Loose bolt localization and torque prediction in a bolted joint using lamb waves and explainable artificial intelligence

Hu, Muping; Yue, Nan; Groves, Roger M.

DOI
10.1177/14759217241241976

Publication date
2024

Document Version
Final published version

Published in
Structural Health Monitoring

Citation (APA)
Hu, M., Yue, N., & Groves, R. M. (2024). Loose bolt localization and torque prediction in a bolted joint using lamb waves and explainable artificial intelligence. *Structural Health Monitoring*. https://doi.org/10.1177/14759217241241976

SHM
Structural Health Monitoring

Original Article

Structural Health Monitoring
1–21
© The Author(s) 2024
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/14759217241241976
journals.sagepub.com/home/shm
Sage

Loose bolt localization and torque prediction in a bolted joint using lamb waves and explainable artificial intelligence

Muping Hu[1,2], Nan Yue[2] and Roger M. Groves[2]

Abstract

With the increasing application of artificial intelligence (AI) techniques in the field of structural health monitoring (SHM), there is a growing interest in explaining the decision-making of the black-box models in deep learning-based SHM methods. In this work, we take explainability a step further by using it to improve the performance of AI models. In this work, the results of explainable artificial intelligence (XAI) algorithms are used to reduce the input size of a one-dimensional convolutional neural network (1D-CNN), hence simplifying the CNN structure. To select the most accurate XAI algorithm for this purpose, we propose a new evaluation method, feature sensitivity (FS). Utilizing XAI and FS, a reduced dimension 1D-CNN regression model (FS-X1D-CNN) is proposed to locate and predict the torque of loose bolts in a 16-bolt connected aluminum plate under varying temperature conditions. The results were compared with 1D CNN with raw input vector (RI-1D-CNN) and deep autoencoders-1D-CNN (DAE-1D-CNN). It is shown that FS-X1D-CNN achieves the highest prediction accuracy with 5.95 mm in localization and 0.54 Nm in torque prediction, and converges 10 times faster than RI-1D-CNN and 15 times faster than DAE-1D-CNN, while only using a single lamb wave signal path.

Keywords

Lamb waves, explainable AI, structural health monitoring, one-dimensional convolutional neural network, loose bolt localization, torque prediction

Introduction

Threaded fasteners, commonly referred to as “industrial rice” due to their ubiquity, are foundational components in various industrial fields, prized for their simplicity and ease of assembly.[1] However, when subjected to various environmental factors such as vibration, impact, and changing temperatures,[2] bolted connections are prone to loosening after assembly, which can directly affect the safety and reliability of the bolted structures, even leading to catastrophic ccidents. Statistics show that approximately 20% of mechanical system failures worldwide each year can be attributed to the self-loosening of threaded bolts or fasteners.[3] In 2017, the offshore supply vessel Red Dawn experienced an engine failure during a major overhaul due to improperly tightened bolts, which caused a nearly $1 million loss.[4] Similarly, in 2015, the Carnival Liberty cruise ship suffered an engine room fire caused by a loose bolt on a fuel injector pump, resulting in losses of $1.7 million.[5] These incidents highlight the critical importance of continuous monitoring of bolted connections to ensure structural safety and reliability.

Structural health monitoring (SHM) is a potential technology that comprises in-service data collection and signal analysis to ensure structural safety and reduce maintenance costs.[6,7] A variety of SHM methods for monitoring bolt loosening have been proposed in recent years, including electro-mechanical impedance methods,[8,9] vibration-based methods,[10,11] electrical conductivity,[12] acoustic emission methods,[13,14] etc. Among these, guided wave-based SHM methods are

[1]College of Aerospace and Civil Engineering, Harbin Engineering University, Harbin, P. R. China
[2]Aerospace Structures & Materials Department, Delft University of Technology, Delft, The Netherlands

Corresponding author:
Muping Hu, College of Aerospace and Civil Engineering, Harbin Engineering University, No. 145 Nantong Street, Nangang District, Harbin, Heilongjiang Sheng 150001, P. R. China.
Email: humuping@hrbeu.edu.cn

considered one of the most promising technologies for large-scale, in-service engineering structural monitoring due to their low energy consumption, remote inspection capabilities, the ability to monitor large areas from a single point, and the ability to inspect hard-to-reach areas.[15,16]

Many studies have applied guided waves to bolt-loosening detection.[17–19] However, most of these studies require manual extraction of signal features, such as wave energy dissipation,[20–22] mode conversion,[23] and nonlinear acoustic properties,[24,25] and then establish relationships between these statistical indicators and bolt connection states through experiments. However, in practical engineering, guided waves are susceptible to industrial noise,[26] which can cause those methods that perform well in a stable laboratory environment to no longer to be able to provide an accurate damage assessment in practice once the working environment or operating conditions (vibration,[27] temperature,[28,29] humidity,[30] etc.) change.[7] In addition, the difficulty of identifying suitable feature extraction methods for large structures with more complex connections can also be an obstacle to the complete automation of the monitoring task.

Deep learning-based SHM (DeepSHM[31]) methods have the potential to address these problems. They can extract high-level features without a priori knowledge in SHM and possess very strong nonlinear analysis and generalization capabilities, which gives them great advantage for in-service monitoring of large structures in unstable environments.[6] However, due to the black-box nature of AI models, their results are often difficult to understand and trust. Therefore, there is an increasing demand for explainable research on AI models. As an emerging technology in the field of AI, explainable artificial intelligence (XAI) can open the black box of AI and help humans to better understand the decisions of AI models.[32]

In recent years, explainability research for the DeepSHM methods has begun to emerge.[33–37] Meister et al.[38] used a CNN for damage classification during the automated fiber laying processes of composite structures and visualized the results of CNN using the Smooth Integrated Gradients algorithm. Geetha and Sim et al.[39] analyzed the metadata of a one-dimensional deep learning model by local XAI and visualized the knowledge transfer within the hidden layers by t-distributed Stochastic Neighborhood Embedding (t-SNE). In our previous work,[40] we employed six different XAI algorithms to analyze the decision-making process of a one-dimensional convolutional neural network (1D-CNN) model for detecting loose bolts and provided physical explanations for its decisions. Chen and Lee[41] used the results of Grad CAM to demonstrate that CNN prefers the information from the high-frequency band for the bearing faults diagnosis, which differs from the traditional methods that rely on low-frequency features. Letzgus et al.[42] proposed two novel XAI algorithms for explaining regression models. Lomazzi et al.[43] employed a layer-wise relevance propagation algorithm to explain the predictions of a CNN model that processes ultrasonic guide waves for damage diagnosis.

However, most of the current explainability research has been focused on explaining the decisions made by AI models, with little application beyond this. Zacharias et al.[44] used the result of Shapley Additive Explanations (SHAP) to do the feature selection of the XGBoost model and provide a design framework consisting of meta-requirements and design principles for explainable feature selection. Polle et al.[45] used the explainable neural network (XANN) architecture to examine and validate the artificially generated signal features and to improve the augmentation process for temperature compensation.

Taking these examples, we identify that XAI can contribute to the improvement of AI models. Therefore, in this paper, we aim to make rational use of the explanatory information generated by XAI and feed it back into the design of the DeepSHM method to reduce the computational costs. A feature sensitivity (FS) and XAI-based 1D-CNN regression model (FS-X1D-CNN) are proposed in this paper, which leverage the results of XAI to reduce the size of the input vector for the 1D CNN model to enhance computational efficiency while ensuring prediction accuracy.

Specifically, the 1D CNN classification model was used to detect the loose bolt in a 16-bolt-connected aluminum plate under temperature variations. Then the decision-making process of the model was analyzed using the six different XAI algorithms. Based on the results of XAI, the importance of each signal in the input vector was assessed, and the signal with the highest importance was selected as the new input to the 1D CNN regression model for loosen blot torque prediction. Therefore, FS-X1D-CNN can also enhance the efficiency of the monitoring of multiple bolts. By focusing on specific pairs of transducers, it can simplify the tasks of signal collection, transmission, and analysis.

Considering that different XAI algorithms provide different assessments of signal importance, accurately selecting the most accurate XAI algorithm is also a challenge. To address this issue, an evaluation method FS is proposed to identify the most accurate XAI algorithms in finding important features. Finally, the FS-X1D-CNN, X1D-CNN, RI-1D-CNN, and deep autoencoders-1D-CNN (DAE-1D-CNN) regression models were used to locate and predict the torque of the loose bolt, and the positioning accuracy, torque prediction accuracy, and convergence time are compared.

Methodology

An FS-X1D-CNN model that utilizes the knowledge from XAI algorithms to reduce the size of the input vector is proposed in this paper, the flowchart of the process is shown in Figure 1. First, signals are collected from the experiment for training and testing of the 1D CNN classification model. The proposed methodology was tested using four piezoelectric ultrasonic sensors loaded onto a bolted aluminum plate. By using each sensor in sequence as an exciter, with the remaining sensors serving as receivers, different monitoring paths can be formed: P1, P2, P3 … Pi. Subsequently, all the signals from these monitoring paths are arranged into a vector which serves as input to the 1D CNN model which is used to classify different bolt-loosening cases. The next step is to analyze the decisions of the classification model using XAI algorithms. Then the FS method is applied to assess the accuracy of various XAI algorithms in identifying important features of all the signals in the raw input vector. Then, utilizing the best XAI algorithm to select the most important signal from the raw input, which is used to train the FS-X1D-CNN model is applied to locate and predict the torque on the loose bolt. The analysis process of the XAI algorithm is not involved in the training and testing of the FS-X1D-CNN model.

One-dimensional convolutional neural network

1D CNN is a specialized type of CNN primarily used for temporal data, which has a high memory and computational efficiency compared to other architectures.[46] It consists of two stages: feature extraction and classification/regression.[47] The feature extraction stage consists of stacking and alternating convolutional layers and pooling layers. Input data are convolved by 1D kernel filters, with each filter sharing the same learnable weight matrix over the entire input data to extract specific features.[48] The convolution calculation is as follows:

$$y_u^{m+1} = \Sigma_{k \in N_m} x_k^m * w_u^{m+1} + b_u^{m+1} \quad (1)$$

where y_u^{m+1} and x_k^m are the feature maps of the u-th and k-th channels in the $(m + 1)$-th and m-th layers, respectively. N_m is the number of convolutional kernels in the m-th layer. w_u^{m+1} and b_u^{m+1} are the weight and bias of the $(m + 1)$-th layers, respectively. * represents the 1D

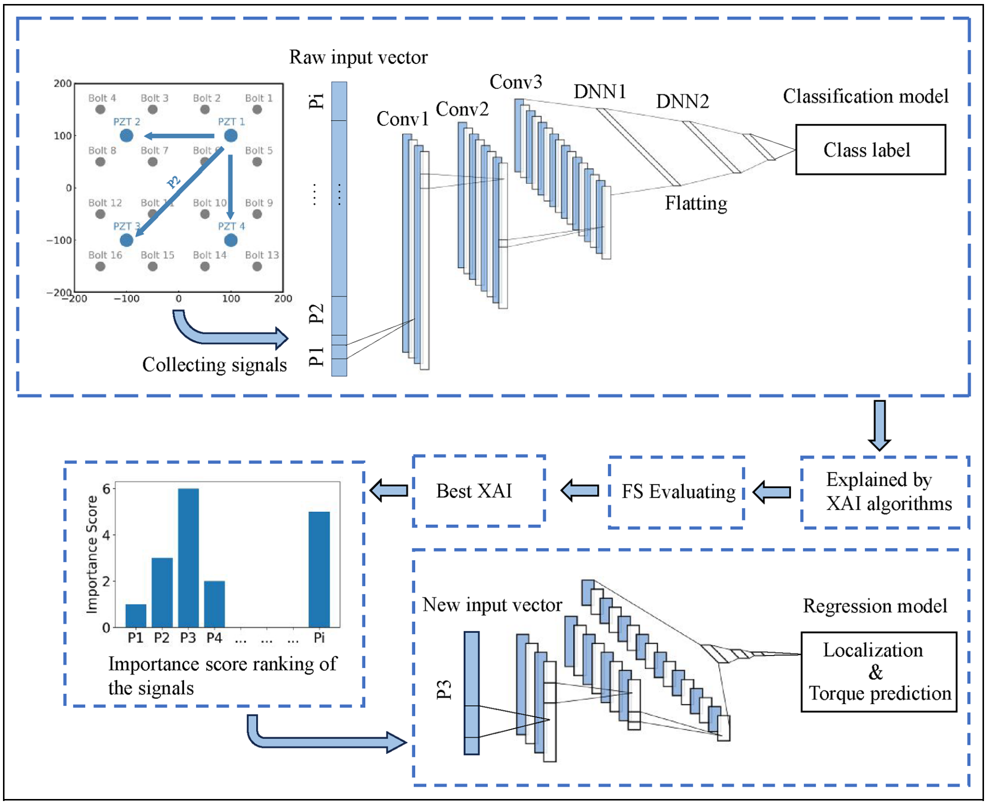


Figure 1. Flowchart of the FS-X1D-CNN model.
1D CNN: one-dimensional convolutional neural network; FS-X1D-CNN: feature sensitivity and XAI based 1D-CNN regression model; XAI: explainable artificial intelligence.

convolutional operation. After the convolution, the output goes to the max-pooling layer.

