

Acoustic emission waveform filtering method based on autoencoder and BP neural network

Cite as: Rev. Sci. Instrum. 96, 094901 (2025); doi: 10.1063/5.0285174

Submitted: 12 June 2025 • Accepted: 6 August 2025 •

Published Online: 2 September 2025



View Online



Export Citation



CrossMark

Shunli Jiang,^{1,a)} Kangpei Zheng,² and Jinghui Jv³

AFFILIATIONS

¹ School of Civil Engineering, Shijiazhuang Tiedao University, Shijiazhuang, China

² School of Civil Engineering, Chongqing Jiaotong University, Chongqing, China

³ School of Future Information Techology, Shijiazhuang University, Shijiazhuang, China

^{a)} Author to whom correspondence should be addressed: 1535957803@qq.com

ABSTRACT

To address the issue of acoustic emission (AE) signal superposition, a waveform filtering method based on an autoencoder and backpropagation (BP) neural network is proposed to effectively separate AE signals from noise. The performance of the model in classifying AE and noise signals was evaluated through pencil lead break experiments and numerical simulations. The results show that the autoencoder–BP neural network model achieved excellent classification performance, with a recognition rate of 96% for AE signals and 98% for noise signals. After filtering using the proposed model, the processed data significantly improved the localization accuracy of AE sources. This study provides an effective AE signal processing method for structural health monitoring systems and holds important significance for enhancing the accuracy of safety monitoring in concrete structures.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0285174>

I. INTRODUCTION

In structural health monitoring, acoustic emission (AE) technology has become a crucial means for detecting defects in materials and assessing structural integrity due to its high sensitivity and non-destructive characteristics.^{1–5} However, in practical applications, AE signals are often obscured by various types of noise, including environmental noise, mechanical noise, and electromagnetic interference.^{6–10} Effectively separating the true AE signals from the noise is key to ensuring detection accuracy and reliability.

Many researchers have conducted extensive work on AE signal classification. Abdelrahman *et al.*¹¹ used wavelet transform to analyze the time-domain and frequency-domain features of signals to separate AE signals from noise. Ren *et al.*¹² employed decision tree methods to classify AE signals based on parameters such as rise time, counts, amplitude, peak frequency, and absolute energy. Ohn and Ohtsu¹³ classified crack AE signals using parameter analysis and SiGMA analysis methods. Aggelis¹⁴ classified concrete crack signals based on AF and RA parameters. Huilan *et al.*¹⁵ studied the direction of crack AE signals based on moment tensor inversion theory, achieving AE signal classification. Liu *et al.*¹⁶ used a four-parameter k-means algorithm for cluster analysis to distinguish

AE signals generated by rock impact loading and rock fracturing. Dong *et al.*¹⁷ studied the classification of rock micro-crack AE signals using an improved RA-AF method. Das *et al.*¹⁸ classified AE signals of different crack models based on the SVM algorithm in machine learning. Thirumalaiselvi and Sasimal¹⁹ classified AE signal datasets recorded at different damage stages using supervised and unsupervised pattern recognition algorithms. Behnia *et al.*²⁰ used unsupervised kernel fuzzy c-means pattern recognition analysis and principal component analysis to classify AE signals of ordinary reinforced concrete specimens and steel fiber reinforced concrete specimens at various damage stages. Ju *et al.*²¹ studied the application of multiple machine learning algorithms to cluster AE signals of known crack modes, with clustering results consistent with actual crack modes. Zhang and Deng²² proposed a method based on statistical analysis of the main frequency characteristics of AE waveforms to determine the optimal ratio of RA value to average frequency, thereby identifying crack modes. Jiang *et al.*²³ studied AE signals using sample entropy and variational mode decomposition, proposing VMD-based SampEn as a new AE signal indicator to distinguish AE signals of different damage modes. Siracusano *et al.*²⁴ constructed a Hilbert–Huang transform framework to classify AE fracture signals. Dai *et al.*²⁵ compared different machine learning

AE crack classification algorithms, concluding that the RF algorithm had better accuracy and efficiency. Li *et al.*²⁶ introduced the synchrosqueezed wavelet transform and discovered that AE waves induced by different mechanisms exhibited different energy distribution patterns, establishing a multi-branch convolutional neural network model with two branches to automatically classify different types of AE waves.

As mentioned above, many researchers have successfully applied various methods to classify acoustic emission (AE) signals. However, the AE waveform filtering method based on an autoencoder and BP neural network offers unique research value compared to other classification algorithms. First, the advantage of the autoencoder in unsupervised learning allows it to automatically extract deep features from AE signals without relying on manually defined parameters, which facilitates the discovery of potential patterns and signal characteristics. Second, the BP neural network demonstrates strong classification capability, particularly in handling nonlinear and complex data, thereby further improving classification accuracy. Unlike studies that rely on a single algorithm, this study combines the strengths of both unsupervised and supervised learning, providing a more comprehensive and robust signal processing approach. Based on this framework, an autoencoder and convolutional neural network (CNN) model was constructed to separate AE signals from noise signals, and the feasibility of the proposed algorithm was validated through pencil lead break (PLB) experiments. The results provide an effective method for AE waveform filtering.

