classification of a series is obtained by voting on the classification of individual windows that constitute the full series. It is observed that this voting process increases test set performance metrics on series predictions considerably in contrast to the predictions on individual windows, as illustrated by the ROC curves and precision-recall (PR) curves in Fig. 8. The ROC curve is closer to the upper left corner in

the voted series predictions (Fig. 8: (b)) than on the individual window predictions (Fig. 8: (a)), which represents an increase in ROC-AUC from 0.95 to 1.0. The PR curve also shifts to the upper right corner (Fig. 8: (d)), with an increase in PR-AUC from 0.52 for voted series predictions to 0.99 for individual window predictions. It can be concluded that voting on windows to obtain the final series prediction decreases the probability of error significantly.

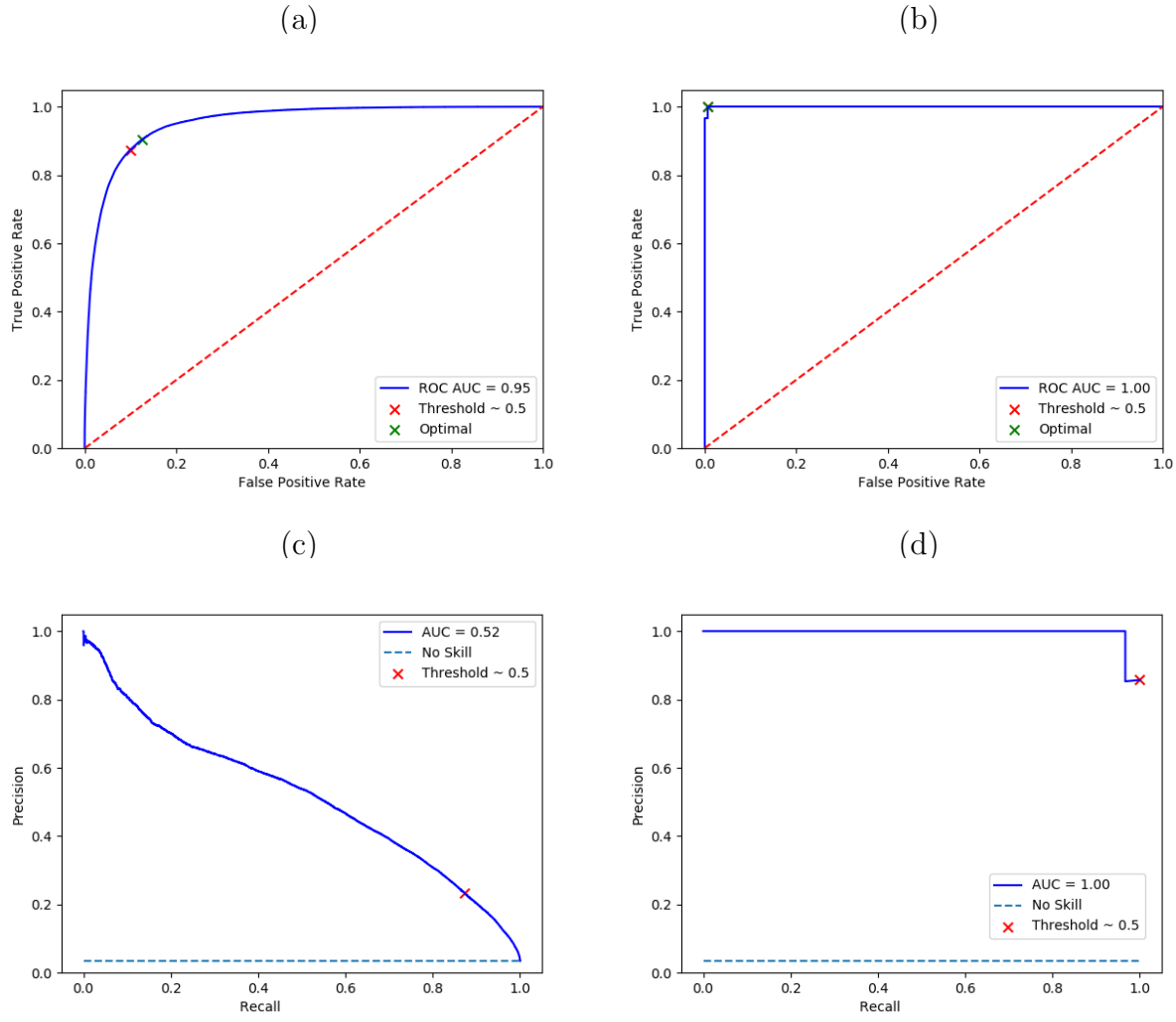

FIG. 8: LSTM performance curves on the QUGS dataset: (a) individual windows ROC, (b) voted series ROC, (c) individual windows PR, (d) voted series PR.

