Journal of Southeast University (English Edition) Vol. 40, No. 3, pp. 221 – 229 Sept. 2024 ISSN 1003—7985

Bridge damage identification based on convolutional autoencoders and extreme gradient boosting trees

Duan Yuanfeng Duan Zhengteng Zhang Hongmei Cheng J. J. Roger

(College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China)

Abstract: To enhance the accuracy and efficiency of bridge damage identification, a novel data-driven damage identification method was proposed. First, convolutional autoencoder (CAE) was used to extract key features from the acceleration signal of the bridge structure through data reconstruction. The extreme gradient boosting tree (XGBoost) was then used to perform analysis on the feature data to achieve damage detection with high accuracy and high performance. The proposed method was applied in a numerical simulation study on a three-span continuous girder and further validated experimentally on a scaled model of a cable-stayed bridge. The numerical simulation results show that the identification errors remain within 2.9% for six single-damage cases and within 3.1% for four double-damage cases. The experimental validation results demonstrate that when the tension in a single cable of the cable-stayed bridge decreases by 20%, the method accurately identifies damage at different cable locations using only sensors installed on the main girder, achieving identification accuracies above 95.8% in all cases. The proposed method shows high identification accuracy and generalization ability across various damage scenarios.

Key words: structural health monitoring; damage identification; convolutional autoencoder (CAE); extreme gradient boosting tree (XGBoost); machine learning

DOI: 10.3969/j. issn. 1003 – 7985. 2024. 03. 001

Bridges represent an indispensable component of transportation infrastructures. Yet, they are subject to progressive degradation over time because of various environmental factors. Furthermore, the volume of traffic and the load that bridges support tend to escalate over the years. Given these circumstances, implementing continuous monitoring practices becomes imperative. Such measures are crucial for guaranteeing the enduring safety and structural integrity of these vital structures[1–2].

With the development of technology and the advancement of data collection methods, structural health monitoring (SHM) technology has become a key tool for bridge maintenance and management. These technologies can not only provide real-time data reflecting the current status of bridges but also identify potential damage to structures using data analysis and other methods[3].

Traditional methods for identifying structural damage in bridges have often been divided into two main categories: those based on theoretical models and those relying on data analysis[4]. The former approach, known as model-based damage detection, employs physical parameters, such as vibrational frequencies, mode shapes, and damping coefficients, to mathematical models to assess the health of a structure, pinpointing the presence and severity of damage. However, despite their ability to deliver quantitative insights into damage location and magnitude, these model-based techniques may fall short when applied to intricate structures, owing to potential inaccuracies in the underlying models and assumptions. Meanwhile, the rise of artificial intelligence (AI) and the proliferation of big data technologies have propelled data-driven methods to the forefront of structural damage detection in bridges[5–7]. Leveraging expansive datasets and sophisticated AI algorithms, contemporary techniques herald new avenues for identifying structural damages. These methods, proficient in autonomously detecting potential damages, scrutinize data from sensors affixed to bridges. Given their exceptional adaptability, data-driven approaches are adept at navigating the complexities of various structures and intricate damage patterns, areas where traditional model-based strategies may falter.

In recent years, the application of and research into damage identification technologies employing machine learning and data analytics have seen significant expansion. This growth underscores the substantial potential for automatically identifying damage-sensitive features within sensor data, facilitating pattern recognition. In this respect, Sun et al.[8] used a convolutional neural network (CNN) and partial least squares regression to estimate bridge node overload and proposed a bridge damage identification and location method. Through numerical simulation, this method was proven capable of accurately and reliably detecting and locating bridge damage under various conditions. Meanwhile, Wang and Cha[9] proposed a structural damage identification method using deep autoencoders and one-class support vector machines.

Received 2024-04-03, **Revised** 2024-05-14.

Biography: Duan Yuanfeng (1977—), male, doctor, professor, ceyfduan@zju.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 52361165658, 52378318, 52078459).

Citation: Duan Yuanfeng, Duan Zhengteng, Zhang Hongmei, et al. Bridge damage identification based on convolutional autoencoders and extreme gradient boosting trees [J]. Journal of Southeast University (English Edition), 2024, 40(3): 221 – 229. DOI: 10.3969/j. issn. 1003 – 7985. 2024. 03. 001.

Through experimental and numerical research verifications, it achieved a detection accuracy of up to 97.4%, especially in minor damage identification. Ma et al. [10] proposed a structure damage detection method based on variational autoencoders. Through numerical simulation and experimental verifications, it achieved accurate damage identification and was found suitable for actual SHM. Ni et al. [11] expanded bridge corrosion data using generative adversarial networks and achieved precise corrosion detection and segmentation using the DeepLab-V3 + network, significantly outperforming the U-Net network. Duan et al. [12] proposed an automated damage identification method for hanger cables in tied-arch bridges using CNN on the Fourier amplitude spectra of acceleration responses, showing improved performance over traditional methods under various damage scenarios and observational conditions.