The structure of the classification/regression stage is fully connected layers and is calculated as follows:

$$y_q^{n+1} = \mathrm{ReLU}\left(w_q^{n+1} x_p^n + b_q^{n+1}\right) = \max\left[0, \left(w_q^{n+1} x_p^n + b_q^{n+1}\right)\right] \tag{2}$$

where y_q^{n+1} is the q-th value in the $(n + 1)$-th layer, x_p^n is the p-th value in the n-th layer, ReLU represents the linear rectification activation function, w_q^{n+1} and b_q^{n+1} are the weights and bias of $(n + 1)$-th layer.

For the model's training, cross-entropy is adopted to calculate the error between the 1D CNN predicted and the true label:

$$\mathcal{L}(\Theta) = -\left[\hat{Y}_C \log\left(F(V_C)\right) + \left(1 - \hat{Y}_C\right)\log(1 - F(V_C))\right] \tag{3}$$

where $\mathcal{L}$ is the loss function, F is the function of 1D CNN model, Θ contains all the parameters (weights and bias) of F, V_C represents the input vector from class C, and $\hat{Y}_C$ is the label of V_C. The 1D CNN is updated using a stochastic gradient descent algorithm with Adam optimizer.[49]

XAI algorithms

Six different XAI algorithms are employed for the explanation of the 1D CNN classification model. Then the XAI result obtained can be used for the calculation of importance scores for the data points of the input vector, a more detailed description can be found in our previous work.[40] Subsequently, the importance scores of all data points are summed to obtain the importance scores for the signal on each monitoring path.

Sensitivity analysis.

Sensitivity analysis[50] uses the square of the derivative value of the input vector to calculate the importance score of each data point. Then, the importance score of the signal can be represented by the sum of the importance scores of all data points:

$$I_{SA}(P_i) = \sum_{s=1}^{s_n}\left(\frac{\partial F}{\partial v_s^{P_i}}\right)^2 \tag{4}$$

P_i represents the i-th monitoring path, s represents the number of data points, s_n is the length of each signal, and $v_s^{P_i}$ represents the s-th data point of the signal on the P_i monitoring path.

Smooth Simple Taylor.

Smooth simple Taylor[40] uses the first-order term of the Taylor expansion of the 1D CNN model at the root point as the importance score for each data point and smooths the results by adding noise for averaging:

$$I_{SST}(P_i) = \sum_{s=1}^{s_n}\left(\frac{1}{N_{noise}}\sum_{1}^{N_{noise}}\frac{\partial F}{\partial v_s^{P_i}}(\tilde{v}) \cdot (v'_s - \tilde{v}_s)\right) \tag{5}$$

where N_{noise} is the number of noisy samples, $\tilde{v}$ is the root vector that represents the neutral points on the decision boundary of the 1D CNN model, and v'_s represents the signal with noise.

Deep Taylor.

In deep Taylor,[51] the importance vector of the neural network's last layer is first calculated. Then, this importance vector is propagated layer by layer forward through Taylor expansion. The transfer between the n-th and $(n + 1)$-th layer can be calculated as:

$$I_{DT}^{(n)}(x_p) = \sum_q \frac{\partial I_{DT}^{(n+1)}(x_p)}{\partial x_p}\bigg|_{\{\tilde{x}_p^{(q)}\}} \cdot \left(x_p - \tilde{x}_p^{(q)}\right) \tag{6}$$

where $\tilde{x}_p^{(q)}$ represents the root point vector in the n-th layer. After backpropagation, the importance scores of the first layer (input vector) are summed to obtain the importance score of each signal:

$$I_{DT}(P_i) = \sum_{s=1}^{s_n} I_{DT}^{(1)}\left(v_s^{P_i}\right) \tag{7}$$

Gradient-weighted class activation mapping.

Grad CAM[52] focuses on the information flow into the last convolutional layer. First, the activations of each channel are weighted by its average gradient. Then, the weighted activations are summed up and the negative values are filtered out by a ReLU function. Finally, the importance vector of the last convolutional layer is mapped back to the input vector through a linear transformation:

$$I_{GC}(v_s) = \mathrm{ReLU}\left(\sum_{k=1}^{N_m} E\left[\frac{\partial F(V_C)}{\partial A_k^m}\right] A_k^m\right) \rightarrow v_s \tag{8}$$

where N_m is the number of convolutional kernels, A_k^m represents the activation in the k-th channel of the m-th layer, C represents the class, $F(V_C)$ represents the score predicted by the neural network for class C, and E represents taking the average along the length direction of the activation A_k^m. Then the importance score of the signal can be calculated as:

$$I_{\mathrm{GC}}(P_i) = \sum_{s=1}^{s_n}\left(I_{\mathrm{GC}}\left(v_s^{P_i}\right)\right) \tag{9}$$

Guided Grad CAM.

Guided Grad CAM builds upon Grad CAM by incorporating the gradient information from the guided backpropagation gradients (GBP)[53]:

$$I_{\mathrm{GGC}}(P_i)=\sum_{s=1}^{s_n}\left(I_{\mathrm{GC}}\left(v_s^{P_i}\right)\cdot \mathrm{GBP}\left(v_s^{P_i}\right)\right) \tag{10}$$

Deep Grad CAM. Deep Grad CAM[40] utilizes the backward propagation mechanism of convolutional layers to transmit the importance vector of the last convolutional layer to the input layer, replacing the linear mapping used in Grad CAM. The propagation rules between the m-th and $(m+1)$-th layers can be expressed as follows:

$$I_{\mathrm{DGC}}^{m}\left(x_{k,i}^{m}\right)=a_{k,i}^{m}\sum_{j}\frac{w_{ij}^{+}}{\sum_{i}a_{k,i}^{m}w_{ij}^{+}+b_{j}^{+}}I_{\mathrm{DGC}}^{m+1}\left(x_{u,j}^{m+1}\right) \tag{11}$$

where $a_{k,i}^{m}$ represents the activation value at the i-th point in the k-th channel of the m-th layer. w_{ij}^{+} and b_{j}^{+} are the positive parts of the network weights and biases, respectively. The importance score of the signal is:

$$I_{\mathrm{DGC}}(P_i)=\sum_{s=1}^{s_n}I_{\mathrm{DGC}}^{(1)}\left(v_s^{P_i}\right) \tag{12}$$

Feature sensitivity

For the reason that different XAI algorithms yield varying importance score results, a new question arises: which XAI algorithm's results should be adopted? To address this question, the FS evaluation method is proposed. This method utilizes the changes in testing accuracy to assess the accuracy of XAI algorithms in the calculation of importance scores:

$$\mathrm{FS}=\frac{1}{N}\sum_{i=1}^{N}\delta\left(F(V_i),\hat{Y}_i\right)-\frac{1}{N}\sum_{i=1}^{N}\delta\left(F\left(V_i^{R_{n_f}}\right),\hat{Y}_i\right) \tag{13}$$

$$\delta(x,y)=\begin{cases}1 & x=y\\ 0 & x\neq y\end{cases} \qquad R_{n_f}=\operatorname{argmax}\left(\Phi(F),n_f\right) \tag{14}$$

where N is the total number of samples in the testing dataset, V_i is the i-th sample in the testing dataset, $F(V_i)$ is the predicted label for V_i, and $\hat{Y}_i$ is the true label. $V_i^{R_{n_f}}$ represents the noise sample, which is obtained by multiplying a random vector over the most important n_f features of the V_i vector, and n_f is the length of the important features. R_{n_f} is the index of the important features, Φ is the XAI function, and different XAI algorithms get different indices of the important features.

Therefore, $\frac{1}{N}\sum_{i=1}^{N}\delta\left(F(V_i),\hat{Y}_i\right)$ can denote the prediction accuracy on the original testing dataset, $\frac{1}{N}\sum_{i=1}^{N}\delta\left(F(V_i^{R_{n_f}}),\hat{Y}_i\right)$ can represent the prediction accuracy on the testing dataset with perturbations added to the important features. A higher FS indicates that when perturbations are added to the features considered important by the XAI algorithm, the testing accuracy is more significantly affected, and therefore these important features are more crucial for the 1D CNN classification model. Thus, FS can be used to evaluate the accuracy of the XAI algorithm in identifying the important features.

Loose bolt localization

Bolt-loosening database

An experimental study was conducted to detect a loose bolt in a double-layered aluminum plate with 16 bolted connections. The sensor array and bolt arrangement are shown in Figure 2(a), while the experimental setup is illustrated in Figure 2(b). The aluminum plates are Al-7075 with a Young's modulus of 71 GPa, a density of 2700 kg/m^3, and a Poisson's ratio of 0.33.[54] Each layer of aluminum plate has a thickness of 2 mm and dimensions of 400 × 400 mm. The Young's modulus of the bolts is 206 GPa, density is 7800 kg/m^3, and Poisson's ratio is 0.3.[55] Four piezoelectric ceramic transducers (PZT) with an 8 mm diameter and 16 steel bolts with 6 mm diameter are positioned on the aluminum plates, the coordinates of which are found in Table 1. The experimental setup consists of an arbitrary waveform generator (Agilent 33502A), a digital oscilloscope (Pico Scope 6402A), and an amplifier (Agilent 33502A).

A 17-class bolt-loosening database was first established. The first class was labeled as "Healthy," which represents the secure connections between the bolts and the plate, where a torque of 5 Nm was applied to each bolt using a torque wrench. The second to seventeenth classes were labeled as "Damage" 1–16, which represent the sequential loosening of bolt 1–16, respectively, where the torque on the bolt was removed. For each damage case, excitation was applied on one of the PZTs and signals were received at the remaining three PZTs, and then the excitation sensors were sequentially transformed so that a total of 12 sets of signals could be received for each damage case.

The sampled length of the received signals was 1500 data points and the total number of samples was 150, of which 100 sets were used to construct the training dataset and 50 sets were used to construct the testing dataset. Then a phase variation of 7% and an amplitude variation of 10% were added to the received signals to simulate the signals collected under different temperature conditions, corresponding to a temperature

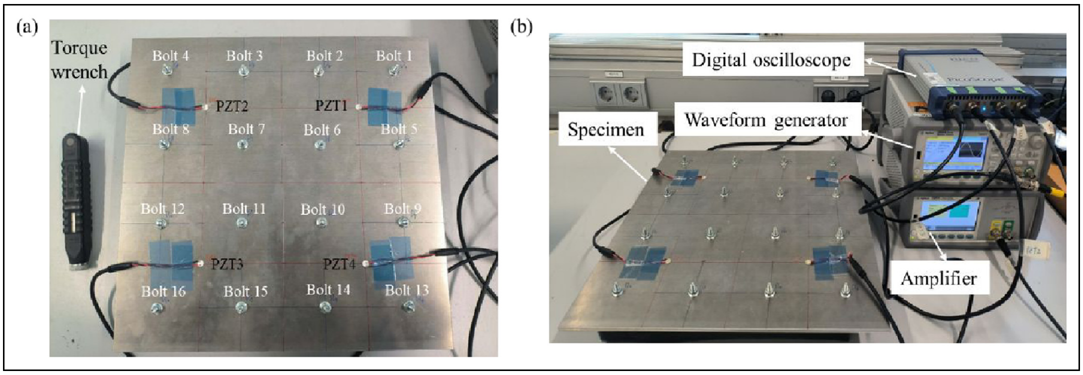


Figure 2. (a) Layout of the aluminum plate and transducer array and (b) experiment setup.

Table 1. Coordinates of bolts and PZT transducers.