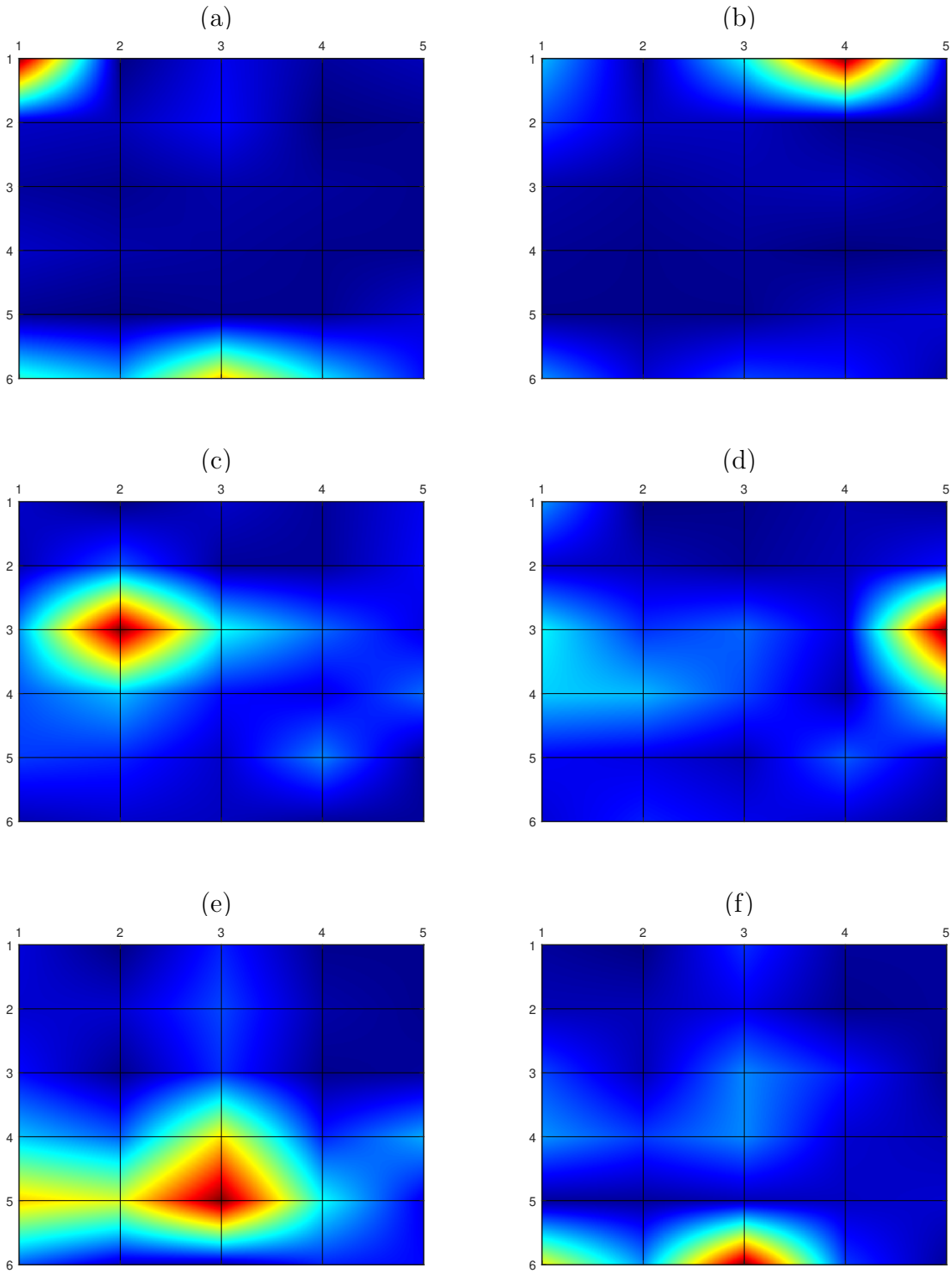

FIG. 9: Damage localization probabilities in QUGS for scenarios where the damaged joint is: (a) Joint-1 [1, 1] (b) Joint-4 [4, 1], (c) Joint-12 [2,3], (d) Joint-15 [5,3], (e) Joint-23 [3,5], and (f) Joint-28 [3, 6].

Damage localization is performed for QUGS using Algorithm 1. In each damage scenario in the QUGS dataset, a single joint out of the 30 joints in the grandstand structure is damaged by loosening of the bolts. Acceleration signals are acquired from all joints, and the proposed localization method gives damage probabilities for each joint location. The distributions of damage probabilities for multiple damage scenarios are presented as heatmaps in Fig. 9. Note that the color blue indicates $P(Damage) \approx 0$ and color red $P(Damage) \approx 1$. For example, Fig. 9 (a) shows the scenario where joint-1 of the structure is damaged (see Fig. 4 for an illustration for numbered joints). The damage appears to be heavily localized at the joint. Fig. 9 (e) shows the scenario where joint-23 (location: [3, 5]) is damaged, and it is clear that the damage is spread out as it is surrounded by 4 connected branches signifying more effect of damage after loosening of bolts.

3.2. Full-scale Study

Damage detection, where classification is more than two classes, is considered a multiclass problem. In this study, two classes of damage are used, namely, pier settlement and rupture of tendons for a full-scale bridge. Both damage classes further have multiple damage levels. Z24 bridge benchmark data (Maeck and Roeck 2003) is used to evaluate the performance of the proposed method for multiclass damage detection. The Z24 bridge was located in the canton Bern near Solothurn, Switzerland. It was a classical post-tensioned concrete two-cell box-girder bridge with a main span of 30 m and two side spans of 14 m, as shown in Fig. 10. The bridge was demolished at the end of 1998 because a new railway adjacent to the highway required a new bridge with a larger side span. The Z24 bridge data was acquired using 16 accelerometers placed at different spans of the bridge as shown in Fig. 11. The Z24 bridge was excited by two shakers, one at the mid-span of the bridge and another at a side-span. Because of the size of the bridge, response was measured in nine setups of up to 15 sensors each, with three accelerometers and the two force sensors common in all setups. The data was sampled at 100 Hz, and a total of 65536 samples were acquired. This data was made publicly available by researchers at the Katholieke Universiteit Leuven and is available

at: https://bwk.kuleuven.be/bwm/z24.

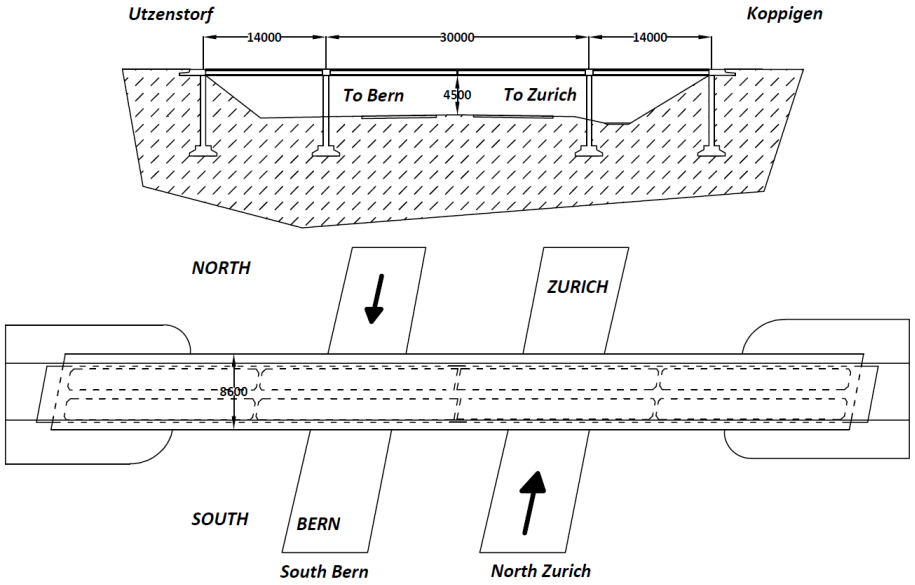

FIG. 10: Schematic of the Z24 bridge.

