2021

Bridge damage identification using deep learning-based Convolutional Neural Networks (CNNs)

Sandeep Sony
Western University, ssony@uwo.ca

Bridge damage identification using deep learning-based Convolutional Neural Networks (CNNs)

Sandeep Sony, PhD

ABSTRACT

In this paper, a novel method is proposed based on windowed-one dimensional convolutional neural network for multiclass damage detection using acceleration responses. The data is pre-processed and augmented by extracting samples of windows of the original acceleration time-series. 1D CNN is developed to classify the signals in multiple classes.

The damage is detected if the predicted classification is one of the indicated damage levels. The damage is quantified using the predicted class probabilities. Various signals from the accelerometers are provided as input to 1D CNN model, and the resulting class probabilities are used to identify the location of the damage. The proposed method is validated using Z24 bridge benchmark data for multiclass classification for two damage scenarios. The results show that the proposed 1D CNN methods performs with superior accuracy for severe damage cases and works well with different type of damage types.

KEYWORDS

Structural health monitoring; 1D CNN; Damage localization; Limited dataset, Data augmentation.

1. INTRODUCTION

In civil infrastructure, continuously increasing heavy traffic, unexpected natural calamities and human-made damages reduce their load-bearing capacity and service life. With ageing, the structures exhibit various damage signatures in several critical locations. In the absence of timely repair and maintenance, progressive damage leads to the collapse of structures. Despite the simplicity, the traditional manual inspection suffers difficulty while scanning the inaccessible areas in large structures such as bridges or tall buildings. Over the past few decades, structural health monitoring (SHM) has been a promising tool to supplement the knowledge of structural integrity over time. However, efficient diagnosis and prognosis of large-scale infrastructure require a reliable assessment of its damage under in-service conditions. In general, damage introduces localized discontinuities that can be captured by acquiring vibration measurements. SHM aims to provide suitable diagnostics and prognosis and assists infrastructure owners and decision-makers in maximizing the safety, serviceability, and functionality of critical structures. An autonomous SHM will allow efficient and cost-effective disaster management and lead to resilient infrastructure with faster recovery under natural disasters. In this paper, an autonomous multiclass damage identification method is proposed by utilizing artificial intelligence in the sequential SHM data, such as vibration measurements.

Data-driven damage diagnosis is a critical component of infrastructure asset management. Although there is a plethora of research on parametric methods based on time-frequency (TF) decomposition methods (Staszewski and Robertson 2006, Hu and Shao 2020, Barbosh et al. 2020, Sony and Sadhu 2020), non-parametric methods (Nakamura et al. 1998, Wang and Ong 2015, Abdeljaber and Avci 2016) have shown significant promises in data-driven SHM methods. Parametric methods include extracting dynamic parameters such as modal parameters and inferring the change in these parameters to detect any possible changes in the structures. On the other hand, non-parametric methods include extracting parameters that are estimated based on the computational models and the parameters are mathematically derived in a statistical sense. For the last couple of decades, the prominent work has been on parametric methods; however, non-parametric methods are not well-explored.

Structural damage identification can be considered as a pattern recognition-based non-parametric problem, which is divided into three stages, namely, data acquisition, feature extraction, and feature classification. With the transformation of SHM data analysis from the traditional method to a more advanced data-driven method, various machine learning

(ML) algorithms have been explored. However, the SHM community has prominently used supervised learning algorithms (Hou and Xia 2020, Avci et al. 2021). In (Gardner et al. 2020), the authors explained the interface between non-destructive evaluation and machine-learning-based SHM for damage detection. The study highlighted the need for a combination of compressive sensing based sparse methodology with data-driven machine learning methods to empower both non-destructive testing and SHM methodologies. Su et al. (2020) presented a critical review of field monitoring of high-rise structures. The study reviewed techniques for comfort assessment, seismic effect, wind effect, and temperature effect on monitoring of supertall structures. Recently, the SHM community has explored both vibration and image data for structural damage identification and localization. With advancements in artificial intelligence, image-based structural monitoring has garnered as a straightforward and inexpensive way to monitor large scale structures. Convolutional Neural Networks (CNNs) are the most used deep neural networks using imagery in the SHM community. Another indirect approach is to transform vibration measurements into TF domain, and the resulting TF images are used as input to 2D CNN. While image-based techniques remain popular and a viable method for SHM (Sony et al. 2021), they involve significant complexity in terms of obtaining a large amount of labelled data, pre-processing and classifying the images. Most of the structural condition assessment problems related to damage diagnosis are accumulative damages such as ongoing material deterioration or expansion of small cracks. Small damages are challenging to detect in images, and often environmental noise exacerbates the structural conditions. These properties make image-based methods unsuitable for real-time SHM applications that may run on mobile devices with limited computational capacity. As a solution, researchers have studied algorithms that directly operate on the vibration time signals.

Similar to images, researchers have also explored deep learning methods for effective damage detection using temporal information from other sequential data, such as acceleration measurements. Guo et al. (2014) proposed sparse coding as a feature extraction method for unlabeled acceleration measurements obtained from wireless sensors. The damage classification was carried out using a CNN, and the results were compared with 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 an 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. A CNN was used to classify damaged signals, and the proposed method achieved a testing error of 2% and showed robustness against environmental noise. Fallahian et al. (2018) explored the applicability of dynamic features such as mode shapes, frequency response functions, and natural frequencies as damage indicators under varying temperatures. The authors used a couple sparse coding and deep neural network as an ensemble method for damage detection and localization. The proposed method was validated on a numerical truss bridge and experimental I-40 benchmark dataset. Furthermore, the temporal acceleration data were converted into 2D contour maps and processed through CNN as images. Bao et al. (2019) proposed a CNN-based anomaly detection using acceleration measurements by converting them into grayscale images. The authors used several anomaly parameters such as missing, minor, outlier, square, drift, and trend data points to train the datasets using a stacked autoencoder architecture. Shang et al. (2020) proposed deep convolutional denoising autoencoders for structural damage detection. The proposed method extracted damage features from field measurements of undamaged structures under environmental noise.

Recently, 1D-CNN 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 authors trained the neural network on a vibration signal database obtained from a truss, 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 or for a multiclass damage scenario. Zhang et al. (2019) utilized the computational powers of 1D CNN to detect changes in structural parameters such as stiffness and mass. Three different structural components were used for data acquisition and model validation, namely, T-shaped steel beam, short and long steel girder bridge, and mean classification accuracy of 98% is achieved by the proposed methodology.

1D CNN was shown to be effective in identifying changes under a compressed dataset. Ni et al. (2019) showed the applicability of 1D CNN with autoencoders for anomaly detection under data compression. The study explored the possibility of using a compressed dataset, which is easy to handle for online real-time monitoring of structures. The proposed algorithm was validated using 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. A recent study by Azmi and Pekcan (2019) explored the concept of transfer learning in vibration measurements. Transfer learning is highly effective when used on similar infrastructure by training one type of structure and testing it on another kind of structure. The authors used a four-story IASC-ASCE SHM model for numerical training, and the proposed model was tested on experimental studies using IASC-ASCE SHM benchmark building and the Qatar University Grandstand Simulator with an accuracy of 90-100%. Recently, Sharma and Sen (2020) showed the applicability of 1D CNN for damage detection in structural steel frames. The study explored the applicability to localize the damage in building structures. Experimental validation was performed on a 2D-steel frame with different damage location and severity of the damage. The method was shown to identify different damage scenarios and the false-positive rate was also evaluated and found to be well within the acceptable limits. Furthermore, Liu et al. (2020) conducted a study by integrating traditional TF methods with the capability of neural networks. The authors used transmissibility function-based 1D CNN to effectively identify damage at the ASCE SHM benchmark structure. The proposed method was compared with time series and fast Fourier transform-based frequency-domain information; the TF signals exhibited more significant damage-sensitive features and stronger stability under excitation interference. 1D CNN primarily exhibited superior performance over artificial neural networks (ANNs) in the context of computation efficiency and noise insensitive for big data (Kiranyaz et al. (2019)). Recently, Bao et al. (2020) evaluated a combination of finite element method (FE) and 1D CNN for localizing damage for a jacket-type offshore structure. The proposed study achieved a high accuracy of 98%; however, the study lacks in two aspects; first, the data was generated synthetically using a finite element model, which might not resemble the actual real-world data with operational and environmental noise contamination. Second, the damage was induced artificially using the FE model, which was highly localized, and the data extracted was highly distinguishable from an undamaged structure that does not concur with the real-world data.

The proposed research explores the existing challenges of multiclass damage localization using 1D CNN. Unlike the simulated data, the real-world data is limited and is noise-contaminated, where multiclass damage localization becomes a significant challenge. To the best of the authors' knowledge, in this paper, 1D CNN is first time introduced for multiclass damage localization with varying damage severity under different damage scenarios. The issue of the limited dataset is solved by augmenting the data using windowing the acceleration measurements, and the classification results are improved using a novel voting approach on the prediction class. The study presents the benefits of using fast, computationally inexpensive 1D CNN with only one hidden layer for limited operational data for damage classification in a full-scale bridge.