Although AI-based damage identification technology has made significant progress, it still has several limitations, including insufficient feature extraction, long computational time, and the challenge of real-time damage detection. Given these limitations, this study introduces a two-phase damage identification framework that combines convolutional autoencoders (CAEs) with extreme gradient boosting trees (XGBoost), aiming to achieve efficient and real-time bridge damage identification.

1 Proposed Method

This section describes the proposed novel bridge damage identification method using structural acceleration data. This method is a two-phase technology combining CAEs and XGBoost using a stacked model integration approach. The main procedures are as follows:

1) Data acquisition. Collect the vertical acceleration of the structure using acceleration sensors installed on the bridge.

2) Data preprocessing. Perform data slicing operation to obtain data samples with a certain size (i. e., 5×512), which will be used for subsequent model training.

3) Feature extraction. Use the CAE model to perform reconstruction training on data samples, achieving feature extraction. CAE automatically learns the features of the input data through its convolutional layer, then extracts key features through the encoder, and finally generates a feature vector of length 256.

4) Damage identification. Use the XGBoost model to analyze the feature samples to achieve damage identification. The model employs its ensemble of decision trees to interpret and correlate the patterns within the features, effectively pinpointing areas of concern. This stage involves training the XGBoost algorithm on the 256-length feature vectors to predict the location and degree of structural damage.

The selection of CAEs and XGBoost for bridge damage identification capitalizes on their unique strengths to address the complexities inherent in studies of bridge damage identification. On the one hand, CAEs excel in compressing data samples into meaningful representations. Compared with deep learning methods such as CNN, CAEs employ unsupervised learning to perform reconstruction training on data samples, offering faster speeds and more efficient feature extraction capabilities while avoiding information loss. On the other hand, XGBoost excels at feature analysis and data mining, standing out among other algorithms for its sophisticated decision-making process. It complements CAEs by leveraging its high performance in predictive tasks to accurately assess the location and severity of damage. XGBoost uses the ensemble method to combine multiple decision trees. Given its efficient training and analysis methods, it has higher accuracy and efficiency than CNN when used to analyze lightweight feature data.

In this study, CAEs and XGBoost are combined using a stacked model integration method for bridge damage identification tasks. CAE is used to perform deep feature extraction from the original acceleration data, whereas XGBoost analyzes the extracted feature to achieve damage localization and damage degree identification.

1.1 CAEs

CAEs represent a specialized form of autoencoders particularly designed for processing grid-structured data, such as images[13]. CAEs leverage the CNN architecture, which is widely recognized for its efficiency in handling image data because of its ability to capture spatial hierarchies of features. The core objective of a CAE is to learn a compressed, feature-rich representation of the input data through an encoding-decoding process, facilitating tasks such as denoising, dimensional reduction, and feature learning in an unsupervised manner.

The architecture of a CAE, shown schematically in Fig. 1, consists of two main components: the encoder and the decoder. The encoder uses convolutional layers to progressively reduce the dimensionality of the input data, capturing essential features in a compressed form through filters that create encoding features. These encoding features summarize specific features at different locations in the input. Meanwhile, the decoder aims to reconstruct the

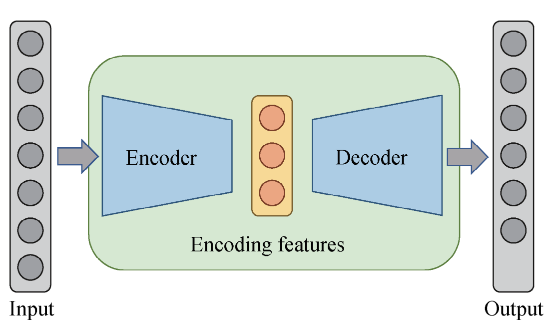

Fig. 1 Architecture of a CAE

original input from the encoding features, employing convolutional and up-sampling layers to gradually restore the spatial dimensions to match the original input size. The quality of this reconstruction indicates the model's efficacy in capturing the underlying data structure. Through this process, the CAE learns the intrinsic representation of the input data and can reconstruct high-quality acceleration data, providing a reliable foundation for further analysis.

Training a CAE involves minimizing a loss function that measures the difference between the input data and its reconstruction. This process enables the CAE to learn efficient data representations, with the encoder capturing the most informative features and the decoder learning to reconstruct the data based on these features. The loss function of CAE usually uses mean square error (MSE) loss, which can be described as

$$r_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2 \tag{1}$$

where x_i is the i-th data point of the input data; $\hat{x}_i$ is the i-th data point of the output data; N is the total number of data points.

In this study, acceleration data samples were fed into the CAE model for specialized reconstruction training. The CAE meticulously processed each sample through its convolutional layers to model and minimize the reconstruction error iteratively. As this error approached zero, indicating a precise representation of the data, the CAE shifted to extracting concise and highly informative features from the data. These features were encapsulated in a reduced-dimensional space, representing the intrinsic characteristics of the original acceleration data and thus effectively capturing the essential dynamics and patterns necessary for subsequent damage identification stages. Therefore, the encoding features, which contained comprehensive information about the original signals, were selected for the next stages.