Number	Bolt 1	Bolt 2	Bolt 3	Bolt 4
Coordinates (mm)	(150, 150)	(50, 150)	(−50, 150)	(−150, 150)
Number	Bolt 5	Bolt 6	Bolt 7	Bolt 8
Coordinates (mm)	(150, 50)	(50, 50)	(−50, 50)	(−150, 50)
Number	Bolt 9	Bolt 10	Bolt 11	Bolt 12
Coordinates (mm)	(150, −50)	(50, −50)	(−50, −50)	(−150, −50)
Number	Bolt 13	Bolt 14	Bolt 15	Bolt 16
Coordinates (mm)	(150, −150)	(50, −150)	(−50, −150)	(−150, −150)
Number	PZT 1	PZT 2	PZT 3	PZT 4
Coordinates (mm)	(100,100)	(−100,100)	(−100, −100)	(100, −100)

PZT: piezoelectric ceramic transducers.

variation range of 25°.[56,57] Consequently, a training dataset of size 12 × 1500 × 1700 and a testing dataset of size 12 × 1500 × 850 were obtained. Then the 12 sets of signals in each experiment were concatenated head-to-tail to form a 1D vector with lengths of 18,000. Consequently, the size of the training dataset is 1700 × 18,000 and the testing dataset is 850 × 18,000. Figure 3 displays training signal examples for the Healthy, Damage 1, and Damage 2 scenarios.

Dimensionality reduction based on XAI

Table 2 shows the detailed architecture of the 1D CNN classification model, which consists of 12 layers in total: (1) an input layer with a length of 18,000. (2) Three convolutional layers with output channels 64, 128, and 256, and kernel sizes 32, 16, and 8, respectively. To expedite the reduction of the feature vector's length, the stride was set to 6, and the max-pooling layer has a size of 2. (3) Two fully connected layers with 512 and 128 neurons, respectively. (4) An output layer with a length of 17.

The training and testing of the model were performed on PyTorch 1.11.0 package with Python 3.7.11 environment, and the hardware platform was a laptop with an AMD Ryzen 7 5800H, 16 GB of RAM, and an NVIDIA GeForce RTX 3050 Ti GPU. After the training, the classification model was applied to the testing dataset. The confusion matrix of the testing dataset is shown in Figure 4, and the overall predicting accuracy is 98.24%. However, the classification model could only provide the class label for the loose bolt, not the decision-making for physical information such as the location or the torque of the bolt, which makes this form of damage detection less intuitive.

Therefore, this study aims to utilize 1D CNN to predict the position and torque of the loose bolt. But in this scenario, the problem addressed by 1D CNN shifts from a classification problem to a regression problem, which leads to the more complex patterns between the input and the output that need to be learned. Consequently, there is a need to further increase the model complexity, which often comes at a higher computational cost. The training iteration curves for the classification and regression models are shown in Figure 5. The models are considered to converge when the slope of the loss curve is close to 0. So, it can be observed that the classification model converges around 1000 epochs, and the regression model converges around 8000 epochs. The converging times are 20 and 400 s, respectively. Regression is 20 times more computationally expensive than classification. Moreover, this

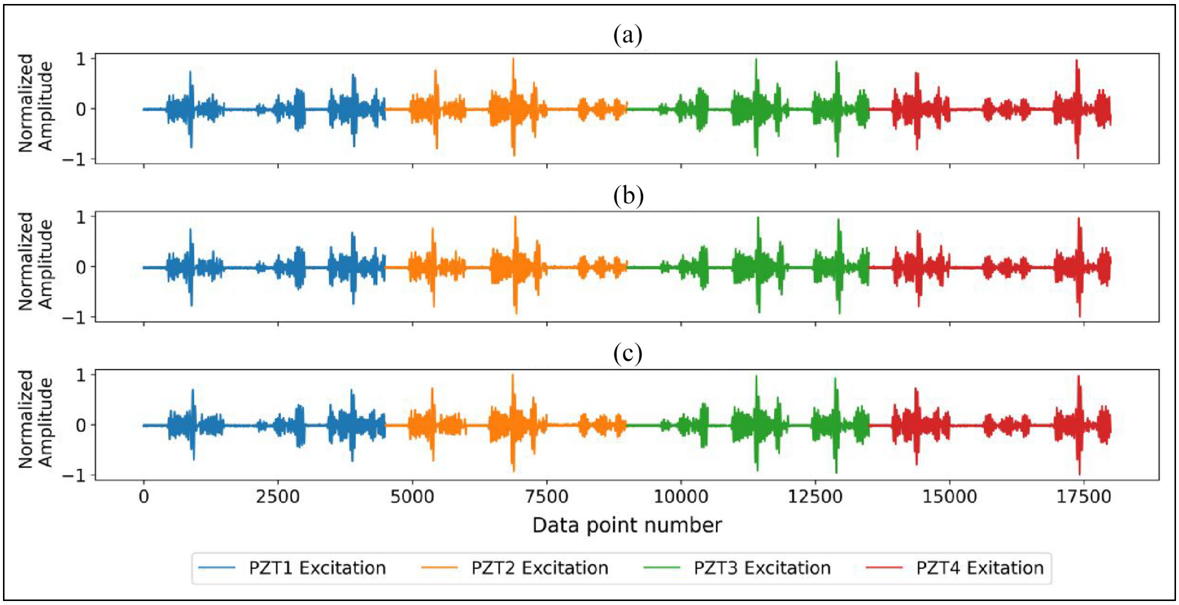


Figure 3. Input vector for 1D CNN (a) from health class in which all the bolts and plate were tightly connected, (b) from Damage 1 class in which bolt 1 was loosed, and (c) from Damage 2 class in which bolt 2 was loosed.
1D CNN: one-dimensional convolutional neural network.

Table 2. Architecture of the 1D CNN classification model.

#	Layer type	Output dimension
0	Input	1 × 18,000
1	Convolution 1D	64 × 2995
2	Max pooling 1D	64 × 1497
3	Convolution 1D	128 × 247
4	Max pooling 1D	128 × 123
5	Convolution 1D	256 × 20
6	Max pooling 1D	256 × 10
7	Fully connected layer	1 × 512
8	ReLu	1 × 512
9	Fully connected layer	1 × 128
10	ReLu	1 × 128
11	Output layer	1 × 17

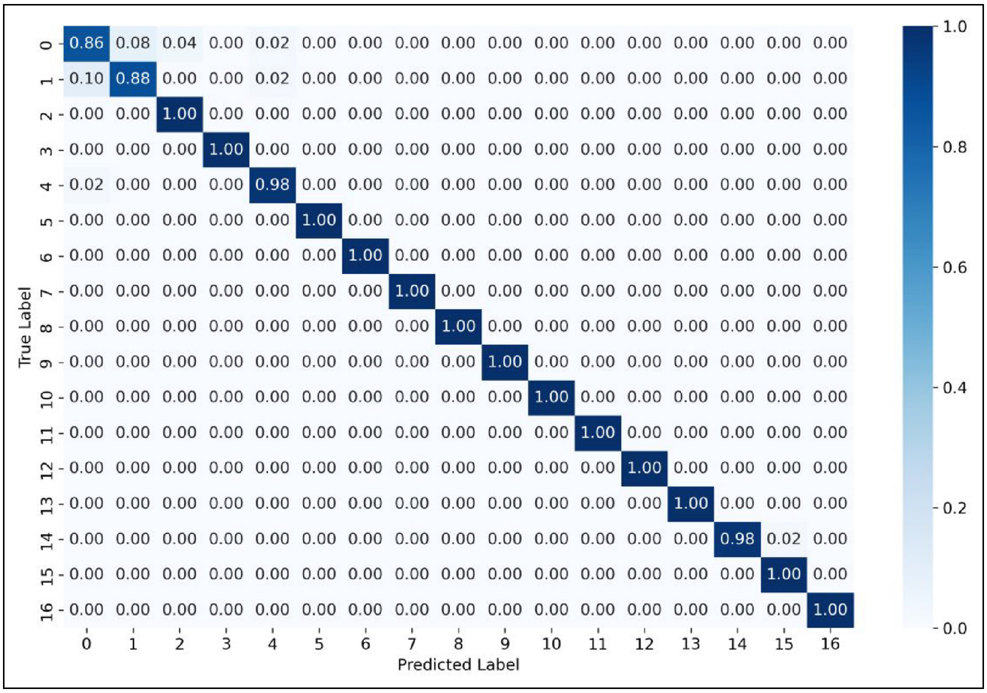
1D CNN: one-dimensional convolutional neural network.

is only for the application on a small 400 × 400 mm aluminum plate. When dealing with more complex, larger structures with more monitoring sensors, the computational cost can increase exponentially.

Therefore, appropriate dimensionality reduction of the input vector is essential for the regression model. Reduced input space leads to significantly lower model complexity, fewer parameters, and less computational costs. However, existing data dimensionality reduction methods like Deep Autoencoders[58] and Principal Component Analysis (PCA)[59] often destroy the structural characteristics of the signal that may be crucial for damage localization, making it challenging to ensure both effective dimensionality reduction and accurate localization.[60,61] But if the redundant signals among the 12 signals in the input vector can be removed and the most representative signal can be found to serve as the new input of the 1D CNN regression model, the dimension of the input vectors will be greatly reduced, while the structural features of the signals can still be preserved.

As shown in Table 3, each input vector consists of signals from 12 monitoring paths. By utilizing XAI to analyze the 1D CNN classification model, it can be determined which monitoring paths are primarily referenced by the 1D CNN during the decision-making process. Figure 6 shows an example of the explanation of the 1D CNN classification model using Smooth Simple Taylor, which is referred to as the importance-score Saliency map. There are two bright regions (P5 and P8) in the map, indicating these two regions have high importance scores and this also means Smooth Simple Taylor believes that the 1D CNN classification model has mainly referred to the information collected from P5 and P8 when determining this sample as the Damage 12 class.

The importance-score Saliency map for all 17 damage scenarios was calculated. Then the number of times that each path is considered to be the most important path under all damage scenarios can be counted. The path with the highest number of times can then be selected as the new input vector to the 1D CNN regression model, achieving the objective of dimensionality reduction for the raw input vector.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.86	0.08	0.04	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.10	0.88	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.02	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.02	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure 4. Confusion matrix on the testing dataset using 1D CNN.
1D CNN: one-dimensional convolutional neural network.

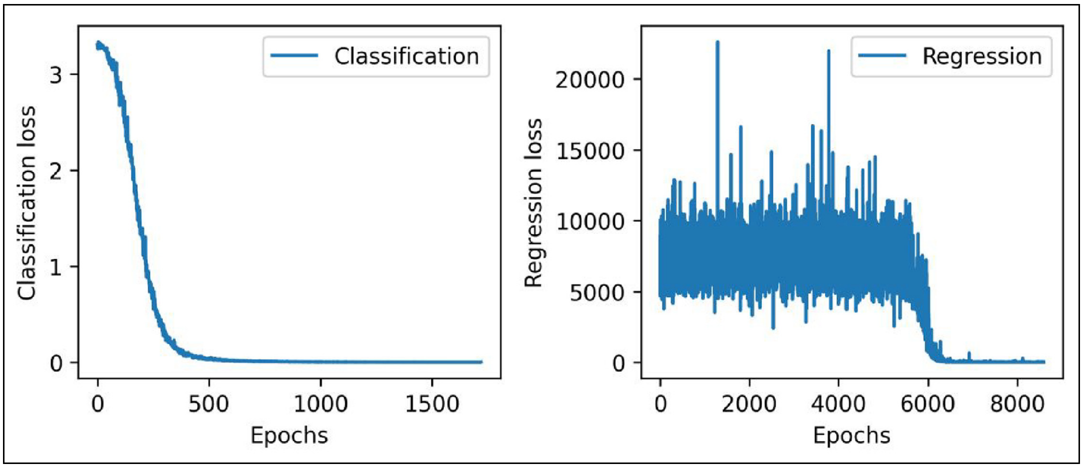


Figure 5. Comparison of training iteration curves between classification and regression model.

The ranking score for the most important path was calculated by six different XAI algorithms, sensitivity analysis (SA), Smooth simple Taylor (SST), deep Taylor (DT), Grad CAM (GCAM), Deep Grad CAM (DCAM), and Guided Grad CAM (GGCAM), which is shown in Figure 7. For the result of SA, P11 is considered the most important. SST identifies P3 as the most crucial. DT and GCAM both highlight P6 as the most important, while DCAM and GGCAM favor P5. These results demonstrate that different XAI algorithms provide varying recommendations for the new input vector.

Therefore, FS was used to evaluate the accuracy of these six XAI algorithms in the calculation of the importance scores of the signals. Figure 8(a) shows the change in testing accuracy of the 1D CNN classification model when adding random perturbations to the important features given by the six XAI algorithms with different feature lengths (n_f) ranging from 0 to 100. It can be observed that the testing accuracy

Table 3. Signals from different monitoring paths for each input sample.