The paper is structured as follows. A brief introduction of the structural damage identification, its need, and a literature review based on 1D CNN techniques are presented in section 1. Section 2 explains the theoretical background of the proposed algorithm, along with the selected architecture of 1D CNN. Furthermore, section 3 presents the capability of the proposed algorithms to identify and localize multiclass damage, the importance of hyperparameter tuning and various metrics used to show the damage parameters of the structures. The conclusions are presented in section 4.

2. PROPOSED METHODOLOGY

2.1 Background

Feedforward fully connected neural networks are the primary building blocks of various neural networks, where "fully connected" means all the neurons are connected to each neuron in the following layer. The objective of a feedforward network is to approximate some function f^*, for developing a classifier, where the model maps an input x to a category y. A feedforward network defines a mapping $y = f(x; \theta)$ and learns the value of a parameter θ that result in the best function approximation. The models are called feedforward due to the flow of information from input to the output through intermediate computations used to define the function f^* (Goodfellow et al. 2017). The input layer takes x as input and processes the computations through the hidden layer and outputs y. The hidden layers can be of any number, and if they are high in number, it is called deep fully connected network.

Convolutional Neural Networks (CNNs) are similar to fully connected with a difference of alternating convolutional, and pooling layers. 1D CNNs (Kiranyaz et al. 2019) are considered the *de facto* standard for various machine learning and speech recognition. 1D CNNs became popular in SHM since last decade due to its computational simplicity in comparison to its parent family of 2D and 3D CNNs as it requires simple array application. Due to its low computational cost, it is possible to train it with shallow networks and also on hand-held devices. 1D CNN configuration is formed by a combination of hidden layer and fully connected (generally, first and last layer) with a suitable filter size in each hidden layer, pooling layer, activation function and loss function. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable kernels, which have a small receptive field, defined by a width and height. The convolution process can be expressed as Eq. 1 (Goodfellow et al. 2017):

$$y(n) = x_i(n) \otimes h_i(m) \tag{1}$$

where, $x_i(n)$ is the input vector of length n and $h_i(m)$ is kernel of length m. The convolutional layers reduce the number of parameters needed for the kernels as all kernels share same spatial location.

Pooling is employed after the convolution layer to reduce the dimension of the convolution output. Pooling is used to reduce the dimensionality of a given mapping while highlighting the prominent feature and it also helps to reduce overfitting. Max pooling refers to picking up maximum value in a window of size f and this window is moved over the input with a stride of length s after each pooling operation. This layer features translational invariance with respect to the filter size. Let m be the size of filter, then the output is estimated as per Eq. 2:

$$M(x_i) = \max\left\{X_{i+k,i+l}||k| \leq \frac{m}{2}, |l| \leq \frac{m}{2}k, l \in \mathbb{N}\right\} \tag{2}$$

Activation functions can be either linear or non-linear. If the inner product of the input (x_i) to a neuron and it's weight (w^T) is denoted by n^* then output of the neuron is a some function of y. Non-linear activations enable the network to learn complex mappings as shown in Eq. 3 - 4.

$$n^* = w^T x_i \tag{3}$$

$$y = f(n^*) \tag{4}$$

Rectified Linear Unit (ReLU) is used as hidden layer activation function in this study which applies the non-saturating activation function. It effectively removes negative values from an activation map by setting them to zero. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

For the output layer, the choice of activation function depends on the type of output. For classification problems, SoftMax activations are preferred and for predictive/regression problems, ReLU is preferred. SoftMax function for n-class problem (representing n probabilities of input belonging to each of n-classes) as shown in Eq. 5:

$$P(\text{class}(i) = j|n^*) = \frac{e^{n_j^*}}{\sum_{k=1}^{n} e^{n_k^*}} \tag{5}$$

A typical 1D CNN architecture used in this study is shown in Fig. 1.

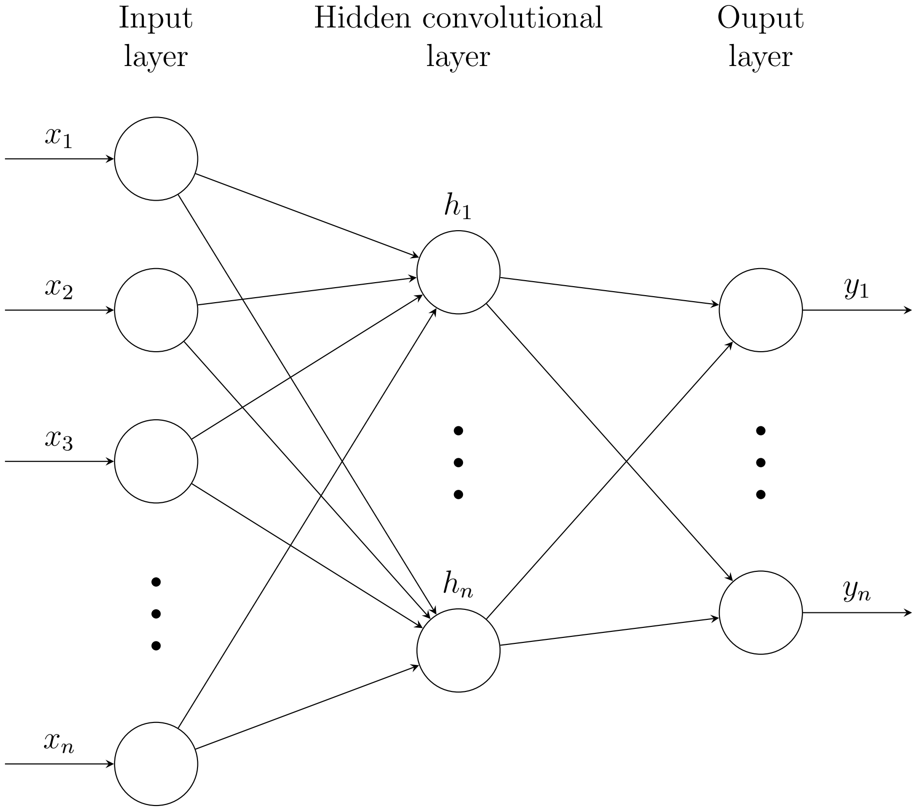


FIG. 1: A typical 1D CNN architecture.

2.2 Multiclass damage detection using windowed-voted 1D CNN

A method based on a 1D CNN is proposed to classify the vibration measurement into various multi-class damage levels. The proposed machine learning model for the multiclass damage classification is a multi-layer 1D CNN network architecture as shown in Fig. 2. Due to the scarcity of vibration-based multiclass data for civil infrastructure and the amount of data required for deep learning application, it is critical to augment the datasets per class. Assuming the availability of limited dataset used in this study and superior neural network performance, data is augmented by windowing the raw acceleration measurement. The various pre-processed sequence of windows are given as input to the model, and the softmax output of the final 1D CNN time step is considered as the set of classification probabilities $P(y = c_i)$ to each class c_i. During training, 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 a single time-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 in the time-series.

A single acceleration measurement is a record of long vibration data acquired using an accelerometer. The acquired measurement from various sensors attached to bridge structure is fit to a normal distribution during pre-processing stage before dividing them into several windows. The normal distribution is selected to improve the convergence rate of models trained on the datasets and prevents large value samples from dominating the input (Ioffe and Szegedy 2015). Next, the segment of the scaled time series is fed into a sequence of continuous windows of size $\mathbf{w}$), and a sequence of such windows of length L is arranged to form one input instance to the 1D CNN. Therefore, the input of the network is a $\mathbf{w}$-dimensional sequence of length L. Multiple such sequences are extracted from acceleration time-series, and each sequence is assigned to a label which is the damage level of the original time-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 which is, multiple sequences per time series, which also, in turn allows training machine learning models with less over-fitting.

In the proposed method, the window size $\mathbf{w}$ and the sequence length L (no. of windows in a sequence) is treated as hyperparameters that are tuned to improve the accuracy of 1D CNN. The hyperparameters include number of layers and number of nodes in each layer, activation function, and batch size for weight updates. Optimal parameters are obtained using a random search algorithm on a hyperparameter space (Bergstra and Bengio 2012). A training session is terminated when, either a specified maximum number of epochs is reached, or early stopping which is the validation loss does not decrease for a specified number of epochs. The final network weights are taken from the epoch with the smallest validation loss. The classification problem is presented as 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 1D CNN as the classification model.