1.2 XGBoost algorithm

XGBoost is a highly efficient and flexible implementation of the gradient boosting decision tree (GBDT)[14] technique, widely recognized for its exceptional performance in machine learning competitions and real-world applications[15].

This advanced ensemble learning technique leverages the iterative training of decision tree[16] models, which are then integrated into a robust composite model. XGBoost excels in feature analysis and data mining, making it particularly suitable for analyzing abstract and complex datasets. For this reason, it is employed in this study to analyze the features extracted by the CAEs.

The core methodology of XGBoost involves optimizing the loss function of the model through gradient descent during each training iteration. This optimization enhances the model's accuracy and responsiveness to various data patterns. XGBoost also incorporates several innovative features to boost performance, including a weighted quantile sketch for the better handling of continuous variables, a regularization term to prevent overfitting, and parallel processing to speed up computations[17-18].

The objective function of XGBoost consists of two parts: a loss function and a regularization term. Assuming a training data set $(\boldsymbol{X}_i, y_i)$, $i = 1, 2, \dots, N$, where $\boldsymbol{X}_i$ is the feature vector, and y_i is the corresponding real label, the goal of XGBoost is to minimize the following objective function:

$$F_{\mathrm{Obj}} = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

where $\hat{y}_i$ is the predicted value of the sample y_i by the model; f_k is the k-th tree; l is the loss function; Ω is the regularization term.

XGBoost has made the following major improvements to the traditional GBDT algorithm:

1) Regularization term. XGBoost introduces a regularization term to control the complexity of the model and prevent overfitting. Compared with traditional GBDT, the regularization term of XGBoost is more flexible and can effectively control the depth of the tree and the number of leaf nodes.

2) Loss function approximation. XGBoost uses an approximate method to solve the objective function, speeding up the model training process. This approximation method can reduce the amount of calculation and improve the efficiency of the algorithm.

3) Parallel processing. XGBoost introduces parallel processing in the training process, which can use multicore processors and distributed computing platforms to accelerate the model training process. Therefore, XGBoost stands as a formidable extension of the GBDT framework, offering enhancements in terms of model accuracy, efficiency, and flexibility, and is thus used in this study.

2 Bridge Overview

2.1 Bridge overview and sensor placement

In this research, a three-span continuous girder finite element model, featured in the 3rd International Competition for Structural Health Monitoring (IC-SHM 2022), served as the primary study object[19-20]. The competition, aimed at advancing SHM technologies, was cohosted by the Asia-Pacific Network of Centers for Research in Smart Structures Technology, alongside Tongji University, Harbin Institute of Technology, and the University of Illinois at Urbana-Champaign. It offered a wealth of datasets and sophisticated models invaluable for validating the damage identification approach proposed herein.

Leveraging this groundwork, this study conducted method validation using data for the three-span continuous girder bridge. The bridge spans a total length of 22 m, as shown in Fig. 2, with the side spans measuring 6 m each and the central span extending 10 m, resulting in a side-to-center span ratio of 0.6. The structure was made entirely of steel. The bridge was equipped with a total of five acceleration sensors across its length: one on each of the side spans and three on the mid-span, all tasked with capturing vertical acceleration. These sensors operated at a sampling rate of 100 Hz and were sequentially labeled A1 to A5. The bridge structure was segmented into 44 elements and 45 nodes in total. To simulate real-world conditions, a moving vehicle load was applied on the bridge, enhancing the authenticity of the sensor-collected response data.

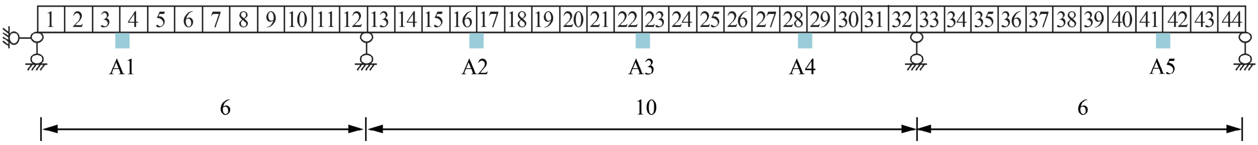

Fig. 2 Bridge overview and sensor placement (unit: m)

2.2 Damage scenario setup

The dataset in this study was derived from acceleration sensors mounted along the bridge's girder, ensuring a thorough data acquisition to observe the bridge's dynamic responses comprehensively through virtual tests by finite element analysis. Within the scope of investigating structural damage identification, intentional damages of varying levels were applied to two distinct segments of a bridge span: Elements 7 and 22. The remainder of the structure remained undamaged. Additionally, the bridge's first frequency was recorded under each damage scenario, with the specifics of eleven damage cases detailed in Table 1.