P1: PZT1–PZT2	P2: PZT1–PZT3	P3: PZT1–PZT4
P4: PZT2–PZT1	P5: PZT2–PZT3	P6: PZT2–PZT4
P7: PZT3–PZT1	P8: PZT3–PZT2	P9: PZT3–PZT4
P10: PZT4–PZT1	P11: PZT4–PZT2	P12: PZT4–PZT3

PZT: piezoelectric ceramic transducers.

decreases most rapidly when changing the important features identified by SST. When the feature length reaches 100, the testing accuracy drops to 26.58% for SST, while for the other XAI algorithms remains above 75%. This indicates that the features identified by SST are the most critical for the 1D CNN's decision-making. Figure 8(b) shows the average FS values for the six XAI algorithms at different n_f. The average

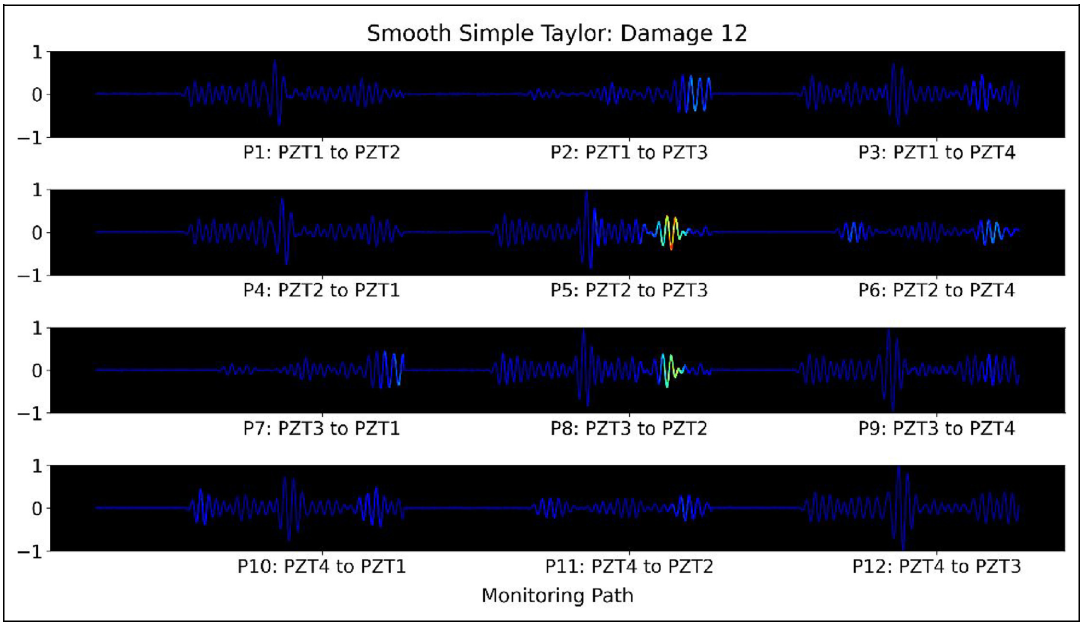


Figure 6. Importance-score Saliency map for the Damage 12 case from Smooth simple Taylor.

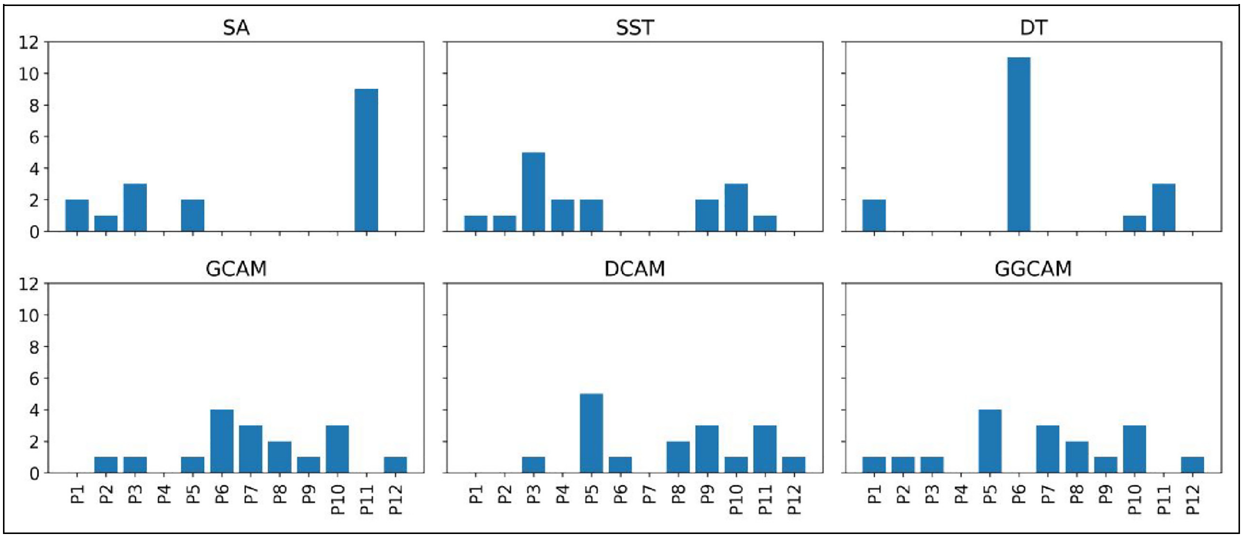


Figure 7. The ranking score for the most important paths was calculated by six different XAI algorithms: sensitivity analysis (SA), Smooth Simple Taylor (SST), deep Taylor (DT), Grad CAM (GCAM), Deep Grad CAM (DCAM), and Guided Grad CAM (GGCAM).
CAM: Class Activation Mapping; XAI: explainable artificial intelligence.

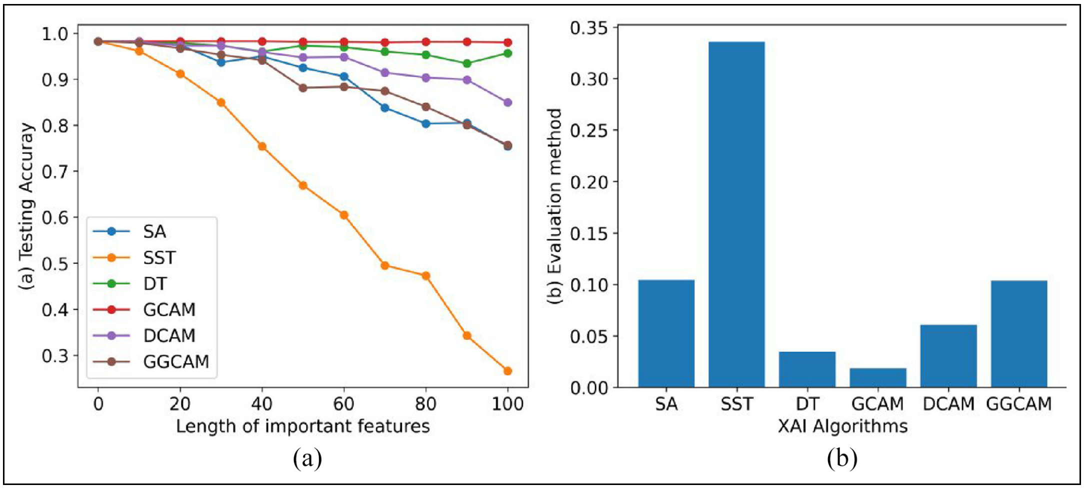


Figure 8. (a) Testing accuracy of 1D CNN classification model on six different feature lengths calculated by XAI algorithms and (b) mean value of FS for the XAI algorithms over six different feature lengths.
1D CNN: one-dimensional convolutional neural network; FS: feature sensitivity; XAI: explainable artificial intelligence.

FS value of SST is significantly higher than that of the other algorithms. Therefore, it can be known that the FS suggests choosing the most important path given by the SST algorithm as the new input vector for the 1D CNN regression model.

Regression result of bolt-loosening database

In this section, the X1D-CNN, RAN-1D-CNN, RI-1D-CNN, and DAE-1D-CNN regression models are, respectively, constructed. The X1D-CNN uses the most important signal determined by each XAI algorithm as input. For example, P11 is used as the input in SA, P3 is used in SST; The RAN-1D-CNN randomly selects a signal from the raw input vector; the RI-1D-CNN uses the raw input vector; the DAE-1D-CNN uses 1500 features obtained from the self-encoding of deep autoencoders as input.

The architectures of X1D-CNN, RAN-1D-CNN, and DAE-1D-CNN are the same, with details in Table 4, which consists of 18 layers in total: (1) an input layer with a length of 1500. (2) Three convolutional layers with output channels 64, 128, and 256, and kernel sizes 16, 16, and 8, respectively. The stride was set to 3, the max-pooling layer has a size of 2, and each convolutional layer employs ReLU and Batch Normalization. (3) Two fully connected layers with 512 and 128 neurons, respectively. (4) An output layer with a length of 2.

The RI-1D-CNN includes an additional fully connected layer, with detail in Table 5, which consists of 20 layers in total: (1) An input layer with a length of 18,000. (2) Three convolutional layers with output channels 64, 128, and 256, and kernel sizes 16, 16, and 8, respectively. The stride was set to 4, the max-pooling layer has a size of 2, and each convolutional layer employs ReLU and batch normalization. (3) Three fully connected layers with 4096, 2048, and 512 neurons, respectively. (4) An output layer with a length of 2.

Table 4. The architecture of the X1D-CNN, RAN-1D-CNN, and DAE-1D-CNN regression models.

#	Layer type	Output dimension
0	Input	1 × 1500
1	Convolution 1D	64 × 495
2	Batch normalization 1D	64 × 495
3	ReLu	64 × 495
4	Max pooling 1D	64 × 247
5	Convolution 1D	128 × 78
6	Batch normalization 1D	128 × 78
7	ReLu	128 × 78
8	Max pooling 1D	128 × 39
9	Convolution 1D	256 × 11
10	Batch normalization 1D	256 × 11
11	ReLu	256 × 11
12	Max pooling 1D	256 × 5
13	Fully connected Layer	1 × 512
14	ReLu	1 × 512
15	Fully connected layer	1 × 128
16	ReLu	1 × 128
17	Output layer	1 × 2

X1D-CNN: XAI-based regression models; RAN-1D-CNN: random-1D CNN regression model; DAE-1D-CNN: deep autoencoders-regression model; 1D CNN: one-dimensional convolutional neural network; XAI: explainable artificial intelligence.

Figure 9 compares the localization results of nine different regression models for 16 different damage scenarios. The XAI-based regression models (X1D-CNN) use the most important signal determined by each XAI algorithm as input. For example, P11 is used as the input in

Table 5. The architecture of the RI-1D-CNN regression model.

#	Layer type	Output dimension
0	Input	1 × 18,000
1	Convolution 1D	64 × 4497
2	Batch normalization 1D	64 × 4497
3	ReLu	64 × 4497
4	Max pooling 1D	64 × 2248
5	Convolution 1D	128 × 559
6	Batch normalization 1D	128 × 559
7	ReLu	128 × 559
8	Max pooling 1D	128 × 279
9	Convolution 1D	256 × 68
10	Batch normalization 1D	256 × 68
11	ReLu	256 × 68
12	Max pooling 1D	256 × 34
13	Fully connected layer	1 × 4096
14	ReLu	1 × 4096
15	Fully connected layer	1 × 2048
16	ReLu	1 × 2048
17	Fully connected layer	1 × 512
18	ReLu	1 × 512
19	Output layer	1 × 2

RI-1D-CNN: raw input-regression model; 1D CNN: one-dimensional convolutional neural network; ReLu: linear rectification activation function.

SA, P3 is used in SST; The random-1D CNN regression model (RAN-1D-CNN) randomly selects a signal from the raw input vector; the raw input-regression model (RI-1D-CNN) uses the raw input vector; the deep autoencoders-regression model (DAE-1D-CNN) uses1500 features obtained from the self-encoding as input.

In the figure, the true damage locations are marked with red crosses, while the predictions of various regression models for the positions of the loose bolts of the testing dataset are represented by blue scatters, and the predictions are connected with the actual locations by lines. From the graph, it is evident that SST and RI exhibit the best localization results, the 16 damage cases can be clearly distinguished from each other, and the distribution of the scatters in all the clusters is relatively concentrated, which indicates the predictions of each damage case are very close.