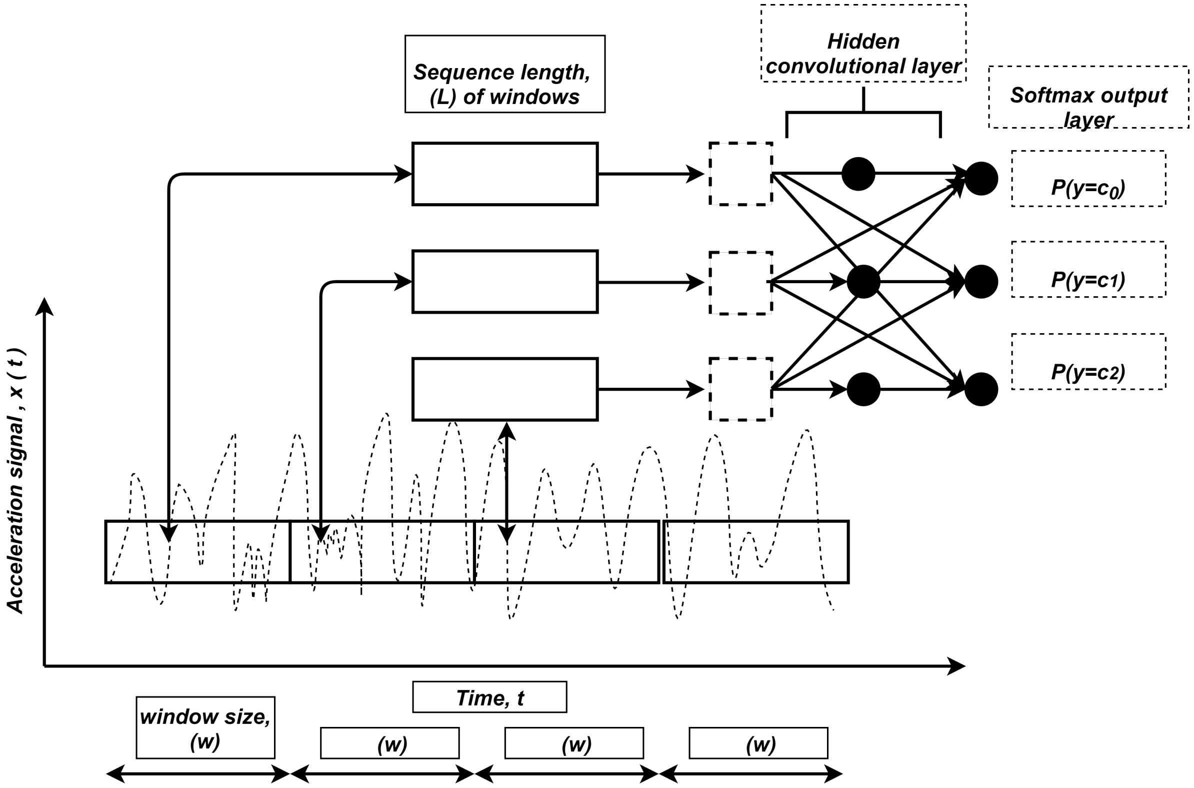

Hidden convolutional layer
Sequence length, (L) of windows
Softmax output layer
P(y=c0)
P(y=c1)
P(y=c2)
Acceleration signal , x (t)
Time, t
window size, (w)
(w)
(w)
(w)

FIG. 2: Extracting data sequences of windows from the vibration data using 1D-CNN architecture.

2.3 Performance criteria

In machine learning, a number of performance metrics are used to evaluate the efficacy of the computational model. These metrics measure different aspects of the obtained results. A brief description is provided below in the context on SHM of the civil infrastructure. The confusion matrix is a tabulation of classifications made by a model, typically with the actual class on rows and predicted class on columns. Table 1 shows the confusion matrix for a multi-class classification problem with three classes (α, β, and γ). As shown, TP_α is the number of true positive samples in class α, i.e., the number of samples that are correctly classified from class α, and $E_{\alpha\beta}$ is the samples from class α that are incorrectly classified as class β, i.e., misclassified samples. Therefore, the false negative in the α class (FN_α) is the sum of $E_{\alpha\beta}$ and $E_{\alpha\gamma}$ ($FN_\alpha = E_{\alpha\beta} + E_{\alpha\gamma}$) which indicate the sum of all class α samples that were incorrectly classified as class β or γ. Simply, FN of any class can be calculated by adding the errors in that class/column. Whereas the false positive for any predicted class which is located in a row represents the sum of all errors in that row. For example, the false positive in class α, (FP_α) is calculated as follows, $FP_\alpha = E_{\beta\alpha} + E_{\gamma\alpha}$. Therefore, for $k * k$ confusion matrix there are k correct classifications and $k^2 - k$ possible errors (Srinivasan 1999). There are various metrics that are derived from confusion metrics and are presented in Table 2.

TABLE 1: Confusion matrix for a multiclass problem.

	True class		
Predicted Class	TP_α	$E_{\beta\alpha}$	$E_{\gamma\alpha}$
	$E_{\alpha\beta}$	TP_β	$E_{\gamma\beta}$
	$E_{\alpha\gamma}$	$E_{\beta\gamma}$	TP_γ

In the context of SHM and multiclass damage detection, only, ROC-AUC, Accuracy, FNR and F1 score are used to evaluate the performance of the proposed method. Accuracy is a primary performance metric used to evaluate the efficacy of model to correctly classify the datasets into various class labels. Another important metric that has not been discussed in the literature is FNR. In SHM context, it is crucial to identify minor damage with minimal to no false negative alarm to prevent any future structure failure. In machine learning context, the FNR values should be at its minimum, while the accuracy, ROC-AUC,and F1 score should be at it maximum.

2.4 Damage localization

Damage localization for multi-class problems is evaluated using Algorithm 1. The whole structure is modeled as one experiment rather than modeling each sensor separately as in (Abdeljaber et al. 2017) and prediction probabilities are acquired for each sensor location. However, as there were multiple sensor locations covering the whole structure and in particular, three different structural components, namely, undamaged pier (UDP), bridge deck (BD), and damaged pier (DP). Only 12 sensor location, 4 each on UDP, BD, DP, respectively as chosen for damage localization purpose. The damage is confirmed if the true predicted probability class is equal to allocated class label for all cumulative windowed series for each sensor location.

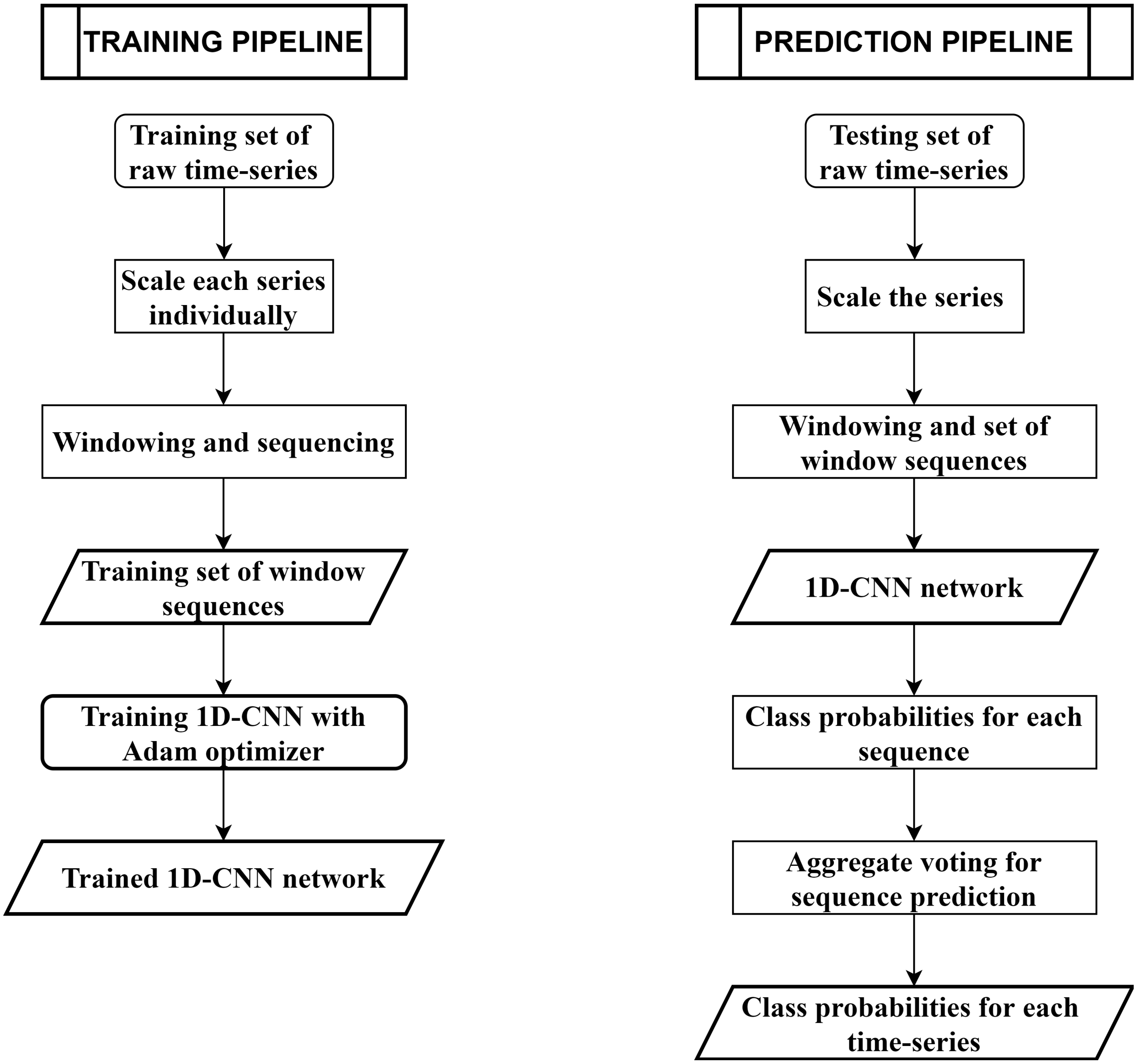

TRAINING PIPELINE
Training set of raw time-series
Scale each series individually
Windowing and sequencing
Training set of window sequences
Training 1D-CNN with Adam optimizer
Trained 1D-CNN network
PREDICTION PIPELINE
Testing set of raw time-series
Scale the series
Windowing and set of window sequences
1D-CNN network
Class probabilities for each sequence
Aggregate voting for sequence prediction
Class probabilities for each time-series