Table 1 Damage cases

Damage case	Damage degree		First frequency/Hz
	Element 7	Element 22	
H0	0	0	9.436 7
M1	0	0.2	9.361 3
M2	0	0.3	9.309 3
M3	0	0.4	9.242 1
S1	0.1	0	9.425 1
S2	0.3	0	9.392 4
S3	0.5	0	9.334 5
D1	0.2	0.2	9.335 6
D2	0.2	0.4	9.216 7
D3	0.4	0.2	9.293 3
D4	0.4	0.4	9.175 1

The simulation produced one healthy state and ten damage cases. The baseline working condition, H0, represents a state without any damage, whereas the conditions M1 to M3 denote states of mid-span damage. S1 to S3 denote states of side-span damage. Specifically, for Element 22 at the mid-span, damages were artificially induced at degrees of 0.2, 0.3, and 0.4 in elastic modulus reduction from M1 to M3. Similarly, for Element 7 situated at the side span, damages were introduced at degrees of 0.1, 0.3, and 0.5 in elastic modulus reduction from S1 to S3. To study the case of multiple damages to the structure, a double-damage condition was also set up, as shown in damage cases D1 to D4 in Table 1. Among them, two damage degrees of 0.2 and 0.4 in modulus reductions were set for the mid- and side-span positions, respectively, with a total of four combinations. The collected signals spanned 2 000 s, generating 200 000 data points for each sensor and culminating in a dataset with a size of 5 × 200 000 for each case.

3 Feature Extraction Using CAEs

Before employing CAEs for feature extraction, data slicing must be performed on the original data set. A sliding window with a window length of 512 and a stride of 128 was used to divide the data into samples of size 5 × 512, which served as the input for the CAEs. This process resulted in a total of 1 536 samples for each damage case. The first 80% of the samples were used as the training set, and the latter 20% were used as an independent validation set.

Carefully designed CAEs can effectively reconstruct acceleration data samples. The structure of the designed CAE network is detailed in Table 2. The autoencoder mainly consisted of two parts: an encoder and a decoder. The encoder part was constructed from five convolutional layers and four batch normalization layers and used the LeakyReLU activation function to introduce nonlinear processing. Correspondingly, the decoder consisted of five deconvolution layers and four batch normalization layers, activated using the ReLU activation function. The output size of the encoder was 256 × 1 × 1, which was used as the feature for subsequent training of the XGBoost model.

To achieve the best possible performance from the model, its hyperparameter setup was meticulously established through a series of experiments and analyses. The following provides an overview of the process for selec-

Table 2 Architecture of the CAE network

Layer	Input shape	Output shape	Kernel size	Stride	Padding	Activation
Conv2d	(1, 5, 512)	(16, 5, 128)	(3, 4)	(1, 4)	(1, 0)	LeakyReLU
Conv2d + BN	(16, 5, 128)	(32, 4, 32)	(2, 4)	(1, 4)	(0, 0)	LeakyReLU
	(32, 4, 32)	(64, 3, 8)	(2, 4)	(1, 4)	(0, 0)	LeakyReLU
	(64, 3, 8)	(128, 2, 2)	(2, 4)	(1, 4)	(0, 0)	LeakyReLU
	(128, 2, 2)	(256, 1, 1)	(2, 2)	(1, 1)	(0, 0)	LeakyReLU
UConv2d + BN	(256, 1, 1)	(128, 1, 4)	(1, 4)	(1, 1)	(0, 0)	ReLU
	(128, 1, 4)	(64, 2, 8)	(2, 4)	(1, 2)	(0, 1)	ReLU
	(64, 2, 8)	(32, 3, 32)	(2, 4)	(1, 4)	(0, 0)	ReLU
	(32, 3, 32)	(16, 4, 128)	(2, 4)	(1, 4)	(0, 0)	ReLU
UConv2d	(16, 4, 128)	(1, 5, 512)	(2, 4)	(1, 4)	(0, 0)	

ting the model hyperparameters. The loss function used the MSE loss, whereas the Adam optimizer was employed to minimize the reconstruction loss. The model's batch size was 64, identified as the ideal size through extensive experimentation. This size struck a balance between training speed and the quality of convergence while considering the limitations of available hardware resources. The training was conducted over 50 rounds, a duration found to be sufficient for CAE to reach convergence.

During the training process of the CAE model, the data samples were reconstructed. Upon reaching 50 epochs, the loss function began showing signs of converging and stabilizing for the remainder of the training period. Fig. 3 (a) displays the log-scaled loss functions for both the training and validation sets. After the CAEs were trained, they could reconstruct the acceleration data, which showed good consistency with the original data, as shown in Fig. 3(b). At this point, CAE effectively captured the underlying patterns in the data. The encoding block extracted all relevant information in the original signal as features to prepare for subsequent XGBoost training aimed at damage identification. This simplified approach high-lights the efficiency of CAE training and its critical role in high-quality feature extraction.