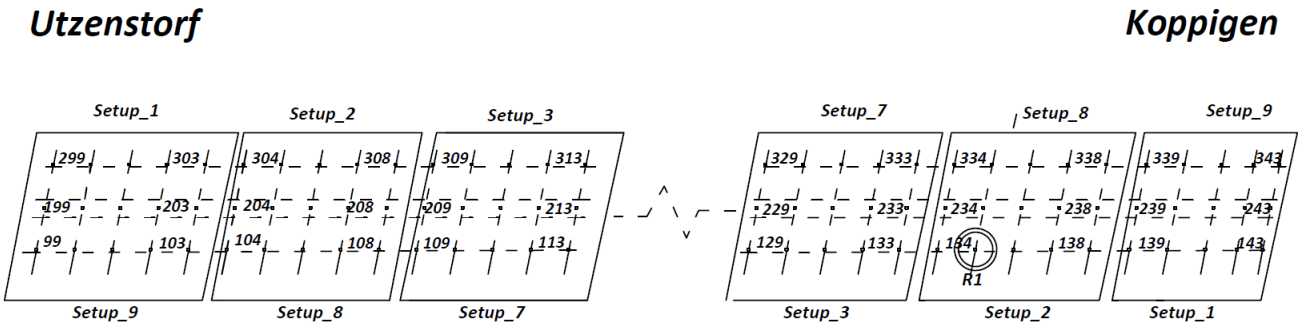

FIG. 11: Sensor placement for the data acquisition.

Several progressive damage scenarios were considered for vibration-based damage analysis. For the brevity of this study, only two different damage scenarios are considered: pier settlement and rupture of tendons. Each damage scenario has multiple level of damage. All these damage scenarios are compared with the baseline undamaged state. It can be observed that each damage scenario have different classes of damage, and they were chosen to evaluate the performance of the proposed method to classify various multi-class, multi-level damage cases. For example, rupture of tendons have three levels, and lowering of pier have four

levels, and together they make a case of two separate damage classes. For detailed explanation of how the damages were induced to the bridge, readers are suggested to referred to (Roeck and Teughels 2004). Multiclass problem is considered based on the type and level of damage. The reference, undamaged condition is considered as class-zero for all the cases and the other damages were assigned classes starting from 1 to n depending upon the level of damage, as shown in Table 5. For example, in the case of rupture of tendons, the damage level classes 1, 2 and 3 correspond to damages induced by rupture of two tendons, then four, and finally six tendons. Similarly, there are four damage level classes for lowering of pier.

TABLE 5: Multiclass problem description for two damage scenarios along with the class label.

Problem	Damage scenario	Class label
0	Undamaged	0
1	Rupture of 2 tendons	1
	Rupture of 4 tendons	2
	Rupture of 6 tendons	3
2	Lowering of pier, 20 mm	1
	Lowering of pier, 40 mm	2
	Lowering of pier, 80 mm	3
	Lowering of pier, 95 mm	4

Optimal hyperparamters of the LSTM model on the Z24 bridge dataset are obtained by performing a random search. Table 6 shows the hyperparameter configuration with highest accuracy. It can be observed that LSTM performed well with w=128. An analysis is performed to understand the effect of window size w on performance P_m. The results are illustrated in Fig. 12, which shows that optimal performance is achieved at w=128, with

highest ROC-AUC and accuracy, and lowest FNR. In case of Fig. 12 (a), the ROC-AUC and accuracy reduces while FNR increases, similarly, in case of Fig. 12(b) the false negative rate remains consistent after w=512 and other metrics are at their peak, due to larger w, the data size reduces per damage class and it leads to overfitting of the model on the data. A similar random search of hyperparameters is conducted to find optimal parameters for the 1D CNN model. It is found that a CNN with a single hidden layer with 32 ReLU nodes and kernel size 16 gives the highest accuracy. 5 repetitions of training and testing sessions are performed to verify the robustness of the models against random initialization of network weights. Accuracy and false negative rate are computed for each session, and the results are shown in Fig. 13.

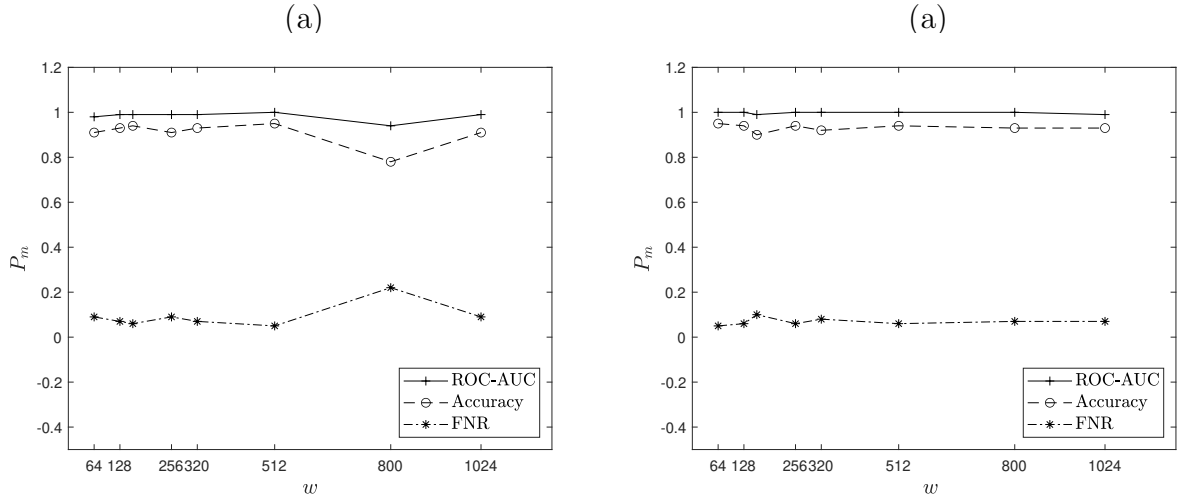

FIG. 12: Performance evaluation of LSTM based on window size for (a) rupture of tendons, and (b) pier settlement.