FIG. 3: Data pipelines for training the 1D CNN network and obtaining predictions for a given time-series.

```
Algorithm 1: Damage localization for multiclass damage localization
  Input: A signal $x(t)$
  Output: Prediction probabilities $P_p(y_c)$ for damage localization.
  (a) The acceleration data is pre-processed into multiple windows time-series and
      damage class-label is allocated to each windowed data.
  (b) The structure is modeled as whole as compared to per joint for computational
      efficiency and ease of modeling.
  (c) The windowed data is trained using 1DCNN using optimal parameters and
      tested on a separate dataset.
  (d) The probabilities of classification are obtained for each joint of every windowed
      series.
  (e) A damage is confirmed at the joint if true predicted probability class is equal to
      allocated class label.
```

TABLE 2: Description of various performance metrics.

Metric	Expression	Remarks
ROC-AUC	R Vs FPR	Degree of separability between classes
Accuracy (A)	$\frac{TP+TN}{TP+FN+FP+TN}$	Less useful for heavily imbalanced data
Precision (P)	$\frac{TP}{TP+FP}$	Positive predicted value
Recall (R)	$\frac{TP}{TP+FN}$	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

3. FULL-SCALE STUDY

3.1 Details of Z24 Bridge

Damage detection, where classification is more than two classes, is considered as a multiclass problem. In this study, two types of damage cases are used, namely, rupture of tendons, and pier settlement of a full-scale bridge, namely, Z24 Bridge. All the damage classes 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 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. 4. 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. During the demolition, the bridge data was acquired using 15 accelerometers placed at different spans of the bridge as shown in Fig. 5. The 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.

the data was acquired by performing various progressive damage scenarios. For the brevity of this study, only three different damage scenarios are considered: failure of anchor head, rupture of tendons, and pier settlement. Each damage scenario has multiple level of damage. All these damage scenarios are compared with the baseline undamaged state. It should be noted 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 damage cases. For example, failure of anchor head have two damage levels, rupture of tendons have three levels, and lowering of pier have four levels, and together they make a case of three separate damage classes. For detailed explanation of how the damages were induced in the bridge, the readers are suggested to refer (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 were damages were assigned classes starting from 1 to n depending upon the level of damage, as shown in Table 3. For example, in the case of rupture of tendons, the damage was induced at first, rupture of two tendons, and second, rupture of four tendons, third, rupture of six tendons, thereby creating three classes of damages for rupture of tendons . Similarly, there are four classes for lowering of pier.

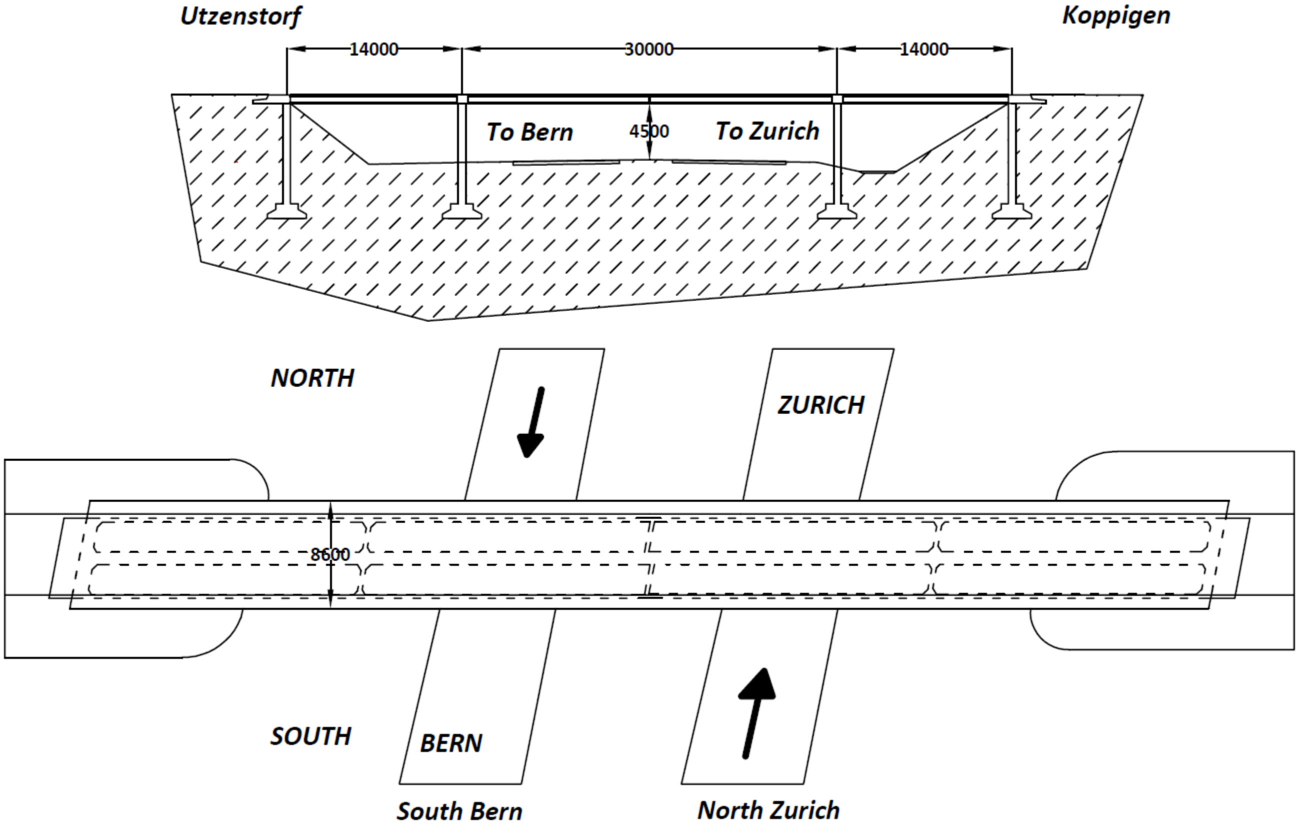


FIG. 4: Schematic of the Z24 bridge.

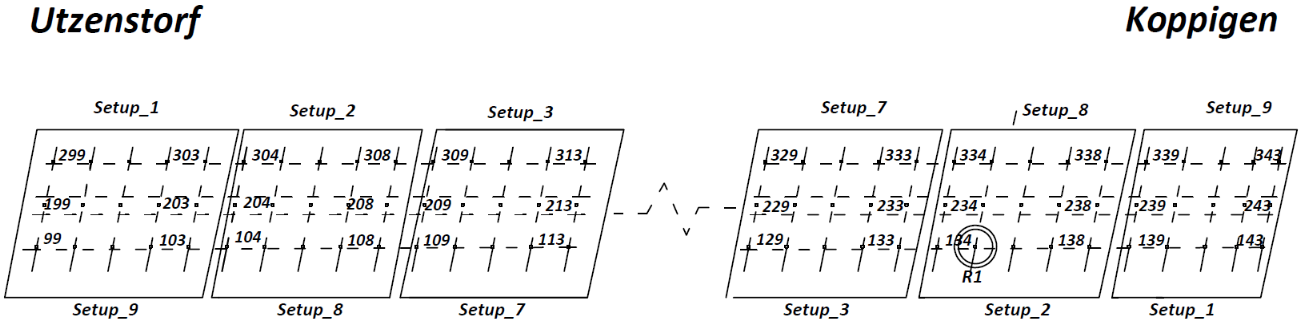


FIG. 5: Sensor placement for data acquisition.

TABLE 3: 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

3.2 Hyper-parameters for the 1D CNN model

An introduction to the full scale study based on various damage scenarios of Z24 Bridge is provided first, followed by hyperparameters used for the computational models are presented, the evaluation metrics based on the proposed method is described later with comparison between window-voted and non-voted results. In this study, a range of hyperparameters are selected first and tuned using random search algorithm to achieve a set of hyperparameter that provides the optimal accuracy. The range of hyperparameters used for 1DCNN are presented in Table. 4. For example, window size is adopted on a range between 64, 128, 160, 256, and 512 samples. Window size is the only external parameter used and is decided by the user. Thus, a sensitivity analysis is performed to understand the behavior of performance evaluation metrics (P_m) under different window sizes (w). Two different metrics are used for sensitivity analysis which are accuracy and FN as they represents overall accuracy of the model and false-negative alarm critical for civil infrastructure.