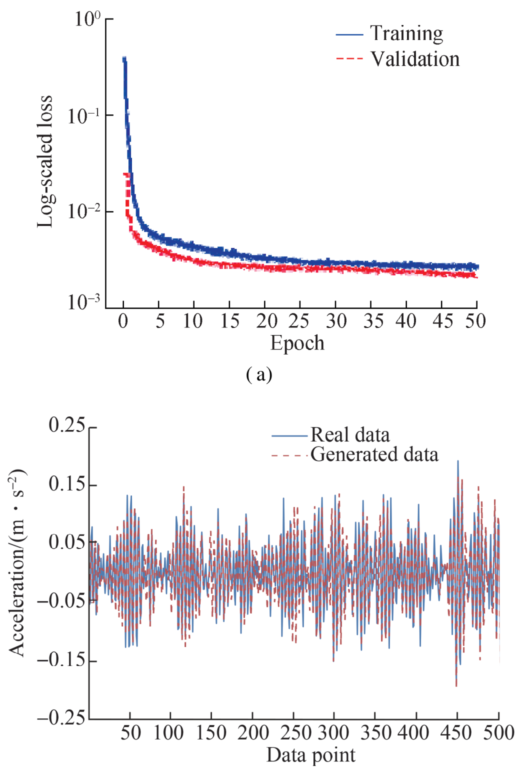

Fig. 3 Training result of CAE. (a) Log-scaled loss function; (b) Reconstruction result

4 Damage Identification Based on XGBoost

4.1 XGBoost model

XGBoost was selected for its superior performance in handling large-scale and high-dimensional data. It is known for its computational efficiency, capability to manage sparse data, and its built-in mechanisms to prevent overfitting. The feature vector extracted by CAE comprised 256 dimensions, exhibiting high-dimensional attributes that pose challenges for analysis. Therefore, XGBoost was selected for this particular task because of its effectiveness in dissecting and gleaning insights from complex features and datasets.

The XGBoost model employed several key parameters: the maximum depth D was set at 6 to limit tree complexity, the learning rate L was 0.1 to moderate updates during training, the subsample ratio S was 0.8 to use 80% of data for tree building, the feature sampling ratio F was 0.9 to sample 90% of features per tree, the minimum child weight W was 1 to prevent overfitting, and the number of trees T was 100 to define the model scale. These settings collectively optimized the model's performance.

4.2 Damage identification

The feature samples obtained from the CAEs were partitioned into two segments. The first 80% were allocated as the training set, and the remaining 20% were designated as the test set. Consequently, each damage case contained 1 230 training samples and 306 test samples. Subsequently, the designed XGBoost model was utilized to perform regression analysis on the CAE features. XGBoost trained and predicted by iteratively adding decision trees, each designed to correct the errors of the previous ones, thereby continuously improving the model accuracy. Fig. 4

displays the MSE on the test set at each stage of adding trees, demonstrating the progressive enhancements in performance throughout the model-building process.

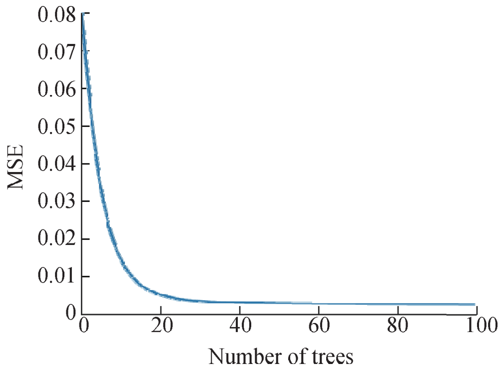

Fig. 4 Prediction error during the XGBoost building process

Fig. 4 shows a sharp decrease in MSE as the number of decision trees increased from 0 to 20, after which the reduction in MSE stabilized, suggesting that the model approached optimal performance. After completing the iterative training of XGBoost, the model was then used for damage prediction. The predicted values of all test samples in each working condition were averaged as the damage degree prediction result of the working condition.

The prediction results for the mid-span damage are shown in Fig. 5(a). The damage cases M1, M2, and M3 represent the degrees of injury from mild to severe, corresponding to real damage degrees of 0.2, 0.3, and 0.4 in modulus reductions, respectively. In the figure, the predicted damage degrees were 0.221, 0.306, and 0.375, respectively, showing that the predicted values were very close to the actual damage degree. For the mild damage condition M1, the predicted damage degree was slightly higher than the actual damage degree by 0.021. This shows that the model had a slight tendency to overestimate the detection of minor damage. For the medium damage condition M2, the difference between the prediction and the actual value was further reduced to 0.006, indicating that the model had higher accuracy at the medium damage degree. Finally, for the severe damage condition M3, the predicted damage degree was slightly lower than the actual value, with a difference of 0.025. This indicates that the model is slightly conservative in predicting severe damage.