It can be observed that for pier settlement which is mounted with 4-sensors for data acquisition, and there are 4 damage levels, the damage is considered as localized around these 4 sensors. The LSTM performs well with highest accuracy and lowest FN rate in comparison to 1D CNN as seen in Fig. 13 (a), and (b). In case of rupture of tendons, where the damage is fairly distributed through out the bridge deck. The damaged signals are not highly distinguishable in comparison to undamaged signal and other severe damage

measurements such as lowering of piers. LSTM's accuracy drops to 0.9 and FN rate increases to 0.2. However, LSTM still performs better than 1D CNN as shown in Fig. 13 (c), and (d). It can be further emphasized that the superior performance of LSTM is attributed to its capability to learn long-term irregular dependencies of complex time signals whereas 1D CNN learns prominently the local neighborhood structure of the signals.

TABLE 6: Optimal configuration of LSTM hyperparameters for the Z24 bridge dataset.

Parameter	Values
Window size	128
No. of windows in a sequence	16
No. of hidden layers	3
Architecture	[128, 64, 32, 5]
Learning rate	0.001
Batch size	512
Training epochs	100 with early stopping

Finally, the optimal parameters are used to evaluate the performance of proposed method on full-series versus voted-windowed samples. It is observed that voting on windowed dataset increases accuracy considerably and it is evident in ROC-AUC and precision-recall (PR)-AUC curves, as presented in Fig. 14, and 15, respectively. It can be observed that voting on windows from non-localized signal increases the probability considerably by allocating the majority class and ignoring the non-prominent class along with increasing the data samples per class. As shown in Fig. 14, voting on individual windows has improved both the ROC-AUC and PR-AUC. However, due to the localized measurement acquisition, and severity of damage in pier settlement, the difference in AUCs of various cases was comparatively similar to the QUGS damage scenario. Moreover, as observed in Fig. 15, where the damage

was considerably distributed in case of rupture of tendons, voting on windows increased the ROC-AUC by 4% and PR-AUC by 10%.

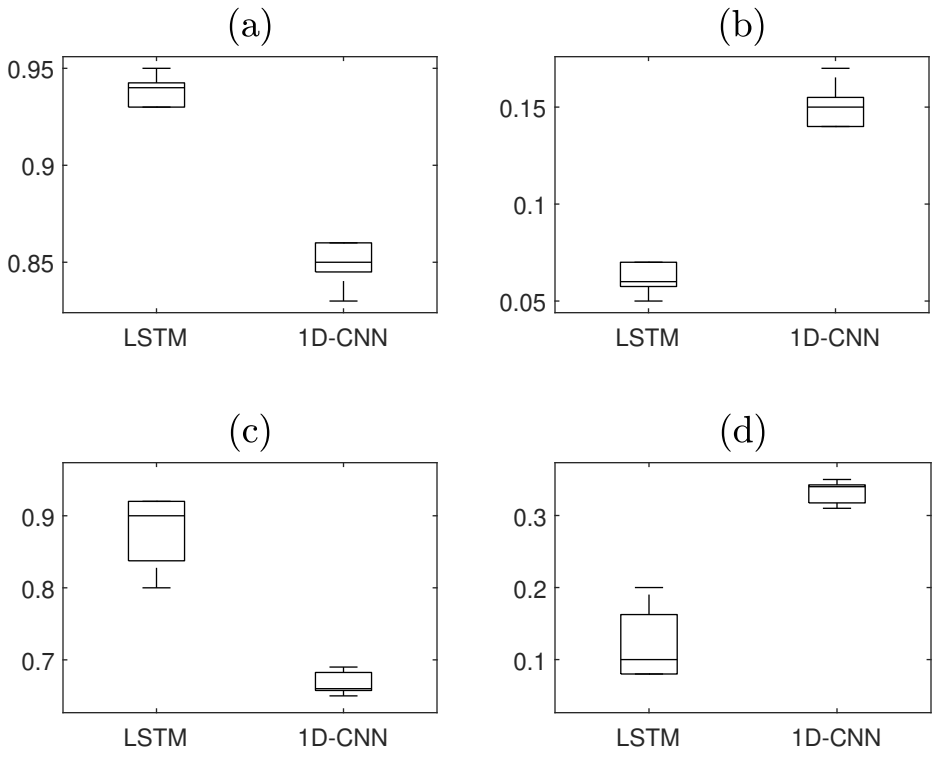

FIG. 13: Performance evaluation of LSTM for random weight initialization for various damage cases in the Z24 bridge where, (a) accuracy for pier settlement, (b) FNR for pier settlement, (c) accuracy for rupture of tendons, (d) FNR for rupture of tendons.