TABLE 4: Hyperparameter used in 1D CNN for tuning by random search algorithm.

Parameter	Values
Window size	64, 128, 160, 256, 512
No. of hidden convolutional layers	1 - 6
No. of filters	1024, 512, 256, 128, 64, 32
No. of fully connected layers	1 or 2 layers with 16 and 32 nodes
Learning rate	0.0003, 0.001, 0.01
Batch size	64, 256, 512
Kernel size	8, 16, 32, 64

The optimal hyperparamters for the Z24 bridge dataset are obtained after tuning and are presented for all the models in Table 5. An analysis is performed to understand the effect of w versus P_m. The results are shown for various damage cases in Fig. 6. For example, Fig. 6 (a-b), shows that the optimal performance is achieved at w=256, with highest ROC and accuracy, and lowest false-negative. Although, the FNR remains consistent after w=512 and other metrics are at their peak, however, due to larger w, the data size reduces per damage class and it leads to over-fitting of the data.

TABLE 5: Optimal configuration of 1D CNN hyperparameters.

Parameter	Values
Window size	256
No. of hidden convolutional layers	1
No. of filters	32
No. of fully connected layers	2 with 32 and 16 nodes, respectively
Learning rate	0.0003
Batch size	256
Kernel size	16

3.3 Random initialization of weights

Deep learning algorithms are iterative and require the user to specify value of initial weights of neurons to initiate the iteration and its optimization. In practice, all weights in the model are randomly drawn from a Gaussian or uniform distribution. The choice of Gaussian or uniform distribution does not seem to matter much but has not been exhaustively studied (Goodfellow et al. 2016). However, the scale (low or high magnitude) have a large effect on both the outcome and optimization procedure. In this study, random initialization with early stopping criteria is used and Adam optimizer (Kingma and Ba, 2014) is used with dropout in each layer for regularization. After acquiring the optimal tuned parameters, a parametric study is conducted to understand variance in the metrics of 1D CNN model for random initialization of weights. The metrics used for evaluating random initialization of weights are ROC-AUC, accuracy, FNR, and F1 score and are shown in Table. 6. It can be observed that for pier settlement, the mean (μ) of ROC-AUC is 0.97 with an accuracy of 0.85. The FNR is 0.15 and the standard deviation (σ) for all the trials is at its minimal of 1%. Similarly, for rupture of tendons, the ROC-AUC is 0.92 with an accuracy of 0.67 and FNR of 0.33 with minimal σ of 2%.

TABLE 6: Random initialization of weights

Pier settlement				
Trials #	ROC-AUC	Accuracy	FNR	F1 score
1	0.98	0.85	0.15	0.85
2	0.97	0.85	0.15	0.85
3	0.98	0.86	0.14	0.86
4	0.97	0.83	0.17	0.83
5	0.98	0.86	0.14	0.86
μ	0.97	0.85	0.15	0.85
σ	0.00	0.01	0.01	0.01
Rupture of tendons				
1	0.92	0.69	0.31	0.69
2	0.93	0.68	0.32	0.68
3	0.90	0.66	0.34	0.66
4	0.91	0.65	0.35	0.65
5	0.92	0.66	0.34	0.66
μ	0.92	0.67	0.33	0.67
σ	0.01	0.02	0.02	0.02

3.4 Effect of window size

The window size used to augment the data is an external parameter apart from other model parameters and it is critical to understand the effect on model performance. It can be observed that the best performance with a combination of maximum ROC-AUC, accuracy and minimum FNR is achieved at 256 samples per window. It is shown in Fig. 6 (a), ROC-AUC increases to 1.0 at 512, 800, 1024 samples per window, however, it leads to over fitting with increased FNR. A similar result can be observed from Fig. 6 (b) with optimal performance at 256 samples per window.

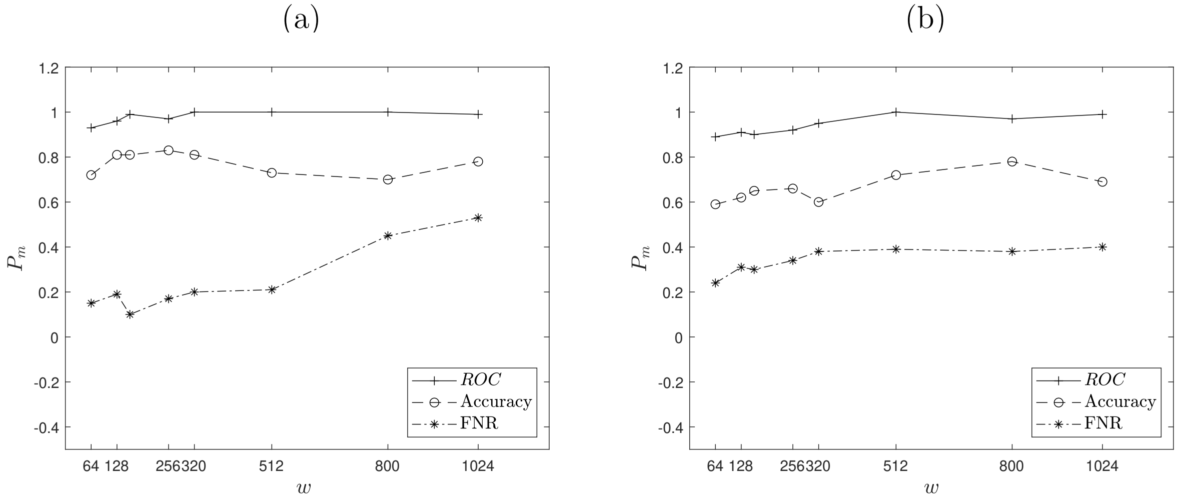


FIG. 6: Performance evaluation of 1D CNN based on window size for (a) pier settlement and (b) rupture of tendons.

3.5 Model performance

The optimal parameters are used to evaluate the performance of proposed model on full-series versus voted-windowed samples. The reason for comparison of the full-series and windows voted is to show the improved performance by voting the windowed samples. It is observed that voting on windowed dataset increases accuracy considerably and it exhibits in ROC-AUC and precision-recall (PR)-AUC curves, as presented in Fig. 7, and 8, 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 augmenting the data samples per class.

It should also be noted that the accuracy in case of pier-settlement is 0.83 and it reduced to 0.66 for rupture of tendons as shown in Table. 7. It can be observed that the FNR increased from 0.17 to 0.34 in case of pier-settlement and rupture of tendons, respectively. The label 0, 1, and 2 are used to denote, training, full-series test set, and windowed-series test set performance metrics, respectively.

TABLE 7: Training and testing performance of 1D CNN.

Lowering of pier								
Dataset	ROC	PR	A	P	R	FPR	FNR	F1
0	0.96	0.88	0.80	0.80	0.80	0.05	0.20	0.80
1	0.95	0.84	0.77	0.77	0.77	0.06	0.23	0.77
2	0.97	0.91	0.83	0.83	0.83	0.04	0.17	0.83
Rupture of tendons								
Dataset	ROC	PR	A	P	R	FPR	FNR	F1
0	0.89	0.75	0.63	0.63	0.63	0.12	0.37	0.63
1	0.87	0.71	0.59	0.59	0.59	0.14	0.41	0.59
2	0.92	0.82	0.66	0.66	0.66	0.11	0.34	0.66

As shown in Fig. 7, voting on samples have improved the AUC for both ROC and precision-recall. It can be observed that in case of pier-settlement, there is meager increase on ROC-AUC, however, there is a considerable improvement in the area under the curve for PR. This behaviour can be attributed to a more localized damage in case of pier settlement. Moreover, as observed in Fig. 8, where the damage was considerably distributed in case of rupture of tendons voting on windows highly increased the PR area under the curve for rupture of tendons. Whereas, ROC-AUC and PR increased by 5.75% and 15.5%, respectively. It should be noted that micro-averaged is the average of area under the curve for all the classes.