The damage at side-span locations was also predicted and analyzed, as shown in Fig. 5(b). The actual damage conditions of S1 to S3 corresponded to real damage degrees of 0.1, 0.3, and 0.5 in modulus reductions, and the predicted damage degrees were 0.129, 0.283, and 0.473, respectively. The figure shows that the damage identification results of the side span were worse than

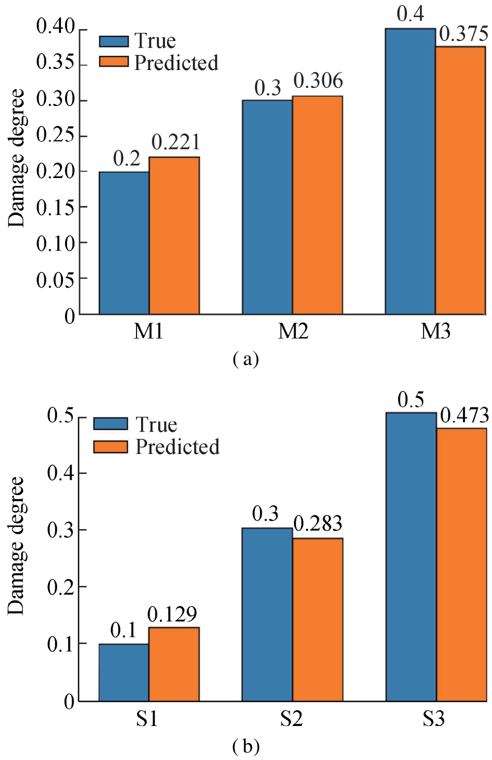

Fig. 5 Single-damage identification results. (a) Mid-span (Element 22); (b) Side span (Element 7)

those of the mid-span. This was because the damaged element in the side span was farther away from the acceleration sensors. Thus, the changes in structural vibration characteristics caused by the damage could not be perfectly reflected by the acceleration signal of the sensors. Additionally, the slight damage S1 and severe damage S3 of the side span were similar to the case of the mid-span. The model tended to overestimate minor damage and underestimate severe damage. This may be caused by the model being affected by the data of the other two working conditions during training. Collectively, the highest identification error noted in single-damage scenarios was 2.9% (Case S1).

The double-damage identification results are shown in Fig. 6. The results show a high correlation between the damage degrees and the predicted values. For a damage degree of 0.2, the model had a slightly overestimated value, and for a damage degree of 0.4, the model prediction value was low. This may have been caused by the model being affected by other data sets during training. Overall, the maximum prediction error for double-damage scenarios was 3.1% (in Case D1), and the predicted results showed good consistency with the actual results. Apparently, the model prediction of damage degree was generally consistent with the actual situation, showing that the model has certain accuracy and reliability in double-damage identification.

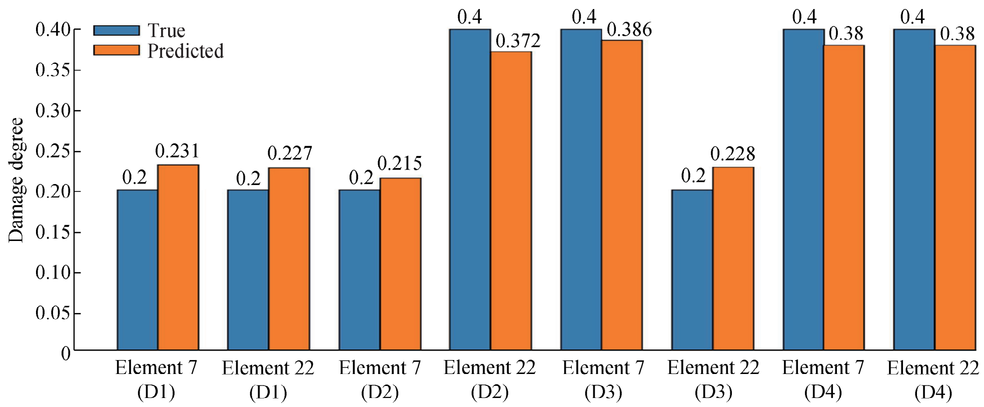

Fig. 6 Double-damage identification results

5 Experimental Validations

To further validate the proposed method, damage classification studies were conducted on a scaled cable-stayed bridge model in the laboratory. The model was based on the Wen Hui Bridge in Hangzhou, with a geometric similarity ratio of 1∶55, and was made of aluminum alloy, as shown in Fig. 7. Five acceleration sensors were arranged on the bridge girder, including one on each side span and three on the mid-span. The sensors collected the vertical acceleration of the structure at a sampling frequency of 512 Hz. The stay cables were numbered sequentially as C1 to C20. The bridge layout and sensor locations are shown in Fig. 8.