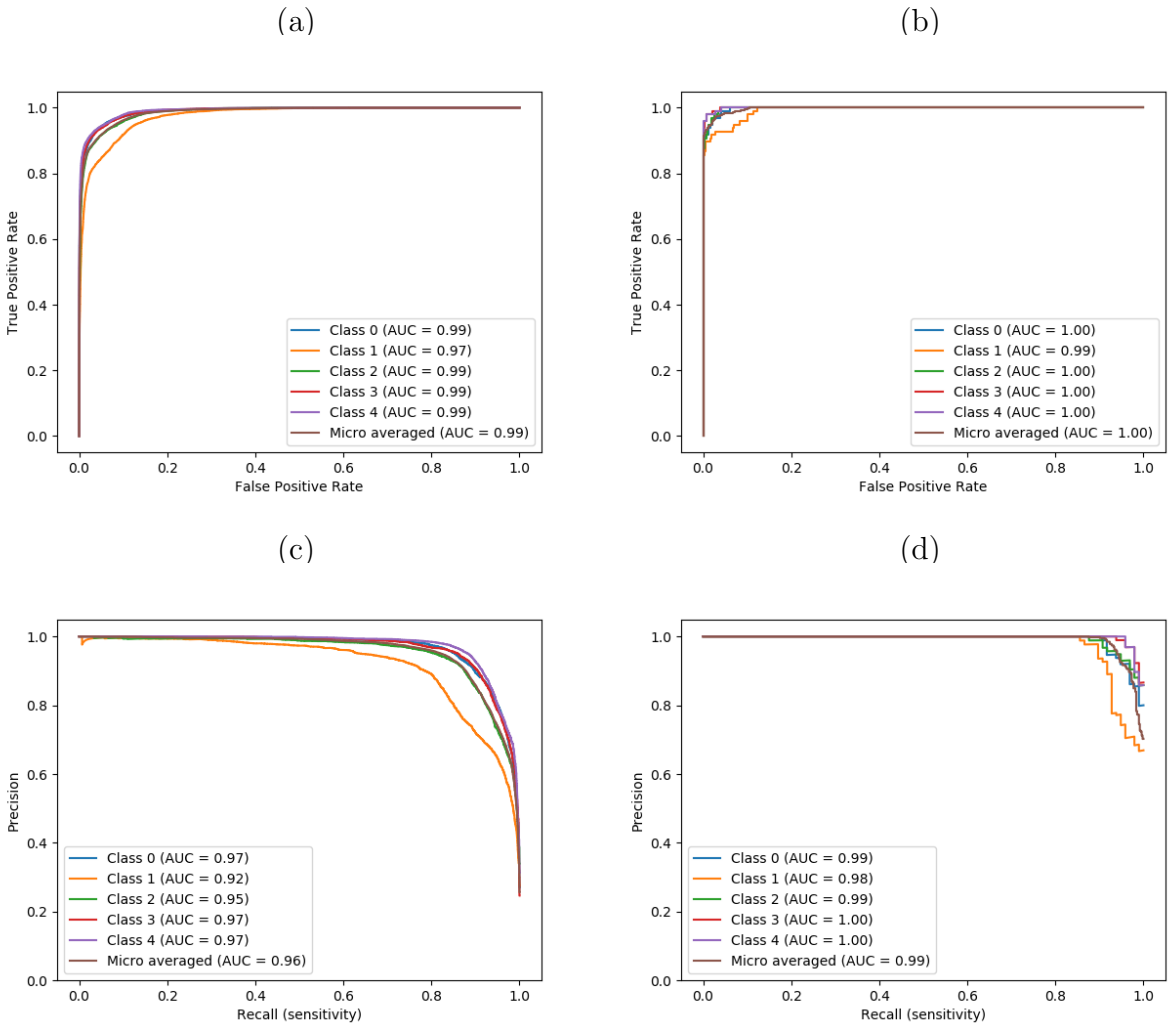

FIG. 14: Performance of LSTM for the Z24 bridge pier settlement (a) individual windows ROC, (b) voted series ROC, (c) individual windows PR, (d) voted series PR.