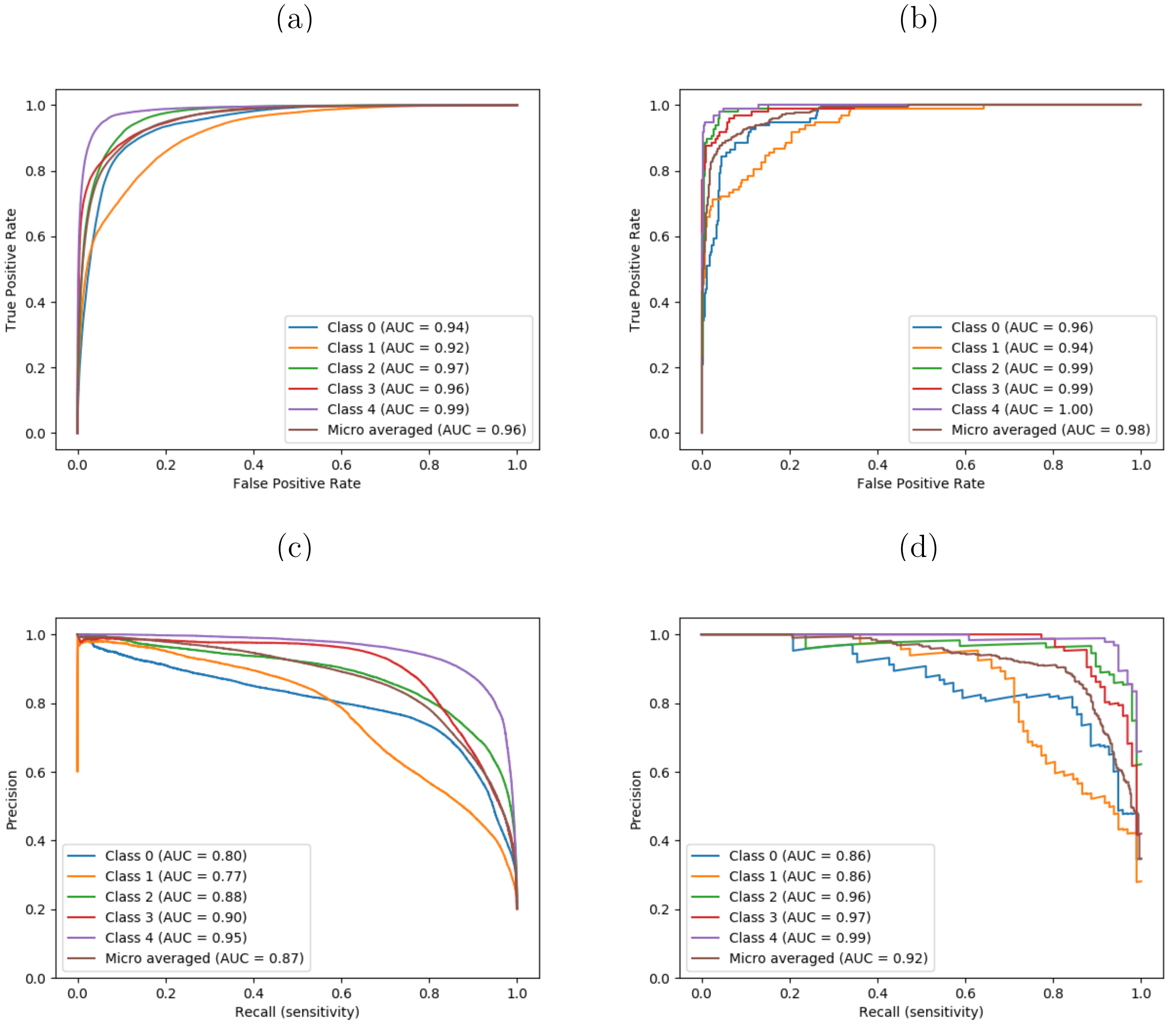
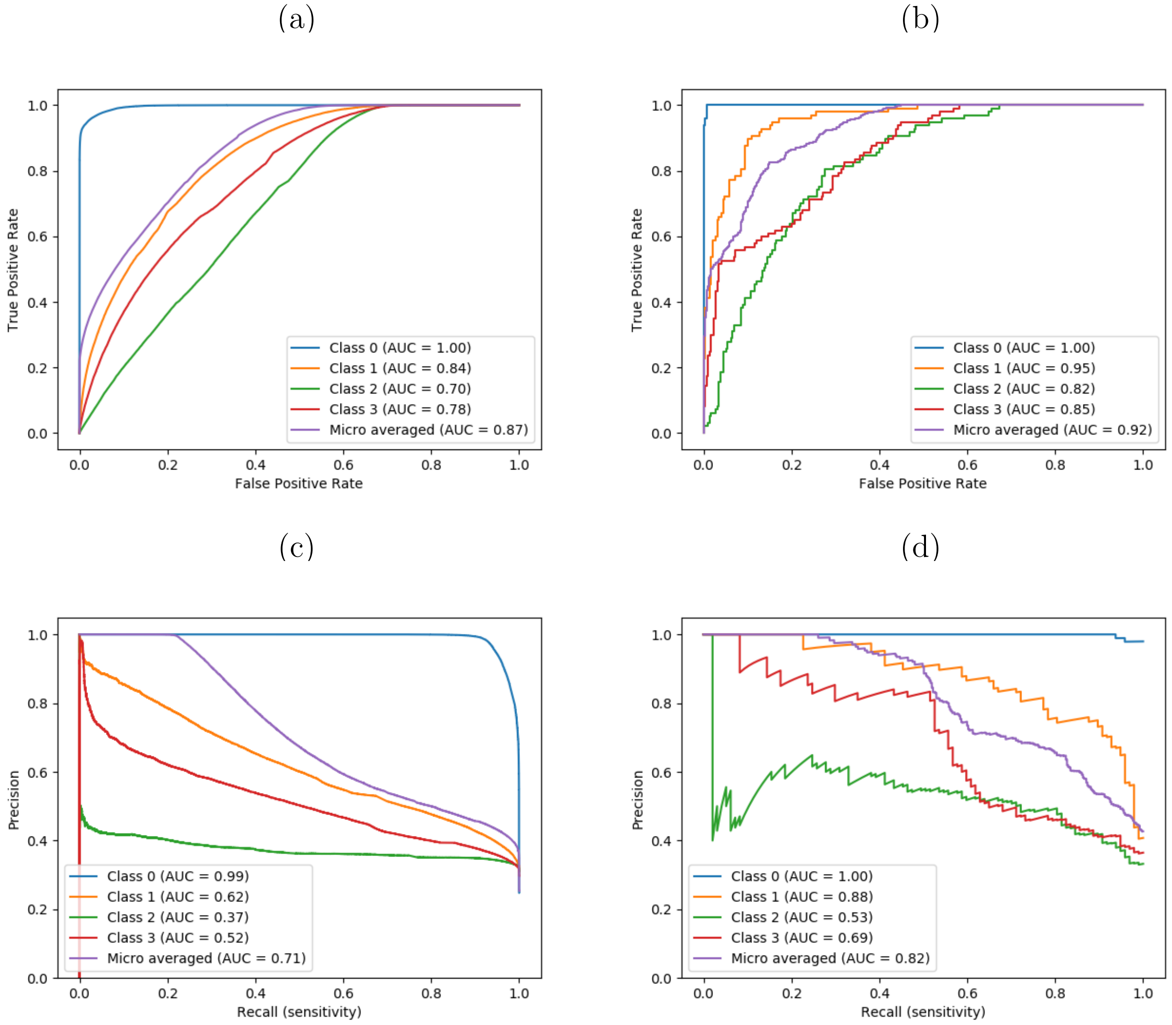


FIG. 7: Performance of 1DCNN by windowing for the Z24 bridge pier settlement (a) series-ROC, (b) windowed-voted ROC, (c) series-PR, (d) windowed-voted PR.


(a)

(b)

(c)

(d)

FIG. 8: Performance of 1DCNN by windowing for Z24 bridge rupture of tendons (a) series-ROC, (b) windowed-voted ROC, (c) series-PR, (d) windowed-voted PR.

3.6 Damage localization

Damage localization is performed using Algorithm 1, for two multiclass damage scenario, namely, lowering of pier and rupture of tendons. The sensor locations are identified first, then, three different structural components of the bridge are used to localize damage and understand the effect of pier settlement and rupture of tendons. 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. Twelve different sensors are used to identify the location of damage, namely, 4 sensors (411, 421, 431, 441) on undamaged pier (UP), 4 sensos (216, 221, 226, 231) on bridge deck (BD), and 4 sensors (511, 521, 531, 541) on damaged pier (DP) as shown in Fig. 9.

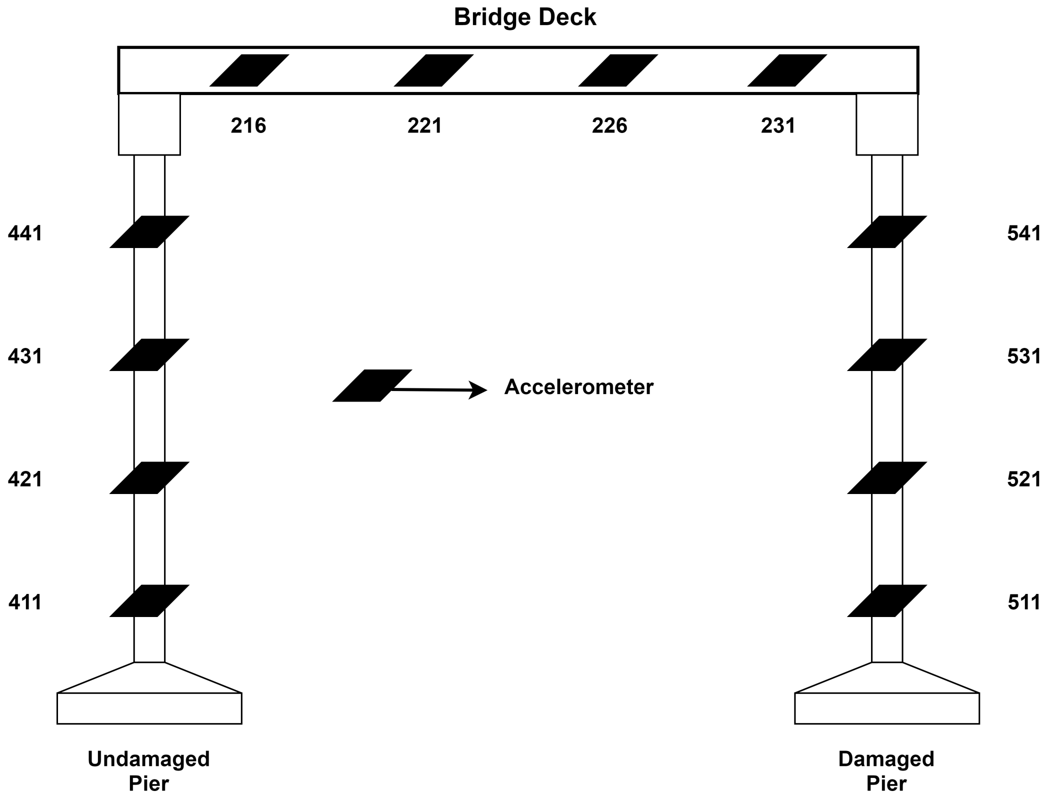


FIG. 9: Schematic showing sensor location and its numbers used in the analysis.