Fig. 7 Scaled cable-stayed bridge model in-field photo

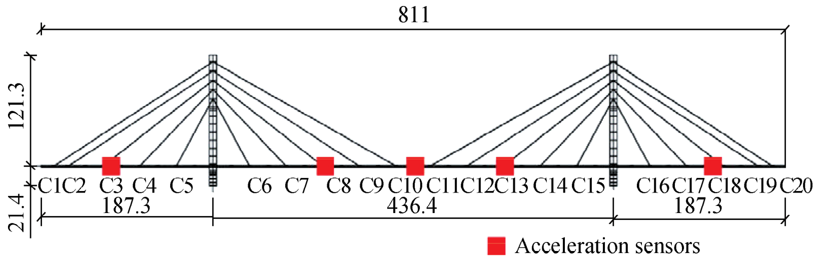

Fig. 8 Cable-stayed bridge sensor layout diagram (unit: cm)

Damage simulation was performed on the cables, which were the vulnerable components of the cable-stayed bridge. In the scale model, the cables had a scaled-down diameter of only 1 mm, making it challenging to introduce damage by altering their cross-sectional area. Consequently, damage was simulated by reducing the cable tension by 20%. The damage conditions of the cables and the corresponding 1st- to 3rd-order natural frequencies of the bridge are shown in Table 3.

Table 3 Damage cases and natural frequencies Hz

Damage case	Case 0	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Damage content	No damage	C4	C6	C8	C13	C15	C17
1st-order frequency	4.922	4.917	4.920	4.921	4.919	4.922	4.918
2nd-order frequency	6.934	6.928	6.932	6.929	6.923	6.931	6.926
3rd-order frequency	9.838	9.826	9.831	9.827	9.824	9.834	9.825

The 20% reduction in the tension force of a single cable had a negligible effect on the overall behavior of the bridge as the scaled-down bridge retained a high stiffness. The first three natural frequencies remained almost unchanged, making it impossible to determine the occurrence of damage based on frequency alone.

By applying the same CAE-XGBoost damage identification procedure mentioned in the former sections, the damage classification results are shown in Fig. 9. There were 448 test samples for each damage case. Almost all the samples were correctly identified, with only a few samples being misclassified as other conditions. For Case

3, 19 samples were incorrectly predicted (95.8% accuracy), with 8 of them being predicted as Case 4. This misclassification occurred due to the close proximity of stay cables C8 and C13, resulting in a small number of samples being incorrectly identified.

Actual \ Predicted	Case 0	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Case 0	444	1	1	0	1	0	1
Case 1	0	446	1	0	0	0	1
Case 2	0	2	441	1	1	1	2
Case 3	1	2	2	429	8	5	1
Case 4	2	0	2	0	444	0	0
Case 5	0	0	0	1	1	446	0
Case 6	0	0	1	3	0	0	444

Fig. 9 Cable-stayed bridge damage identification results

6 Conclusions

1) This method maintained the integrity of the original data by extracting features directly from raw acceleration data, eliminating the need for transformation into images or any other preprocessing and thereby avoiding information loss.

2) Numerical simulations demonstrated that the proposed method can accurately identify damages in a three-span continuous girder under six single-damage scenarios, with identification errors all within 2.9%; in four double-damage scenarios, the errors remained within 3.1%.

3) In the experimental validations on a scaled cable-stayed bridge, the proposed method successfully identified cable damage at various locations using only acceleration sensors installed on the main girder, achieving a recognition accuracy of over 95.8% in all damage cases.

Acknowledgments

The authors would like to express heartfelt gratitude to Prof. Sun Limin and Prof. Xia Ye of Tongji University for providing the model and data of the three-span continuous girder used in this study through the 3rd International Competition for Structural Health Monitoring (IC-SHM 2022), co-hosted by the Asia-Pacific Network of Centers for Research in Smart Structures Technology (ANCRiSST) and Tongji University.

References

[1] Zhu H P, Shen Z H, Weng S. Damage identification for vertical stiffness of joints of periodic continuous beams based on spectral element method[J]. *Journal of Southeast University (English Edition)*, 2023, **39**(4): 323 − 332. DOI: 10.3969/j.issn.1003-7985.2023.04.001.

[2] Jin W L, Yu Y F, Bai Y L. Life prediction and sensitivity analysis of reinforced concrete beams after corrosion and fatigue damage[J]. *Journal of Southeast University (Natural Science Edition)*, 2024, **54**(2): 260 − 267. DOI: 10.3969/j.issn.1001-0505.2024.02.002. (in Chinese)

[3] Tu Y M, Lu S L, Wang C. Damage identification of steel truss bridges based on deep belief network[J]. *Journal of Southeast University (English Edition)*, 2022, **38**(4): 392 − 400. DOI: 10.3969/j.issn.1003-7985.2022.04.008.

[4] Sun L M, Shang Z Q, Xia Y, et al. Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection[J]. *Journal of Structural Engineering*, 2020, **146**(5): 04020073. DOI: 10.1061/(asce)st.1943-541x.0002535.

[5] An Y H, Chatzi E, Sim S H, et al. Recent progress and future trends on damage identification methods for bridge structures[J]. *Structural Control and Health Monitoring*, 2019, **26**(10): e2416. DOI: 10.1002/stc.2416.