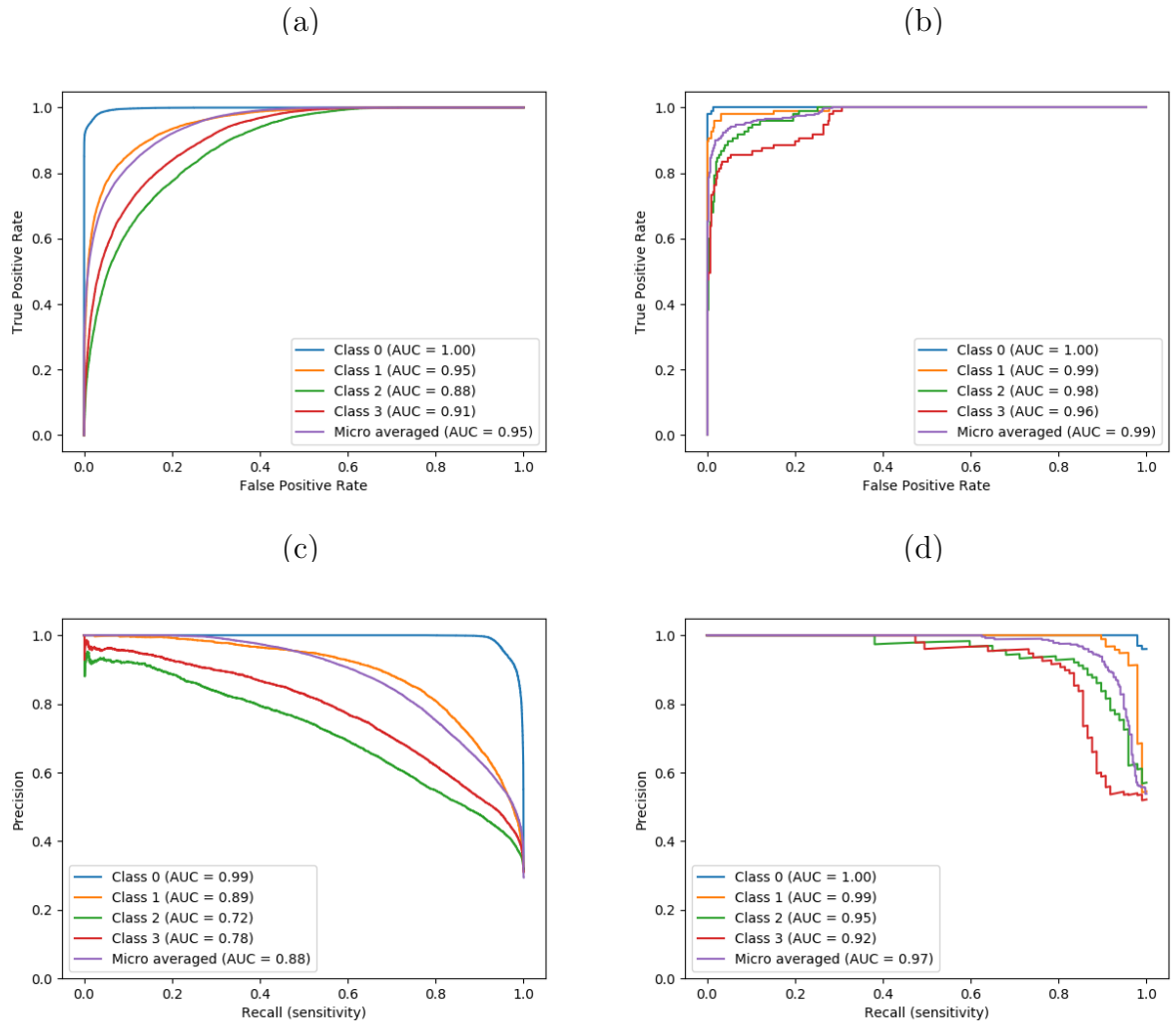

FIG. 15: Performance of LSTM by windowing for Z24 bridge rupture of tendons (a) individual windows ROC, (b) voted series ROC, (c) individual windows PR, (d) voted series PR.

Damage localization is performed using Algorithm 1, for multi-level and multiclass lowering of pier and rupture of tendons damage scenario. In these damage scenarios, the damage is not highly localized as in the experimental study of section 3.. Therefore, it is challenging to localize each measurement sensor and its location. Three different structural components of the bridge is used to localize damage and understand the effect of pier settlement. An undamaged pier (Utzenstorf), bridge deck, and damaged pier (koppigen) are used for representation of predicted probability (P_p) and infer damages in three components. The Koppigen pier is used for inducing the damage by lowering it in several increments starting

with 20 mm , 40 mm, 80 mm , and moving to 95 mm at the last stage. The P_p is plotted against the sensor number and a dash-dotted average of P_p of structural component is shown as a representation of combined P_p for corresponding structural component as shown in Fig. 16. For example, Fig. 16 (a, b, c) represents P_p for undamaged pier (UDP), bridge deck (BD), and damaged pier (DP) for 20 mm lowering of piers. Similarly, Fig. 16 (d, e, f) and (g, h, i) are for 40 mm and 85 mm lowering of piers, respectively.

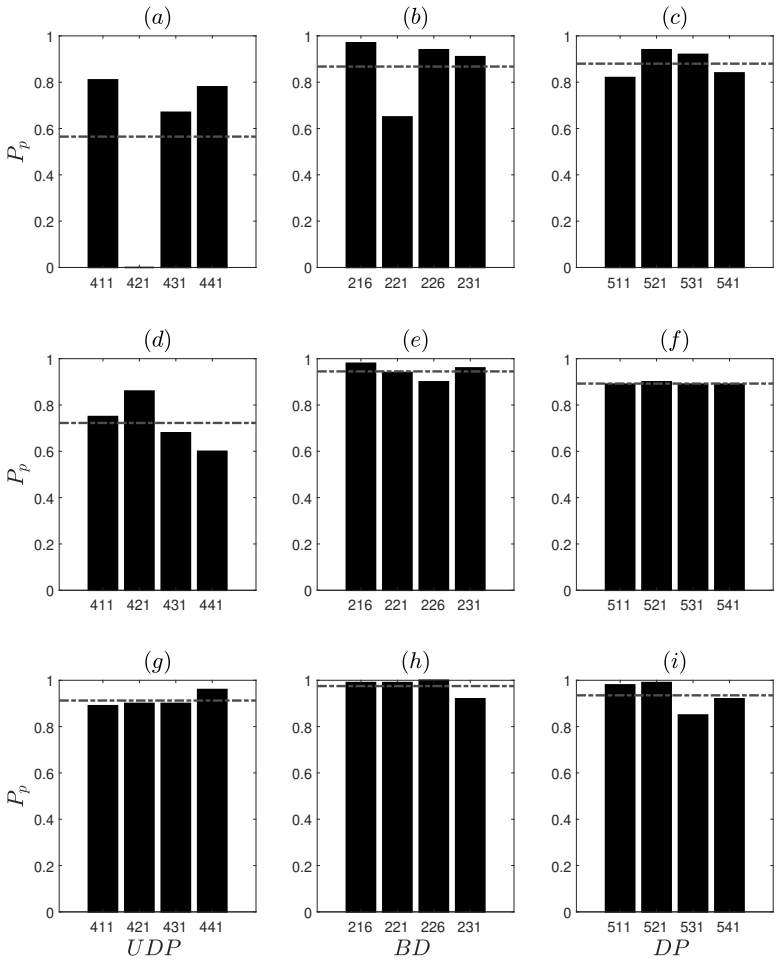

FIG. 16: Damage localization for lowering of pier for three damage levels, where, (a, b, c) are for 20 mm lowering of piers, (d, e, f) are for 40 mm lowering of piers, (g, h, i) are for 95 mm lowering of piers.

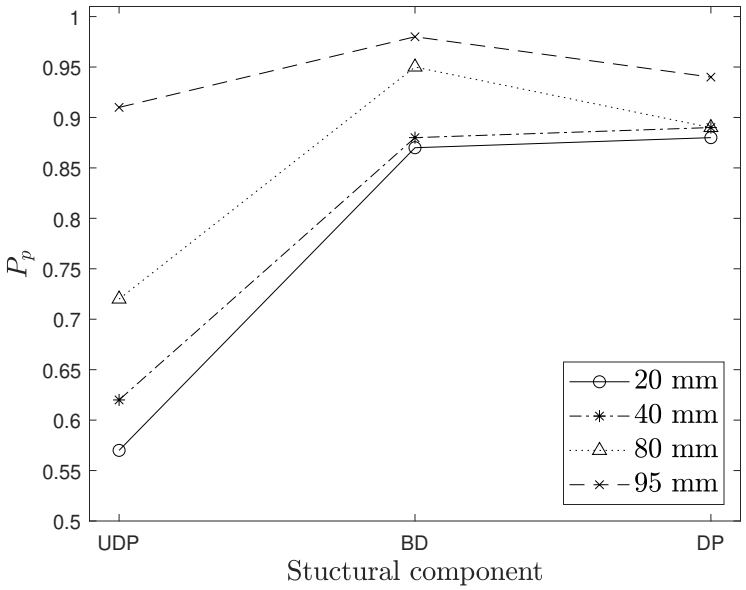

FIG. 17: Damage localization for lowering of pier, where legend shows the amount of pier-settlement.