On the other hand, SA, DT, DCAM, and GCAM show some larger prediction errors in the cases of Damage 1, 2, 5, and 6. The predictions from RAN are more scattered, and there is a tendency for adjacent damage locations to overlap. DAE has the worst result, as it struggles to determine the positions of the loose bolts, which proves that if the signal structure is disrupted during the dimensionality reduction process, it can adversely affect localization accuracy.

Figure 10 compares the training iteration curves of nine different regression models. SA, SST, DT, and GCAM reached convergence at 4000 iterations. DCAM, GGCAM, and RI reached convergence at 6000 iterations. DAE converged at 1500 iterations. RAN showed the poorest performance and did not fully converge even after 6000 iterations.

Figure 11(a) compares the average Euclidean distances between the predicted and the true locations of the loose bolt for nine different regression models. It can be observed that SST and RI have the smallest Euclidean distances, measuring 5.81 and 6.19 mm, respectively. DAE has the largest Euclidean distance at 59.22 mm, while the errors for the other XAI-based regression models are all less than 16 mm. Figure 11(b) compares the testing loss for each model, showing a similar trend to Figure 11(a). This indicates that the results obtained by XAI-based dimensionality reduction are superior to those obtained using deep autoencoders. Retaining the complete signal structure characteristics is beneficial for loose bolt localization.

Table 6 provides a comparison of nine regression models, including their parameter number, converging time, Euclidean distances, and testing loss. As shown in the table, the parameter number for the X1D-CNN model remains consistent at 1,117,378, while RI-1D-CNN has the highest at 45,492,034, and DAE-1D-CNN is at 1,379,782. As for the converging time, SA, SST, DT, and GCAM exhibit faster convergence speeds, reaching convergence within 20 s. DCAM and GGCAM converge at 40 s, RAN reaches convergence at 50 s, while RI takes 200 s to converge, and DAE is the slowest, converging at 500 s. Therefore, it is shown that X1D-CNN regression models converge much faster than RI-1D-CNN and DAE-1D-CNN. For inference time, the RI-1D-CNN is 0.233 s, while the inference times for X1D-CNN, RAN-1D-CNN, and DAE-1D-CNN are 0.036 s. The inference time for RI-1D-CNN which uses the raw signal as input is 6.47 times longer than that of the other models.

Combining the results from Figure 11, it can be known that the SST-regression model has the shortest convergence time and the highest localization accuracy, which aligns with the predictions of the FS method. Therefore, selecting an XAI algorithm using the FS method and then using that XAI algorithm for dimensionality reduction can significantly reduce the model training time while ensuring the accuracy of the loose bolt localization. This also means that the FS-X1D-CNN method can achieve the precise monitoring of all 16 bolts with only one signal from the P3 path.

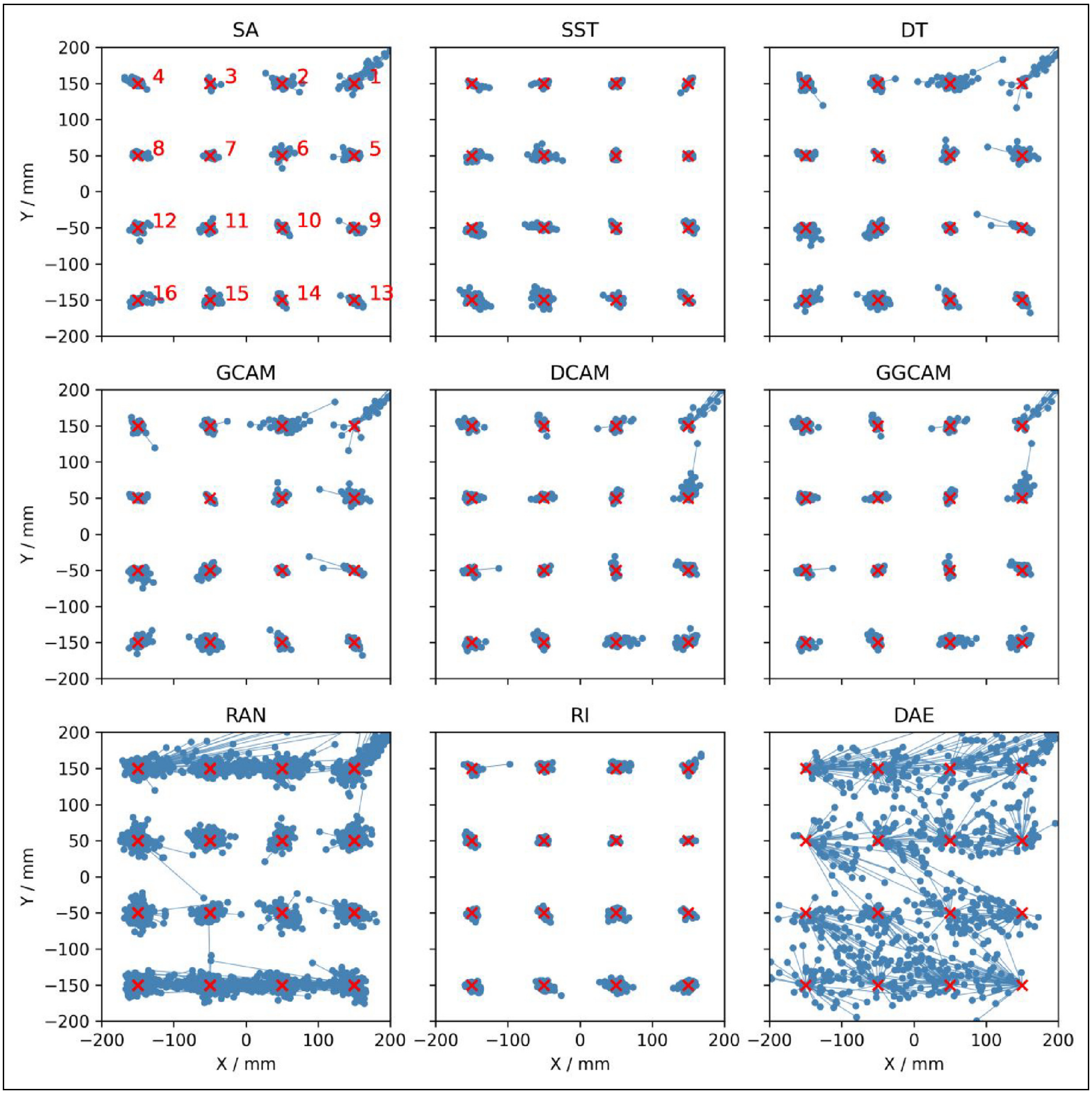


Figure 9. Localization results of 16 different damage scenarios were calculated by nine different regression models.

Loose bolt localization and torque prediction

Torque-predicting database

The experimental setup and specimen are identical to those described in the section "Bolt-loosening database." An additional 10 classes of torque variations are added to the bolt-loosening database: The 18th to 22nd classes were labeled Damage 17–Damage 21, representing torque values of 1, 2, 3, 4, and 5 Nm applied to bolt 6, respectively. The 23rd to 27th classes were labeled Damage 22–Damage 26, which represent the torque on bolt 10 were 1, 2, 3, 4, and 5 Nm, respectively. Therefore, the torque-predicting database consists of a total of 27 classes. The training dataset size is 2700 × 18,000, and the testing dataset size is 1350 × 18,000.

Dimensionality reduction based on XAI

First, a 1D CNN classification model was used to classify the 27 damage classes, and 97.11% was achieved in the testing dataset. Then, XAI algorithms were applied to analyze the 1D CNN classification model. Figure 12 shows the ranking score of the most important paths obtained by the XAI algorithms. SA, SST, and DT provided the same answer: P10 is the most important path. GCAM considered P11 the most important, DCAM

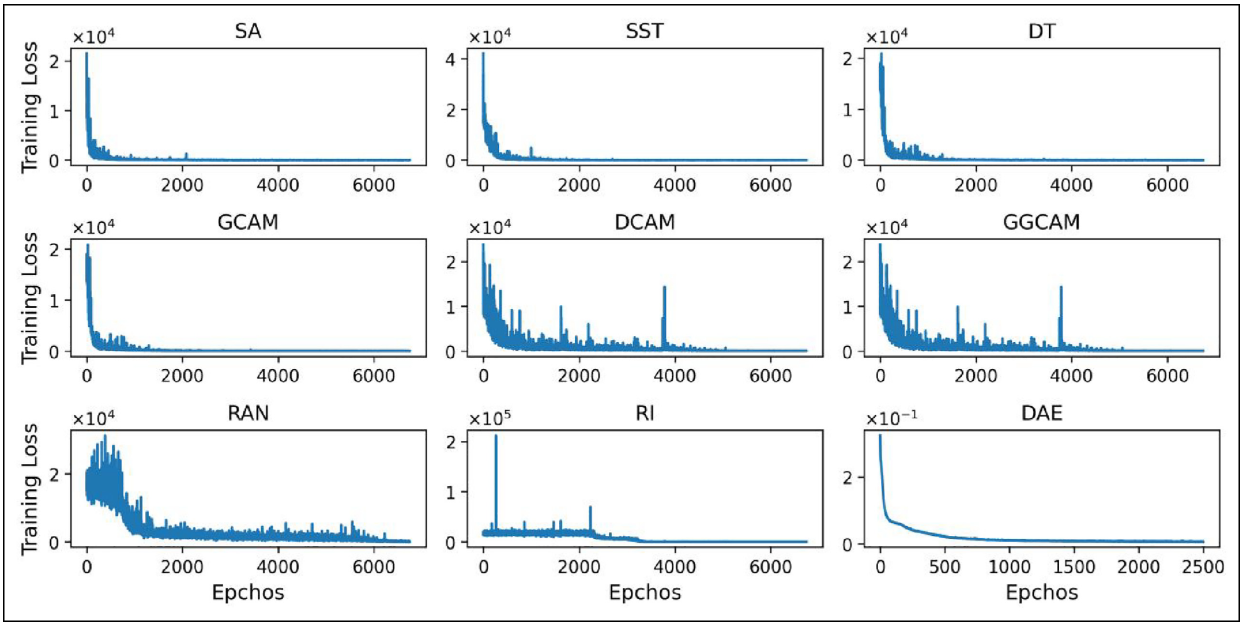


Figure 10. Training iteration curves for different regression models.

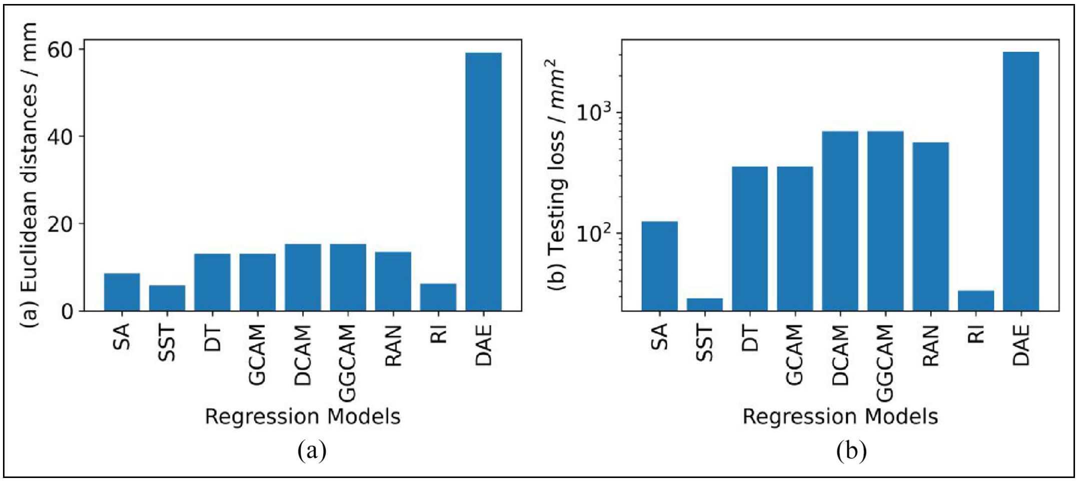


Figure 11. Localization error for different regression models: (a) Euclidean distance and (b) testing loss.

Table 6. The comparison of parameters number, converging time, Euclidean distance, and testing loss between different regression models.