[6] Karimi S, Mirza O. Damage identification in bridge structures: Review of available methods and case studies[J]. *Australian Journal of Structural Engineering*, 2023, **24**(2): 89 − 119. DOI: 10.1080/13287982.2022.2120239.

[7] Yang D H, Sun J Z, Yi T H, et al. Early warning technology of long-span bridge bearing deterioration considering time lag effects of thermal-induced displacement[J]. *Journal of Southeast University (Natural Science Edition)*, 2024, **54**(2): 268 − 274. DOI: 10.3969/j.issn.1001-0505.2024.02.003. (in Chinese)

[8] Sun H S, Song L, Yu Z W. A deep learning-based bridge damage detection and localization method[J]. *Mechanical Systems and Signal Processing*, 2023, **193**: 110277. DOI: 10.1016/j.ymssp.2023.110277.

[9] Wang Z L, Cha Y J. Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage[J]. *Structural Health Monitoring*, 2021, **20**(1): 406 − 425. DOI: 10.1177/1475921720934051.

[10] Ma X R, Lin Y Z, Nie Z H, et al. Structural damage identification based on unsupervised feature-extraction via Variational Auto-encoder[J]. *Measurement*, 2020, **160**: 107811. DOI: 10.1016/j.measurement.2020.107811.

[11] Ni Y H, Lu H, Ji C, et al. Comparative analysis on bridge corrosion damage detection based on semantic segmentation[J]. *Journal of Southeast University (Natural Science Edition)*, 2023, **53**(2): 201 − 209. DOI: 10.3969/j.issn.1001-0505.2023.02.003. (in Chinese)

[12] Duan Y Y, Chen Q Y, Zhang H M, et al. CNN-based damage identification method of tied-arch bridge using spatial-spectral information[J]. *Smart Structures and Systems*, 2019, **23**: 507 − 520. DOI: 10.12989/SSS.2019.23.5.507.

[13] Masci J, Meier U, Ciresan D, et al. Stacked convolutional auto-encoders for hierarchical feature extraction[C]// *International Conference on Artificial Neural Networks*. Espoo, Finland, 2011: 52 − 59.

[14] Friedman J H. Greedy function approximation: A gradient boosting machine[J]. *The Annals of Statistics*, 2001, **29**

(5): 1189 − 1232. DOI: 10.1214/aos/1013203451.
[15] Chen T Q, Guestrin C. XGBoost: A scalable tree boosting system[C]//*Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, USA, 2016: 785 − 794.
[16] Song Y Y, Lu Y. Decision tree methods: Applications for classification and prediction[J]. *Shanghai Archives of Psychiatry*, 2015, **27**(2): 130 − 135. DOI: 10.11919/j.issn.1002-0829.215044.
[17] Sagi O, Rokach L. Approximating XGBoost with an interpretable decision tree[J]. *Information Sciences*, 2021, **572**: 522 − 542. DOI: 10.1016/j.ins.2021.05.055.
[18] Asselman A, Khaldi M, Aammou S. Enhancing the prediction of student performance based on the machine learning XGBoost algorithm[J]. *Interactive Learning Environments*, 2023, **31**(6): 3360 − 3379. DOI: 10.1080/10494820.2021.1928235.
[19] Lei X J, Sun L M, Xia Y. Lost data reconstruction for structural health monitoring using deep convolutional generative adversarial networks[J]. *Structural Health Monitoring*, 2020, **20**: 2069 − 2087. DOI: 10.1177/1475921720959226.
[20] Li Y X, Ni P, Sun L M, et al. A convolutional neural network-based full-field response reconstruction framework with multitype inputs and outputs[J]. *Structural Control and Health Monitoring*, 2022, **29**(7): e2961. DOI: 10.1002/stc.2961.

基于卷积自动编码器和极端梯度提升树的桥梁损伤识别

段元锋　段政腾　章红梅　郑荣俊

（浙江大学建筑工程学院，杭州 310058）

摘要:为了提升桥梁损伤检测的准确性和效率,提出了一种新的数据驱动损伤识别方法. 首先采用卷积自动编码器(CAE)通过数据重构方式从桥梁结构加速度信号中提取关键特征,然后通过极端梯度提升树(XGBoost)对特征数据进行分析,从而实现高精度和高泛化能力的损伤识别. 将所提方法用于一座三跨连续梁的数值模拟研究,并在一座斜拉桥缩尺模型上进行了试验验证. 数值模拟结果表明,三跨连续梁6种单损伤工况的识别误差均在2.9%以内,4种双损伤工况的识别误差均在3.1%以内. 试验验证结果表明,斜拉桥单根拉索的索力减小20%时,该方法仅通过主梁上的传感器即可准确识别不同位置的拉索损伤,在各个损伤案例中均达到了95.8%以上的识别精度. 所提方法在不同的损伤场景下均具有较高的识别精度和泛化能力.

关键词:结构健康监测;损伤识别;卷积自动编码器;极端梯度提升树;机器学习

中图分类号:TU39