Due to non-localization of measurement acquisition, it is difficult to infer damage location while considering each sensor separately. However, it is possible to compare average P_p of each structural component for various damage cases. The results considering average P_p for each structural component and various damage levels are shown in Fig. 17. Although there is no correlation between P_p and damage severity, however, as the severity increases, the signals becomes more distinguishable and LSTM learns the classification more effectively. It can be observed from Fig. 17 that UDP shows lowest predicted probability due to its similarity to undamaged-baseline signal, however, both BD, and DP shows higher prediction accuracy. The reason for bridge deck's highest probability of damage is attributed to the surface area and larger affect of differential pier-settlement in the whole structural system. The bridge suffers higher changes in structural responses (deflection, bending moment, shear) than at damaged pier itself, as it acted as a support. Similarly, for rupture of tendons, the most affected area would be the bridge deck and the damage induced due to rupture of tendons will create a non-localized and distributed damage throughout the bridge deck in comparison to

bridge piers. The damage localization per sensors is avoided due to non-conclusive inference and a comparison between structural components of the bridge is provided directly in Fig. 18. It can be observed that rupture of 6 tendons prove to be worse damage level scenario in comparison to 2 and 4 tendons.

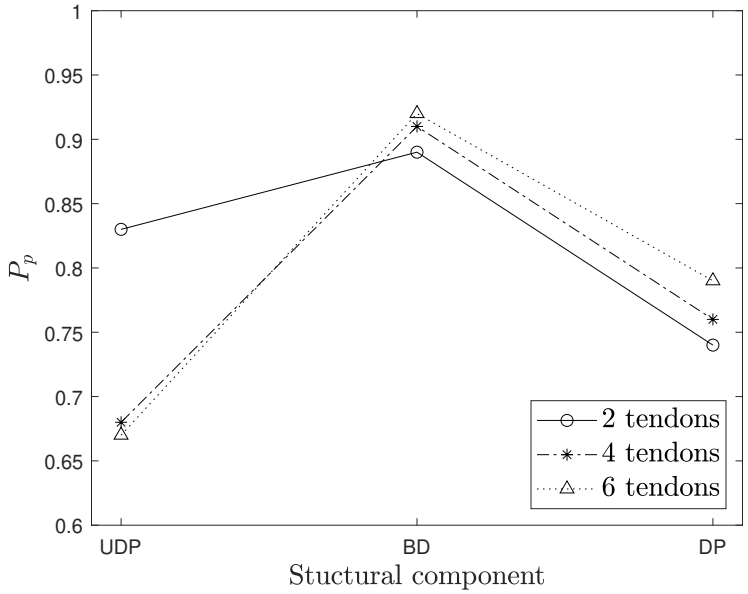

FIG. 18: Damage localization for rupture of tendons, where, the legend shows number of tendon ruptures.

4. CONCLUSION

In this paper, damage localization using a windowed-LSTM based deep learning algorithm is employed for multi-class, and multi-level damage detection. Various classes starting from binary to a maximum of five classes were classified into multi-class damage level. Limited dataset is augmented using windowing of the time-series measurements and the prediction accuracy is improved by novel voting approach on windowed classes. It is observed that the proposed algorithm performs well with non-localized and irregular sample sizes, and learns the long-term dependencies. The proposed algorithm is analyzed with sensitivity analysis on window-size as the external parameter to the model. A parametric study is also presented for random initialization of weights. The accuracy improvement of the proposed algorithm

is illustrated through a comparison between a single series dataset and windowed-voted for ROC and precision-recall AUC. In this paper, it is demonstrated that a simple LSTM architecture is capable of classifying the time series signals into multi-class and multi damage levels with high accuracy. The capability of LSTM models warrants the need to evaluate the remaining useful life of the structures which is a critical component of damage prognosis, that will be considered in a future study.

REFERENCE

1. Abdeljaber, O., Avci, O., Kiranyaz, M. S., Boashash, B., Sodano, H., and Inman, D. J. (2018). 1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data. Neurocomputing, 275, 1308–1317.

2. Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., and Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. Journal of Sound and Vibration, 388, 154–170.

3. Azimi, M., Eslamlou, A. D., and Pekcan, G. (2020). Data-driven structural health monitoring and damage detection through deep learning: State-ofthe- art review. Sensors (Switzerland) (Vol. 20).

4. Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. The Journal of Machine Learning Research, 13(1), 281-305.

5. Cawley, P. (2018). Structural health monitoring: Closing the gap between research and industrial deployment. Structural Health Monitoring, 17(5), 1225–1244.

6. Cha, Y-J., Choi, W., and Buyukozturk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361-378.

7. Chen, G. (2016). A gentle tutorial of recurrent neural network with error backpropagation. arXiv preprint arXiv:1610.02583.

8. Erazo, K., Sen, D., Nagarajaiah, S., and Sun, L. (2019). Vibration-based structural health monitoring under changing environmental conditions using Kalman filtering. Me-

567 chanical Systems and Signal Processing, 117(August 2018), 1–15.
568 9. Gatti, M. (2019). Structural health monitoring of an operational bridge: A case study.
569 Engineering Structures, 195(April 2017), 200–209.
570 10. Gulgec, N. S., Takáč, M., and Pakzad, S. N. (2017). Structural damage detection using
571 convolutional neural networks. Conference Proceedings of the Society for Experimental Me-
572 chanics Series, 3 Part F2(June), 331–337.
573 11. Guo, J., Xie, X., Bie, R., and Sun, L. (2014). Structural health monitoring by using
574 a sparse coding-based deep learning algorithm with wireless sensor networks. Personal and
575 Ubiquitous Computing, 18(8), 1977–1987.
576 12. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural compu-
577 tation, 9(8), 1735-1780.
578 13. Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network train-
579 ing by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
580 14. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J.
581 (2019). 1D Convolutional Neural Networks and Applications: A Survey, 1–20. Retrieved
582 from http://arxiv.org/abs/1905.03554
583 15. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with
584 deep 990 convolutional neural networks. Communications of the ACM, 60(6), 84–90.
585 16. Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning
586 applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.
587 17. Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge
588 systems using deep learning with Bayesian optimization. Computer-Aided Civil and Infras-
589 tructure Engineering, 34(5), 415–430.
590 18. Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural
591 networks for sequence learning. arXiv preprint arXiv:1506.00019.
592 19. Liu, T., Xu, H., Ragulskis, M., Cao, M., and Ostachowicz, W. (2020). A data-
593 driven damage identification framework based on transmissibility function datasets and one-

dimensional convolutional neural networks: Verification on a structural health monitoring benchmark structure. Sensors (Switzerland), 20(4), 1–25.