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. 10 for 20 mm, and 40 mm and Fig. 11 for 80 mm, and 95 mm lowering of pier, respectively. For example, Fig. 10 (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. 10 (d, e, f) is for 40 mm lowering of piers, respectively. It can be observed that proposed algorithm does not provide conclusive evidence for nominal damage of 20 mm, however shows some evidence of damaged pier. However, Fig. 11 (a, b, c) and (d, e, f) shows localization of damage for 80 mm and 95 mm, and it can be observed that the localization is clearly identified where the P_p is highest for damaged pier followed by bridge deck which is affected by differential settlement of one of the piers.

Although there is no correlation between P_p and damage severity, however, as the severity increases, the signals becomes more distinguishable and 1D CNN learns the classification more effectively. It can be observed from Fig. 12 that UDP shows lowest predicted probability due to its similarity to the response of the undamaged structure, however, both BD, and DP shows higher prediction accuracy. The reason for bridge deck's highest probability 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.

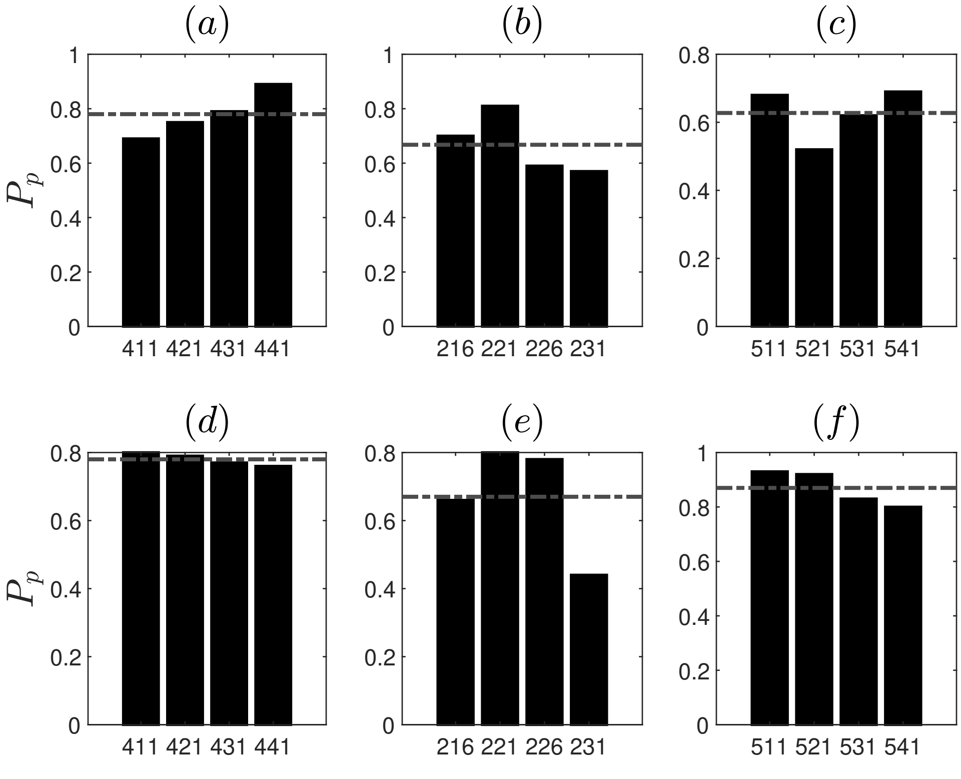


FIG. 10: Damage localization for lowering of pier for two damage levels, where, (a, b, c) are for 20 mm lowering of piers, and (d, e, f) are for 40 mm lowering of piers.

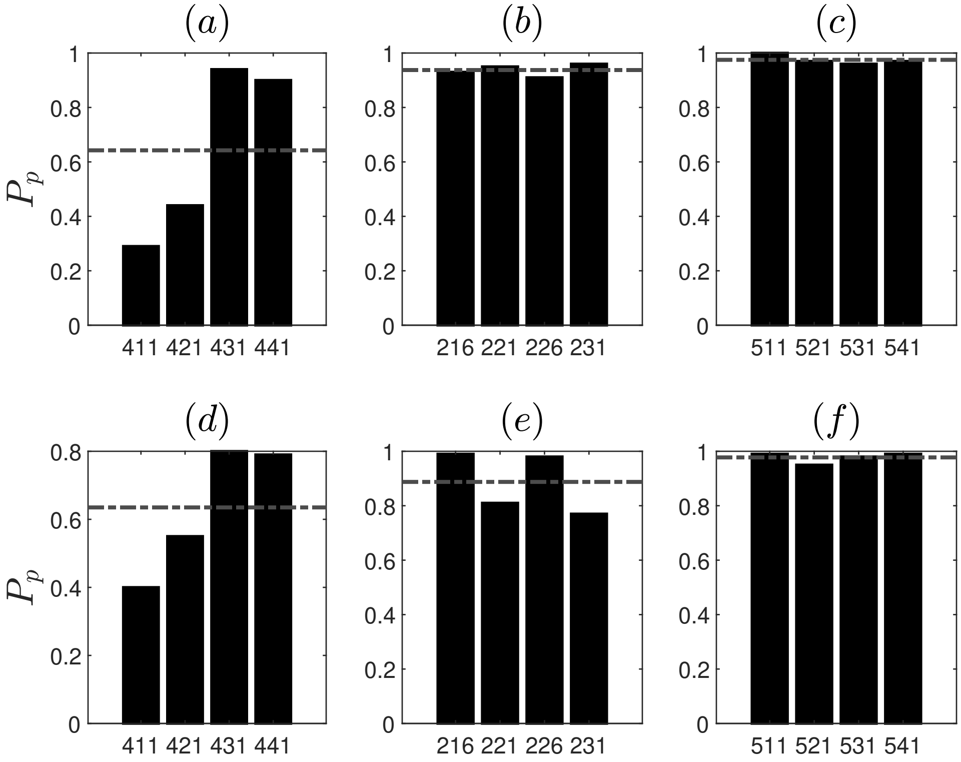


FIG. 11: Damage localization for lowering of pier for three damage levels, where, (a, b, c) are for 80 mm lowering of piers, and (d, e, f) are for 95 mm lowering of piers.

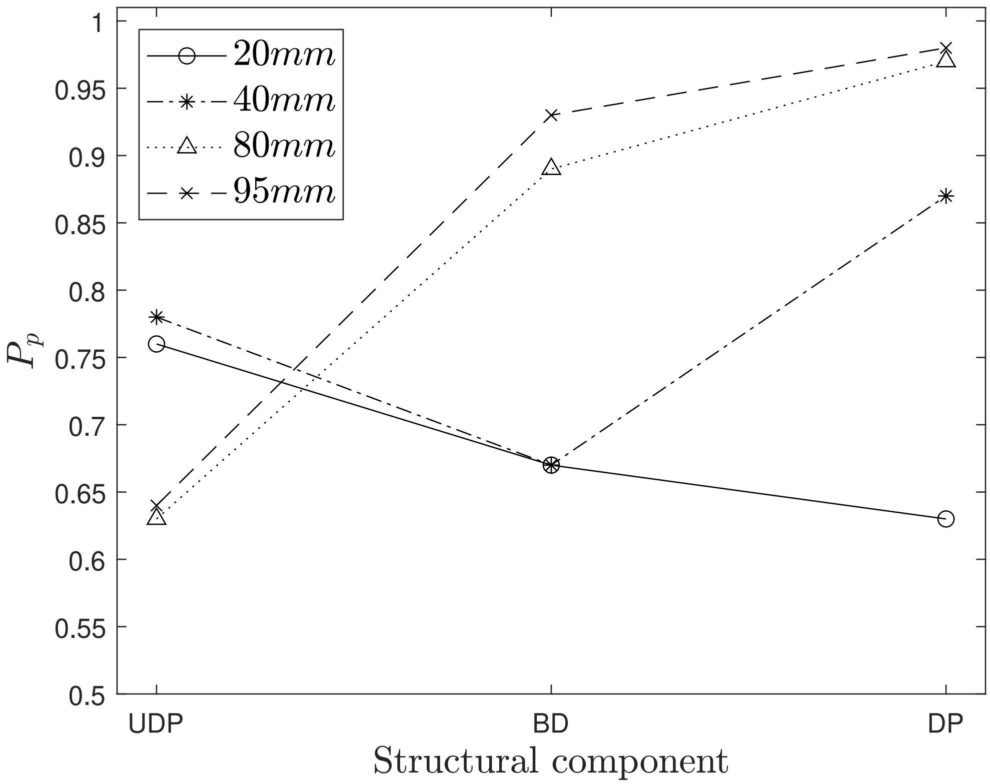


FIG. 12: Damage localization for lowering of pier.