Model name	Parameters number	Converging time (s)	Inference time (s)	Euclidean distance (mm)	Testing loss (mm^2)
SA	1,117,378	20	0.036	8.57	124.64
SST	1,117,378	20	0.036	5.81	28.84
DT	1,117,378	20	0.036	13.03	355.96
GCAM	1,117,378	20	0.036	13.03	355.96
DCAM	1,117,378	40	0.036	15.31	693.49
GGCAM	1,117,378	40	0.036	15.31	693.49
RAN	1,117,378	50	0.036	13.51	563.60
RI	45,492,034	200	0.233	6.19	33.50
DAE	1,379,782	500	0.036	59.22	3125.86

CAM: class activation mapping; SA: sensitivity analysis; SST: smooth simple Taylor; DT: Deep Taylor; GCAM: Grad CAM; DCAM: Deep Grad CAM; GGCAM: Guided Grad CAM; RAN: random; RI: raw input; DAE: deep autoencoders.

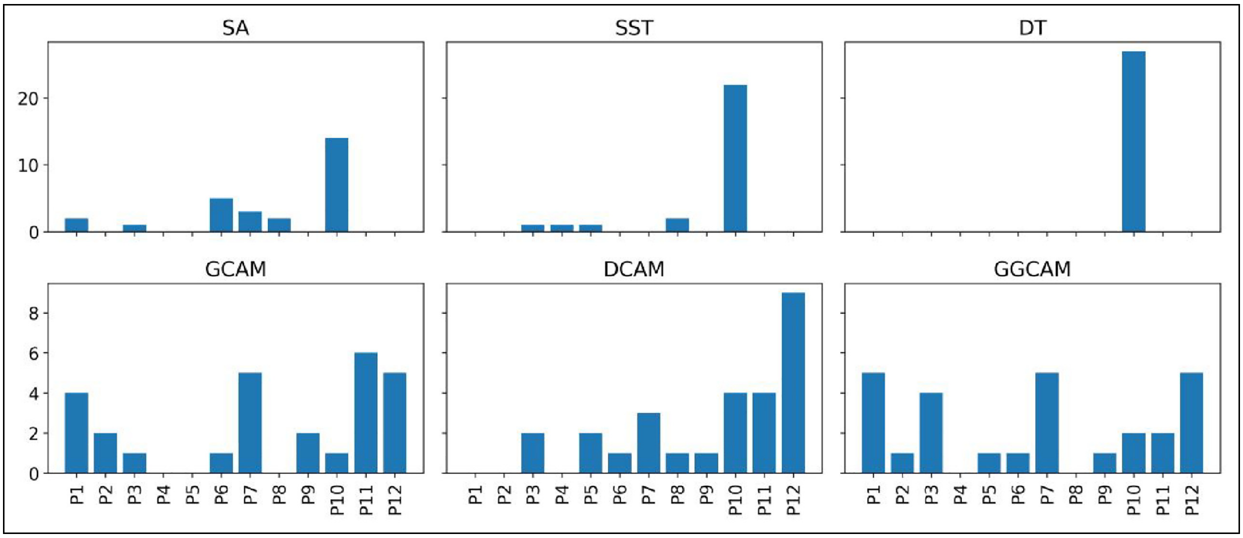


Figure 12. The ranking score for the most representative paths is calculated by six different XAI algorithms in the torque-predicting database.
XAI: explainable artificial intelligence.

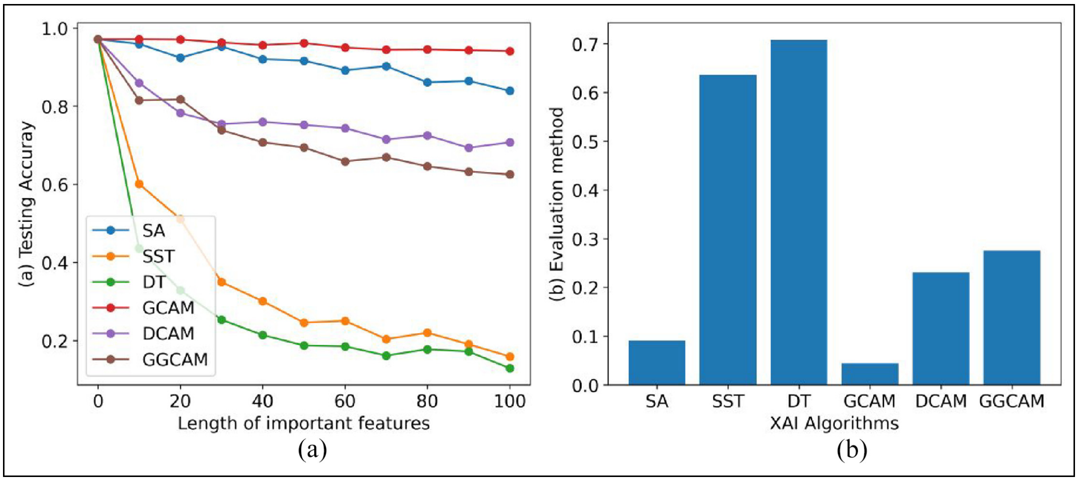


Figure 13. (a) Testing accuracy of the 1D CNN classification model on six different feature lengths calculated by XAI algorithms and (b) mean value of FS for the XAI algorithms over six different feature lengths.
1D CNN: one-dimensional convolutional neural network; FS: feature sensitivity; XAI: explainable artificial intelligence.

found P12 to be the most important, and in GGCAM, P1, P7, and P12 were tied for first place. Eventually, P1 was chosen as the best path by GGCAM.

Figure 13(a) shows the change in testing accuracy of the 1D CNN classification model when adding random perturbations to the important features given by the six XAI algorithms with different feature lengths (n_f) ranging from 0 to 100. It can be observed that DT and SST have similar performance. Changing the important features identified by these two algorithms results in the fastest drop in testing accuracy. When the feature length reaches 100, the prediction accuracy drops to 12.89% and 15.93%, respectively, while the other XAI algorithms can still remain above 60%. Therefore, in this database, the important features identified by DT and SST are the most crucial for the decision-making of the 1D CNN classification model.

Figure 13(b) displays the average FS values for XAI algorithms across six different n_f. DT has the highest average FS value, followed by SST. The average FS values are much higher than those of the other algorithms. Therefore, FS suggests the most important path given by the DT or SST as the input of the 1D CNN regression model.

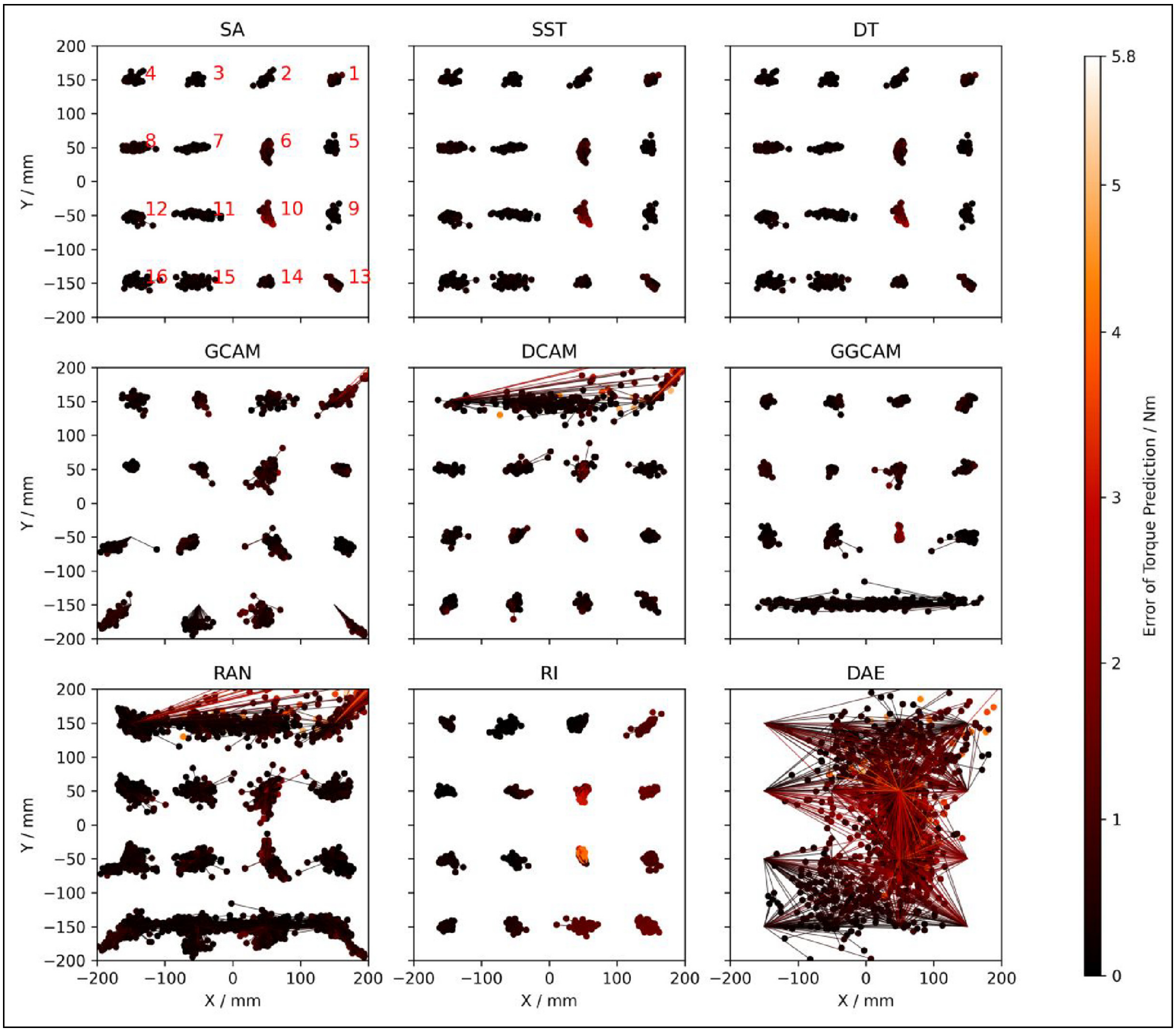


Figure 14. Loose bolt localization and torque prediction of 26 different damage scenarios calculated by nine different regression models.

Regression result of torque prediction database

Figure 14 shows the localization and torque prediction of the loose bolt by nine different regression models for 27 damage scenarios. In the figure, the predictions for the positions and torques of the loose bolts are represented by scatter points from black to red. The closer the color is to red, the larger the prediction error, while the closer it is to black, the smaller the prediction error.

It can be observed that SA, SST, and DT provide the most accurate predictions. The scatter points for the 27 different damage scenarios are distinct, and the scatter points in all the clusters are relatively concentrated, which suggests that the predictions for each damage scenario are close to the true locations. In addition, except for the clusters on bolt 10, where the color is close to red, the color of all other clusters is close to black. This signifies that, apart from the torque prediction error of approximately 3 Nm on bolt 10, the torque prediction error on all other bolts is close to 0 Nm.

Regarding the results for other algorithms, RI provides an accurate localization of the loose bolts. However, it exhibits larger torque prediction errors for bolts 6 and 10, shown in bright red. Clusters on bolts 1, 5, 9, 13, and 14 are shown in dark red. GCAM has some deviations in the predictions for loose bolts 1, 2, 12, and 13. DCAM's predictions for loose bolts 1–4 are not ideal, with some overlap in the predicted positions. GGCAM shows overlapping predictions for bolts 13–16. RAN's predictions are quite dispersed, and neighboring damage scenarios tend to overlap. DAE performs the worst, with almost no capability to determine the loose bolt's position and significant errors in torque value prediction.

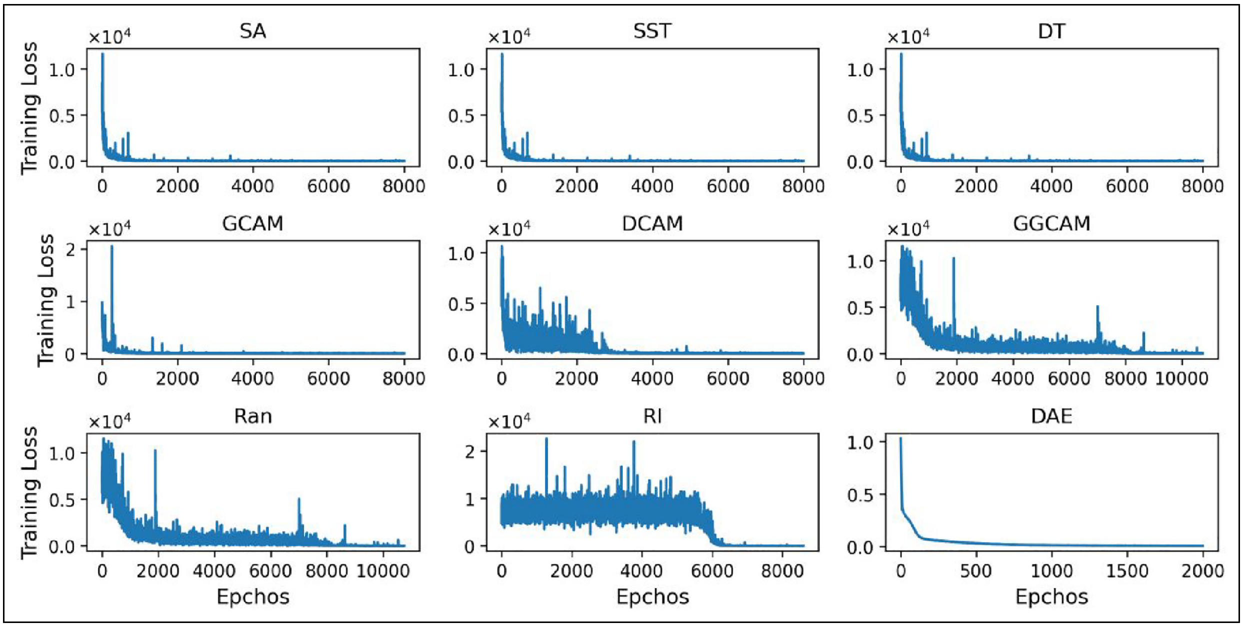


Figure 15. Training iteration curves for different regression models.