20. Maeck, J., and Roeck De. G. (2003), "Description of Z24 benchmark," *Mechanical Systems and Signal Processing*, 17(1), 127-131.

21. Ni, F. T., Zhang, J., and Noori, M. N. (2020). Deep learning for data anomaly detection and data compression of a long-span suspension bridge. Computer-Aided Civil and Infrastructure Engineering, 35(7), 685–700.

22. Okayasu, M., and Yamasaki, T. (2019). Structural Health Monitoring System for Remote Inspection of Material Failure. Journal of Nondestructive Evaluation, 38(2), 1–6.

23. Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., and Romero-Troncoso, R. J. (2019). Recurrent neural network model with Bayesian training and mutual information for response prediction of large buildings. Engineering Structures, 178, 603–615.

24. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., and Iyengar, S. S. (2019). A Survey on Deep Learning. ACM Computing Surveys, 51(5), 1–36.

25. Qarib, H., and Adeli, H. (2014). Recent advances in health monitoring of civil structures. Scientia Iranica, 21(6), 1–30.

26. Rafiei, M. H., and Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. Structural Design of Tall and Special Buildings, 26(18), 1–11.

27. Rafiei, M. H., and Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. Engineering Structures, 156(July 2017), 598–607.

28. Roeck, De. G., and Teughels, A. (2004), "Structural damage identification of the highway bridge Z24 by FE model updating," Journal of Sound and Vibration, 278, 589-610. 29. Salehi, H., and Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. Engineering Structures, 171, 170–189.

30. Sharma, S., and Sen, S. (2020). One-dimensional convolutional neural network-based damage detection in structural joints. Journal of Civil Structural Health Monitoring, (0123456789).

31. Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale 1079 Image Recognition. CoRR, abs/1409.1556.

32. Sirca, G. F., and Adeli, H. (2012). System identification in structural engineering. Scientia Iranica, 19(6), 1355–1364.

33. Sony, S. (2021). Towards Multiclass Damage Detection and Localization using Limited Vibration Measurements. PhD Thesis. https://doi.org/10.31237/osf.io/zbyue

34. Sony, S., and Sadhu, A. (2019). Identification of progressive damage in structures using time-frequency analysis. CSCE Annual Conference, Laval (Greater Montreal), June 12-15, Canada.

35. Sony, S., and Sadhu, A. (2020). Synchrosqueezing transform-based identification of time-varying structural systems using multi-sensor data. Journal of Sound and Vibration, 486, 115576.

36. Sony, S., and Sadhu, A. (2021). Multivariate empirical mode decomposition–based structural damage localization using limited sensors. Journal of Vibration and Control, 10775463211006965.

37. Sony, S., and Sadhu, A. (2020). Synchrosqueezing transform-based identification of time-varying structural systems using multi-sensor data. Journal of Sound and Vibration, 486, 115576.

38. Sony, S., Dunphy, K., Sadhu, S., and Capretz, M. (2021). A Systematic Review of Convolutional Neural Network-based Structural Condition Assessment Techniques. Engineering Structures, 226, 111347.

39. Sony, S., Laventure, S., and Sadhu, A. (2019). A literature review of next-generation smart sensing technology in structural health monitoring. Structural Control and Health Monitoring, 26(3), e2321.

40. Sun, L., Shang, Z., Xia, Y., Bhowmick, S., and Nagarajaiah, S. (2020). Review of Bridge

Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection. Journal of Structural Engineering, 146(5).

41. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2014). Going Deeper with Convolutions, arxiv:1409.4842.

42. Yang, R., Singh, S. K., Tavakkoli, M., Amiri, N., Yang, Y., Karami, M. A., and Rai, R. (2020). CNN-LSTM deep learning architecture for computer vision-based modal frequency detection. Mechanical Systems and Signal Processing, 144, 106885.

43. Ye, X. W., Jin, T., and Yun, C. B. (2019). A review on deep learning - based structural health monitoring of civil infrastructures. Smart Structures and Systems, 24(5), 567–586.

44. Ying, Y., Garrett, J. H., Oppenheim, I. J., Soibelman, L., Harley, J. B., Shi, J., and Jin, Y. (2013). Toward data-driven structural health monitoring: Application of machine learning and signal processing to damage detection. Journal of Computing in Civil Engineering, 27(6), 667–680.

45. Zeiler, M.D., and Fergus, R. (2013). Visualizing and understanding convolutional networks. arXiv:1311.2901v3.

46. Zhang, A., Wang, K. C. P., Li, B., Yang, E., Dai, X., Peng, Y., and Chen, C. (2017). Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. Computer-Aided Civil and Infrastructure Engineering, 32(10), 805–819.

47. Zhang, J., Cao, X., Xie, J., and Kou, P. (2019). An Improved Long Short-Term Memory Model for Dam Displacement Prediction. Mathematical Problems in Engineering, 2019.

48. Zhang, Y., Miyamori, Y., Mikami, S., and 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.

49. Zhao, X., Li, S., Su, H., Zhou, L., and Loh, K. J. (2018). Image-based comprehensive maintenance and inspection method for bridges using deep learning. ASME 2018 Conference on Smart Materials, Adaptive Structures and Intelligent Systems, SMASIS 2018, 2, 1–7.