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 the 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. 13. It should be noted that rupture of tendons affects bridge deck highly and it is shown in Fig. 13, however, the proposed algorithm could not clearly show the affect of rupture of 2 and 4 tendons while rupture of 6 tendons prove to be worse damage level scenario.

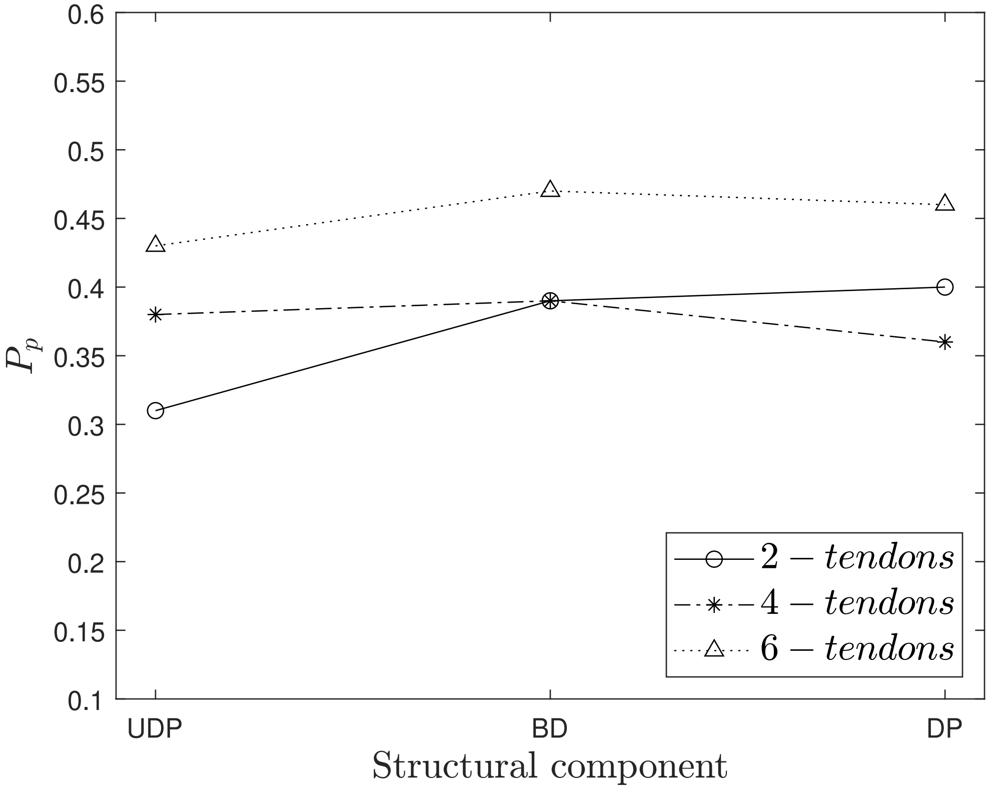


FIG. 13: Damage localization for rupture of tendons.

5. CONCLUSIONS

In this paper, damage localization using a windowed-1D CNN is employed for multi-class, and multi-level damage detection. Limited dataset is augmented using windowing of the time-series measurements and the prediction accuracy is improved by a novel voting approach on windowed classes. It is observed that due to non-localization of sensors for data acquisition, damage localization for lower level of damage scenario is challenging to predict. However, this improves the severity of the damage. 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 work, it is demonstrated that a simple 1D CNN architecture with only one hidden layer is capable of classifying the time series signals into multi-class with high accuracy. The future work is reserved to improve the algorithm to identify minor level of damage with superior performance.

ACKNOWLEDGMENTS

The author would like to acknowledge funding provided to the first author through Government of Ontario, Canada as Ontario Graduate Scholarship (OGS), and Queen Elizabeth II Graduate Scholarship in Science and Technology (QEGS-II) for PhD studies.

REFERENCES

1. Abdeljaber, O., and Avci, O. (2016). Non-parametric structural damage detection algorithm for ambient vibration response: utilizing artificial neural networks and self-organizing maps. Journal of Architectural Engineering, ASCE, 22(2), 04016004.

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. Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., Inman, D J. (2021). A review of vibration-based damage detection in civil engineering structures: from traditional methods to machine learning and deep learning based applications, 147, 107077.

4. Azimi, M., and Pekcan, G. (2020). Structural health monitoring using extremely compressed data through deep learning. Computer-Aided Civil and Infrastructure Engineering, 35(6), 597-614.

5. Bao, X., Fan, T., Shi, C., and Yang, G. (2020). One-dimentional convolutional neural network for damage detection of jacket-type offshore platforms. Ocean Engineering, 108293.

6. Bao, Y., Tang, Z., Li, H., and Zhang, Y. (2019). Computer vision and deep learning-based data anomaly detection method for structural health monitoring. Structural Health Monitoring, 18(2), 401-421.

7. Barbosh, M., Singh, P., and Sadhu, A. (2020). Emperical mode decomposition and its variants: a review with applications in structural health monitoring, Smart Materials and Structures, 29(9), 093001.

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

9. Fallahian, M., Khoshnoudian, F., and Meruane, V. (2018). Ensemble classification method for structural damage assessment under varying temperature. Structural Health Monitoring, 17(4), 747-762.

10. Gardner, P., Fuentes, R., Dervilis, N., Mineo, C., Pierce, S.G., Cross, E.J, and Wor-

den, K. (2020). Machine learning at the interface of structural health monitoring and non-destructive evaluation, Philosophical Transactions of Royal Society, A378: 20190581.

11. Goodfellow, I., Bengio, Y., and Courville, A. (2017). Deep learning. Cambridge, MA: MIT Press.

12. Gulgec, N. S., Takáč, M., and Pakzad, S. N. (2017). Structural damage detection using convolutional neural networks. Conference Proceedings of the Society for Experimental Mechanics Series, 3 Part F2(June), 331–337.

13. Guo, J., Xie, X., Bie, R., and Sun, L. (2014). Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks. Personal and Ubiquitous Computing, 18(8), 1977-1987.

14. Hou, R., and Xia, Y. (2020). Review on the new development of vibration-based damage identification for civil engineering structures: 2010-2019, Journal of Sound and Vibration, 115741.

15. Ioffe, S., and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

16. Kankanamge, Y., Hu, Y., and Shao, X. (2020). Application of wavelet transform in structural health monitoring, Earthquake Engineering and Engineering Vibrations, 19, 515-532.

17. Kingma, D.P., and Ba, J.L. (2017). ADAM: a method for stochastic optimization. 3rd International Conference for Learning Representations, San Diego, arXiv:1412.6980v9.

18. Liu, T., Xu, H., Ragulskis, M., Cao, M., and Ostachowicz, W. (2020). A data-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.

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

20. Nakamura, M, Masri S. F., Chassiakos, A.G., and Caughey T. K. (1998). Amethod for

non-parametric damage detection through the use of neural networks. Earthquake Engineering and Structural Dynamics, 27, 997-1010.

21. 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.

22. Shang, Z., Sun, L., Xia, Y., and Zhang, W. (2020). Vibration-based damage detection for bridges by deep convolutional denoising autoencoder. Structural Health Monitoring, (online first).

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

24. 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.

25. Sony, S., Dunphy, K., Sadhu, S., and Capretz, M. (2020). A Systematic Review of Convolutional Neural Network-based Structural Condition Assessment Techniques. Engineering Structures. (Accepted).

26. Srinivasan, A. (1999). Note on the location of optimal classifiers in n-dimentional roc space. Technical Report PRG-TR-2-99, Oxford University Computing Laboratory, England.

27. Staszewski., W. J., and Robertson, A.N. (2006). Time–frequency and time–scale analyses for structural health monitoring, Philosophical Transactions of Royal Society, A365, 1851.

28. Su, J., Xia, Y., and Weng, S. (2020). Review on field monitoring of high-rise structures. Structural Control and Health Monitoring, e2629.

29. Wang, V. Z., and Ong, K.C.G. (2015). Nonparamteric statistical formulations for structural health monitoring. Computers and Structures. 148, 63-74.

30. 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.