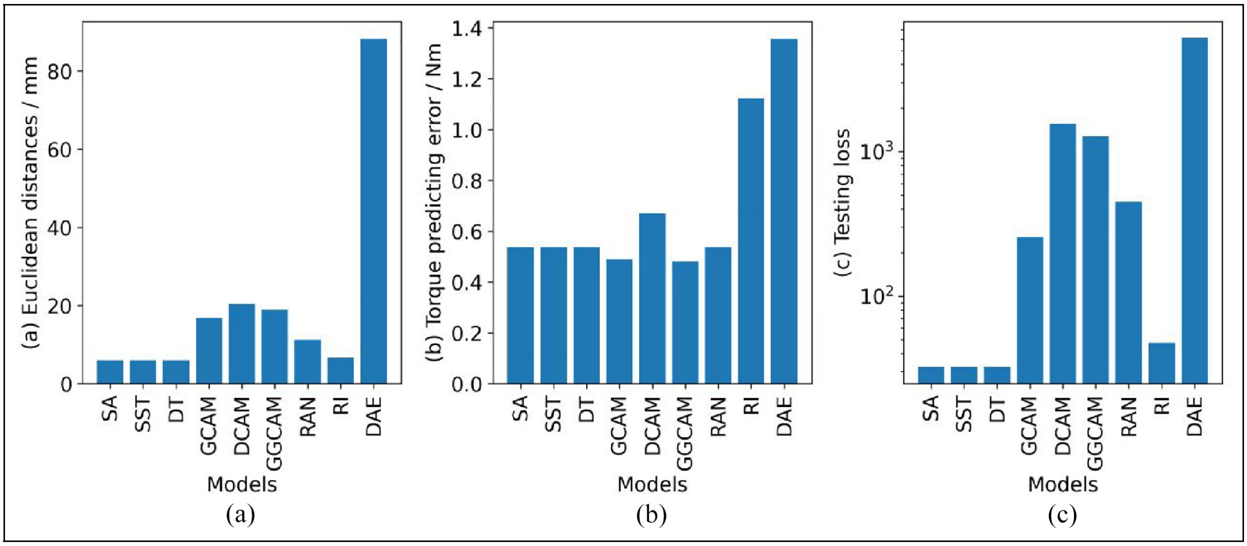


Figure 16. Localization error for different regression models: (a) Euclidean distance, (b) torque-predicting error, and (c) testing loss.

Figure 15 compares the training iteration curves of nine different regression models. SA, SST, DT, GCAM, and DCAM reached convergence at 6000 iterations. RI reached convergence at 8000 iterations. DAE converged at 1000 iterations. RAN and GGCAM showed the poorest performance and did not fully converge even after 10,000 iterations.

Figure 16(a) compares the average Euclidean distance between the predictions and the true locations of the loose bolts for nine different models. It shows that the Euclidean distance of SA, SST, and DT is the smallest, which is 5.95 mm. RI is followed by 6.64 mm. DAE has the largest Euclidean distance at 88.32 mm, while the other XAI-based regression models have errors of no more than 21 mm. Figure 16(b) compares the torque-predicting errors of each model. The errors of GCAM and GGCAM are very close, with values of 0.49 and 0.48 Nm, respectively. Followed by SA, SST, DT, and RAN at 0.54 Nm, the error of RI is 1.12, and the error of 1.36 Nm for DAE is the highest.

Table 7. The comparison of parameters number, converging time, Euclidean distance, and testing loss between different regression models.

Model name	Parameters number	Converging time (s)	Inference time (s)	Euclidean distance (mm)	Torque-predicting error (Nm)	Testing loss (mm^2)
SA	1,117,507	40	0.045	5.95	0.54	32.25
SST	1,117,507	40	0.045	5.95	0.54	32.25
DT	1,117,507	40	0.045	5.95	0.54	32.25
GCAM	1,117,507	40	0.045	16.78	0.49	254.76
DCAM	1,117,507	40	0.045	20.40	0.67	1563.40
GGCAM	1,117,507	80	0.045	17.21	0.53	1189.80
RAN	1,117,507	80	0.045	11.25	0.55	444.63
RI	45,492,547	400	0.321	6.64	1.12	47.42
DAE	1,379,911	600	0.045	88.32	1.36	6117.51

CAM: class activation mapping; SA: sensitivity analysis; SST: smooth simple Taylor; DT: Deep Taylor; GCAM: Grad CAM; DCAM: Deep Grad CAM; GGCAM: Guided Grad CAM; RAN: random; RI: raw input; DAE: deep autoencoders.

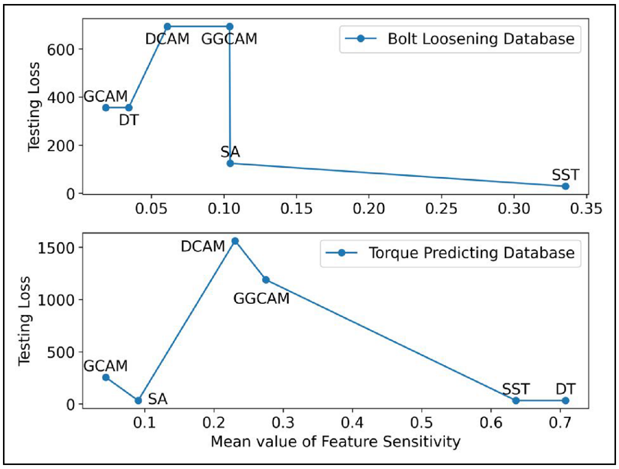


Figure 17. Relationship between the mean value of feature sensitivity and testing loss.

Figure 16(c) compares the testing loss for each model, which combines both localization and torque prediction errors. It shows that SA, SST, and DT have the lowest testing loss, measuring 32. RI follows with a testing loss of 47.42, while DAE has the highest testing loss at 6117.51. Overall, considering both types of errors, SA, SST, and DT perform the best in terms of prediction accuracy.

Table 7 provides a comparison of nine regression models, including their parameters number, converging time, Euclidean distances, torque-predicting error, and testing loss. As shown in the table, the parameter number for the X1D-CNN model remains consistent at 1,117,507, while RI-1D-CNN has the highest at 45,492,547, and DAE-1D-CNN is at 1,379,911. As for the converging time, SA, SST, DT, GCAM, and DCAM exhibit faster convergence speeds, reaching convergence within 40 s. GGCAM and RAN have intermediate regression speeds, reaching convergence at around 80 s, while RI takes 400 s to converge, and DAE is the slowest, converging at 600 s. Summarizing XAI-based regression models converges much faster than using raw input and deep autoencoders. For inference time, the RI-1D-CNN is 0.321 s, while the inference times for X1D-CNN, RAN-1D-CNN, and DAE-1D-CNN are 0.045 s. The inference time for RI-1D-CNN which uses the raw signal as input is 7.13 times longer than that of the other models.

Combining the results from Figure 16, it can be known that the SA, SST, and DT models have the shortest convergence time and the highest positioning accuracy, which aligns well with the FS method predictions. Because in the FS evaluation results, the FS values of SST and DT are much higher than those of other algorithms. Therefore, in the torque-predicting database, it is possible to achieve precise monitoring of all 16 bolts and accurate torque prediction using signals from just the P10 path.

Discussion

The research findings above indicate that the FS-X1D-CNN can realize the positioning and torque prediction of loose bolts in 16-bolted plates with only a single lamb wave signal under the condition of temperature change. However, the current study does not consider other influences apart from temperature variation. In practical engineering, environmental factors are much more complex, including vibration or electromagnetic effect generated by machine operation, external structural loads, and variations in sensor performance during service, etc.[7] Therefore, more realistic conditions need to be taken into account in the future research to further improve the robustness of FS-X1D-CNN.

Figure 17 shows the relationship between the mean value of FS and testing loss on the bolt-loosening database and torque-predicting database. it can be seen that the FS method proposed in this paper can accurately select the best-performing XAI algorithm on different datasets. For instance, in the bolt-loosening database, the FS value for SST is significantly higher than that of other algorithms, and the SST-regression model exhibits the lowest localization error. In the torque-predicting database, SST and DT have FS values that are close and significantly higher than other algorithms. Ultimately, both the SST-regression model and DT-regression model achieve the same level of performance, with the highest accuracy among all regression models.

However, FS cannot provide a ranking of XAI algorithms that perfectly match their predictive performance. For example, in the bolt-loosening database, the testing loss of GCAM and DT are smaller than DCAM and GGCAM, but their FS values are smaller. In the torque-predicting database, the testing loss level of the SA-regression model is on par with SST and DT, but SA does not score well in FS. Therefore, it should be noted that FS cannot identify all potentially high-performing XAI algorithms, and it cannot predict the precise ranking of all XAI algorithms. Part of the reason for this may stem from the fact that XAI's choice of finding the most important signal is limited, as XAI algorithms pick one signal from a finite set of 12 signals. Such selection is discrete, which may lead to difficulties in effectively distinguishing the performance of different XAI algorithms.

Therefore, while FS can currently select the best XAI algorithm for different datasets, there is still a need to further improve the accuracy of performance evaluation of different XAI algorithms. In addition, FS-X1D-CNN can help improve the efficiency of bolt monitoring. For structures with multiple bolt connections, there are numerous degrees of loosening damage for each bolt. Collecting, transmitting, and analyzing signals for each damage scenario would be time-consuming and labor-intensive. FS-1D-CNN narrows down the monitoring scope to specific sensors, greatly enhancing the efficiency of data analysis work.

Conclusion

This paper proposed an FS-X1D-CNN model, which utilizes XAI algorithms for the dimensionality reduction of the input vector for the 1D CNN regression model and uses the FS evaluation method to select the most accurate XAI algorithm. This approach has successfully reduced the training time of the 1D CNN regression models 10-fold while ensuring the highest detection accuracy on both the loose bolt database and the torque-predicting database. The conclusions are as follows:

1. FS can be utilized for the selection of XAI algorithms. FS successfully predicted that SST performs best in the loose bolt database, while DT performs best in the torque-predicting database. Therefore, selecting XAI algorithms based on FS values can obtain the best-performing regression model.
2. FS-X1D-CNN model demonstrates advantages in loose bolt localization and torque prediction. In the loose bolt database, the FS-X1D-CNN model achieves a convergence time of 20 s, and a localization error of 5.81 mm, which outperforms the RI-1D-CNN model with 200 s and 6.19 mm. In the torque-predicting database, the FS-X1D-CNN model has a convergence time of 40 s, a localization error of 5.95 mm, and a torque prediction error of 0.54 Nm, which is superior to the RI-1D-CNN model with 400 s, 6.64 mm, and 1.12 Nm.
3. Preserving the complete structure of the signal during data dimensionality reduction is crucial. The DAE-1D-CNN model has errors of 59.22 mm in the loose bolt database and 88.32 mm and 1.36 Nm in the torque-predicting database. These errors are significantly larger than those of other regression models. This is because the autoencoding process of deep autoencoders disrupts the complete structure of the signals, leading to the loss of information in the time series.

Author contributions

Muping Hu: Conceptualization, Methodology, Software, Investigation, Writing—Original Draft. Nan Yue: Conceptualization, Methodology, Investigation, Writing—Review & Editing, Supervision. Roger M. Groves: Conceptualization, Methodology, Resources, Writing—Review & Editing, Funding acquisition.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the China Scholarship Council (No. 202206680031); and the European Union Horizon Europe OVERLEAF project (No. 101056818).

ORCID iDs

Muping Hu https://orcid.org/0000-0002-4133-4088
Roger M. Groves https://orcid.org/0000-0001-9169-9256

References

1. Bickford JH and Oliver M. *Introduction to the design and behavior of bolted joints: non-gasketed joints*. Boca Raton: CRC Press, 2022.
2. Gong H, Deng X, Liu J, et al. Quantitative loosening detection of threaded fasteners using vision-based deep learning and geometric imaging theory. *Automat Constr* 2022; 133: 104009.
3. Kaminskaya VV and Lipov AV. Self-loosening of bolted joints in machine tools during service. *Soviet Eng Res* 1990; 10: 81–85.
4. Dawn R. *National transportation safety board marine accident brief*, No. DCA18FM009, USA, 6 February 2019.
5. Liberty C. *National transportation safety board marine accident brief*, No. DCA15FM035, USA, 21 June 2017.
6. Yang Z, Yang H, Tian T, et al. A review on guided-ultrasonic-wave-based structural health monitoring: from fundamental theory to machine learning techniques. *Ultrasonics* 2023; 133: 107014.
7. Gorgin R, Luo Y and Wu Z. Environmental and operational conditions effects on lamb wave based structural health monitoring systems: a review. *Ultrasonics* 2020; 105: 106114.
8. Zhuang Y, Kopsaftopoulos F, Dugnani R, et al. Integrity monitoring of adhesively bonded joints via an electromechanical impedance-based approach. *Struct Health Monit* 2018; 17: 1031–1045.
9. Wang F, Ho SCM, Huo L, et al. A novel fractal contact-electromechanical impedance model for quantitative monitoring of bolted joint looseness. *IEEE Access* 2018; 6: 40212–40220.
10. Amerini F, Barbieri E, Meo M, et al. Detecting loosening/tightening of clamped structures using nonlinear vibration techniques. *Smart Mater Struct* 2010; 19: 085013.
11. Li QK and Jing XJ. Fault diagnosis of bolt loosening in structures with a novel second-order output spectrum-based method. *Struct Health Monit* 2020; 19: 123–141.
12. Argatov I and Sevostianov I. Health monitoring of bolted joints via electrical conductivity measurements. *Int J Eng Sci* 2010; 48: 874–887.
13. Sui X, Duan Y, Yun C, et al. Bolt looseness detection and localization using wave energy transmission ratios and neural network technique. *J Infrastruct Intell Resilience* 2023; 2: 100025.
14. Du F, Wu S, Sheng R, et al. Investigation into the transmission of guided waves across bolt jointed plates. *Appl Acoust* 2022; 196: 108866.
15. Mitra M and Gopalakrishnan S. Guided wave based structural health monitoring: a review. *Smart Mater Struct* 2016; 25: 053001.
16. Gao G, Chen H, Hu N, et al. Experimental observation of static component generation by lamb wave propagation in an elastic plate. *Ultrasonics* 2021; 117: 106537.
17. Liao WL, Sun H, Wang YS, et al. A novel damage index integrating piezoelectric impedance and ultrasonic guided wave for damage monitoring of bolted joints. *Struct Health Monit* 2023; 22: 3514–3533.
18. Gao X and Wang W. Bolt load looseness measurement for slip-critical blind bolt by ultrasonic technique: a feasibility study with calibration and experimental verification. *Struct Health Monit* 2024; 23: 527–538.
19. Tong T, Hua J, Gao F, et al. Identification of bolt state in lap joint based on propagation model and imaging methods of lamb waves. *Mech Syst Signal Process* 2023; 200: 110569.
20. Zhang Z, Liu M, Su Z, et al. Quantitative evaluation of residual torque of a loose bolt based on wave energy dissipation and vibro-acoustic modulation: a comparative study. *J Sound Vib* 2016; 383: 156–170.
21. Du F, Wu SW, Xing SS, et al. Temperature compensation to guided wave-based monitoring of bolt loosening using an attention-based multi-task network. *Struct Health Monit*. DOI: 10.1177/14759217221113443.
22. Kędra R and Rucka M. Preload monitoring in a bolted joint using lamb wave energy. *B Pol Acad Sci-Tech* 2019; 67: 1161–1169.
23. Kim N and Hong MS. Measurement of axial stress using mode-converted ultrasound. *NDT&E Int* 2009; 42: 164–169.
24. Zhang Z, Liu ML, Liao YZ, et al. Contact acoustic nonlinearity (Can)-based continuous monitoring of bolt loosening: hybrid use of high-order harmonics and spectral sidebands. *Mech Syst Signal Process* 2018; 103: 280–294.
25. Zhang M, Shen Y, Xiao L, et al. Application of subharmonic resonance for the detection of bolted joint looseness. *Nonlinear Dynam* 2017; 88: 1643–1653.
26. Jiang X, Ma K, Lu S, et al. Structure bolt tightening force and loosening monitoring by conductive Mxene/Fpc pressure sensor with high sensitivity and wide sensing range. *Sens Actuators A: Phys* 2021; 331: 113005.
27. Lu X, Soh CK and Avvari PV. Lamb wave propagation in vibrating structures for effective health monitoring. In: *Proceedings of SPIE - The international society for optical engineering*, San Diego, California, 2015; vol. 9438, p. 94381Z.
28. Wang P, Zhou W, Bao Y, et al. Ice monitoring of a full-scale wind turbine blade using ultrasonic guided waves under varying temperature conditions. *Struct Control Health Monit* 2018; 25: e2138.
29. Salmanpour MS, Khodaei ZS and Ferri Aliabadi MH. Impact damage localisation with piezoelectric sensors under operational and environmental conditions. *Sensors (Switzerland)* 2017; 17: 1178.
30. Lee J and Cho Y. Using lamb waves to monitor moisture absorption in thermally fatigued composite laminates. *J Korean Soc Nondestr Test* 2016; 36: 175–180.
31. Ewald V, Groves RM and Benedictus R. DeepSHM: a deep learning approach for structural health monitoring based on guided lamb wave techniques. In: *Conference on sensors and smart structures technologies for civil, mechanical, and aerospace systems*, Denver, CO, 27 March 2019, p. 10970.

32. Weber L, Lapuschkin S, Binder A, et al. Beyond explaining: opportunities and challenges of XAI-based model improvement. *Inf Fusion* 2023; 92: 154–176.
33. Ewald V, Venkat RS, Asokkumar A, et al. Perception modelling by invariant representation of deep learning for automated structural diagnostic in aircraft maintenance: a study case using DeepSHM. *Mech Syst Signal Process* 2022; 165: 108153.
34. Meister S, Wermes M, Stuve J, et al. Investigations on explainable artificial intelligence methods for the deep learning classification of fibre layup defect in the automated composite manufacturing. *Compos Part B-Eng* 2021; 224: 109160.
35. Pandey P, Rai A and Mitra M. Explainable 1-D convolutional neural network for damage detection using lamb wave. *Mech Syst Signal Process* 2022; 164: 108220.
36. Park JH, Jo HS, Lee SH, et al. A reliable intelligent diagnostic assistant for nuclear power plants using explainable artificial intelligence of GRU-AE, LightGBM and SHAP. *Nuclear Eng Technol* 2022; 54: 1271–1287.
37. Brito LC, Susto GA, Brito JN, et al. An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery. *Mech Syst Signal Process* 2022; 163: 108105.
38. Meister S, Wermes M, Stuve J, et al. Cross-evaluation of a parallel operating SVM-CNN classifier for reliable internal decision-making processes in composite inspection. *J Manuf Syst* 2021; 60: 620–639.
39. Geetha GK and Sim SH. Fast identification of concrete cracks using 1D deep learning and explainable artificial intelligence-based analysis. *Automat Constr* 2022; 143: 104572.
40. Hu MP, Yue N, Salmani Pour Avval S, et al. *Explainable artificial intelligence study on bolt loosening detection using lamb waves* (in Review).
41. Chen HY and Lee CH. Vibration signals analysis by explainable artificial intelligence (XAI) approach: application on bearing faults diagnosis. *IEEE Access* 2020; 8: 134246–134256.
42. Letzgus S, Wagner P, Lederer J, et al. Toward explainable AI for regression models. *arXiv preprint: 2112.11407v2*, 2023.
43. Lomazzi L, Fabiano S, Parziale M, et al. On the explainability of convolutional neural networks processing ultrasonic guided waves for damage diagnosis. *Mech Syst Signal Process* 2023; 183: 109642.
44. Zacharias J, von Zahn M, Chen J, et al. Designing a feature selection method based on explainable artificial intelligence. *Electronic Markets* 2022; 32: 2159–2184.
45. Polle C, Bosse S and Axel SH. Damage location determination with data augmentation of guided ultrasonic wave features and explainable neural network approach for integrated sensor systems. *Comput Chem Eng* 2024; 13: 32.
46. Kiranyaz S, Avci O, Abdeljaber O, et al. 1D Convolutional neural networks and applications: a survey. *Mech Syst Signal Process* 2021; 151: 107398.
47. Patterson J and Gibson A. *Deep learning: a practitioner's approach*. Sebastopol, California: O'Reilly, 2017.
48. Almutairi M, Nikitas N, Abdeljaber O, et al. A methodological approach towards evaluating structural damage severity using 1D CNNS. *Structures* 2021; 34: 4435–4446.
49. Kingma DP and Ba J. Adam: a method for stochastic optimization. In: *International conference on learning representations (ICLR)*, San Diego, 2015, pp. 1–15. DOI: 10.48550/arXiv.1412.6980.
50. Simonyan K, Vedaldi A and Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv preprint: 1312.6034*, 2013.
51. Montavon G, Lapuschkin S, Binder A, et al. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recogn* 2017; 65: 211–222.
52. Selvaraju RR, Cogswell M, Das A, et al. Grad-Cam: visual explanations from deep networks via gradient-based localization. *Int J Comput Vision* 2019; 128: 336–359.
53. Springenberg JT, Dosovitskiy A, Brox T, et al. Striving for simplicity: the all convolutional net. *arXiv preprint*: 14126806, 2014.
54. The Aluminum Association. *Aluminum standards and data*. Washington, DC: Aluminum Association, 2000.
55. Humer C, Holl S, Kralovec C, et al. Damage identification using wave damage interaction coefficients predicted by deep neural networks. *Ultrasonics* 2022; 124: 106743.
56. Yue N and Aliabadi MH. A scalable data-driven approach to temperature baseline reconstruction for guided wave structural health monitoring of anisotropic carbon-fibre-reinforced polymer structures. *Struct Health Monit* 2020; 19: 1487–1506.
57. Lee B, Manson G and Staszewski W. Environmental effects on lamb wave responses from piezoceramic sensors. *Mater Sci Forum, Trans Tech Publ* 2003; 440: 195–202.
58. Hinton GE and Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* 2006; 313: 504–507.
59. Jolliffe IT. *Principal component analysis for special types of data*. New York, NY: Springer, 2002.
60. Yu Y, Wang C, Gu X, et al. A novel deep learning-based method for damage identification of smart building structures. *Struct Health Monit* 2018; 18: 143–163.
61. Lin YZ, Nie ZH and Ma HW. Structural damage detection with automatic feature-extraction through deep learning. *Comput-Aided Civil Infrastruct Eng* 2017; 32: 1025–1046.

Appendix

Acronyms

Acronyms	Explanation
AI	Artificial intelligence
SHM	Structural health monitoring
DeepSHM	Deep learning-based SHM
XAI	Explainable artificial intelligence
1D CNN	One-dimensional convolutional neural network
SA	Sensitivity analysis
SST	Smooth simple Taylor
DT	Deep Taylor
GCAM	Gradient-weighted class activation mapping
DCAM	Deep gradient-weighted class activation mapping
GGCAM	Guided gradient-weighted class activation mapping
FS	Feature sensitivity which utilizes the changes in testing accuracy to assess the accuracy of XAI algorithms in the calculation of importance scores
X1D-CNN	XAI-based regression models that use the most important signal determined by each XAI algorithm as input
FS-X1D-CNN	Utilize FS to select the optimal XAI algorithm, and then use the optimal XAI algorithm to pick the most important signals as input
RAN-1D-CNN	Random-1D CNN regression model randomly selects a signal from the raw input vector as input
RI-1D-CNN	Raw input-regression model uses the raw input vector as input
DAE-1D-CNN	Deep autoencoders-regression model uses 1500 features obtained from the self-encoding as input