II. AUTOENCODER AND BP NEURAL NETWORK MODEL

The algorithmic process of the autoencoder and BP neural network model is as follows:

- (1) Preprocessing of acoustic emission waveforms. The raw waveform data undergo preprocessing operations, such as de-meaning, taking the absolute value, and normalization, in order to meet the data input requirements of neural network models.
- (2) Construction and training of the autoencoder network. An autoencoder network is built, and the standardized data samples are input into the autoencoder model. Through the encoder's denoising and feature extraction, data that capture the main characteristics of the waveform can be obtained.
- (3) Construction and training of the BP neural network. A BP (backpropagation) neural network is constructed, and the feature data extracted by the autoencoder are fed into the BP neural network. The model outputs a two-dimensional vector for each sample: the first component represents the probability that the sample is noise and the second represents the probability that the sample is a signal. The sum of the two probabilities is 1.0. The classification result of each sample is determined by the component with the higher probability. By applying this classification process to all samples, the signal-to-noise classification results for each sample in the original data can be obtained.

A. Data preprocessing

1. De-meaning

Acoustic emission signals are typically non-stationary and time-varying and may contain certain DC offsets or background noise. In response to these characteristics, this study proposes the first step in the waveform data preprocessing procedure—de-meaning of the waveform data. De-meaning refers to the removal of the average deviation from the data. The specific calculation process is shown in the following equations:

$$A = \frac{1}{n} \left(\sum_{i=1}^n y_i \right), \quad (1)$$

defining y_i as follows:

$$y_i = y_i - A, \quad (2)$$

where n is the number of sampling points of the waveform data, A is the average value of the waveform data, and x is the amplitude of the acoustic emission signal waveform.

2. Absolute value transformation

To eliminate the influence of positive and negative waveform fluctuations, this study proposes the second step in the waveform data preprocessing procedure—absolute value transformation. For each point in the discrete waveform data, the original value is replaced by the absolute value of its amplitude, as shown in the following equation:

$$y_i = |y_i|. \quad (3)$$

3. Normalization

To eliminate the impact of amplitude differences among different waveform data, this study proposes the third step in the waveform data preprocessing procedure—data normalization—to improve the comparability of waveform data features. At the same time, data normalization is a fundamental step in data mining. Here, linear normalization (min-max scaling) is applied to process the data, and the specific procedure is shown in the following equation:

$$y_i = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}}, \quad (4)$$

where y_{\max} and y_{\min} are the maximum and minimum values of the waveform data, respectively. For each time point value in the waveform data, it is replaced by the ratio of the difference between the original value and the minimum value of the entire waveform to the range of the waveform (the difference between the maximum and minimum values). The adjusted value will fall within the interval (0.0, 1.0).

B. Principles of the autoencoder and BP neural network model

1. Principle of autoencoder

An autoencoder is a form of unsupervised learning neural network. Depending on the learning approach, it can be categorized into different types, such as sparse autoencoder, regularized

autoencoder, and variational autoencoder. In unsupervised learning, the input data are used solely based on their features without involving their label categories. This approach is particularly useful in scenarios where there are insufficient background knowledge and principles about the problem, where the time and effort required for manual labeling are too high, or where the number of features is so large that labeling becomes challenging. An autoencoder typically consists of two parts: an encoder and a decoder. Its purpose is to serve as a feature detector to compress and reduce the dimensionality of the data when the input data have high dimensions and numerous features. It is an efficient data representation method similar to principal component analysis (PCA). The structure of an autoencoder is usually symmetrical around the middle layer. The part from the input layer to the middle layer is called the encoder, which compresses and reduces the noise and dimensionality of the data. The part from the middle layer to the output layer is called the decoder, which decompresses and restores the data. Taking the neural network model structure shown in Fig. 1 as an example, let the input data be \vec{x} , which are passed forward through two fully connected layers with a gradually decreasing number of neurons. The

output of the middle hidden layer is \vec{h} . Then, it is passed forward through two fully connected layers with a gradually increasing number of neurons, and finally, the output is $\hat{\vec{x}}$ from the output layer. This model structure constitutes a classic autoencoder.

The calculation formulas for the intermediate hidden layer and the output reconstructed data of the autoencoder are shown in the following equations:

$$\vec{h} = f(w^{(1)}x + b^{(1)}), \quad (5)$$

$$\hat{\vec{x}} = f(w^{(2)}\vec{h} + b^{(2)}), \quad (6)$$

where $w^{(1)}$, $w^{(2)}$, $b^{(1)}$, and $b^{(2)}$ are network parameters and $f(\cdot)$ is the activation function.

2. BP neural network principles

In the 1980s, a unidirectional, multilayer, feedforward neural network connected by numerous neurons was gradually developed.

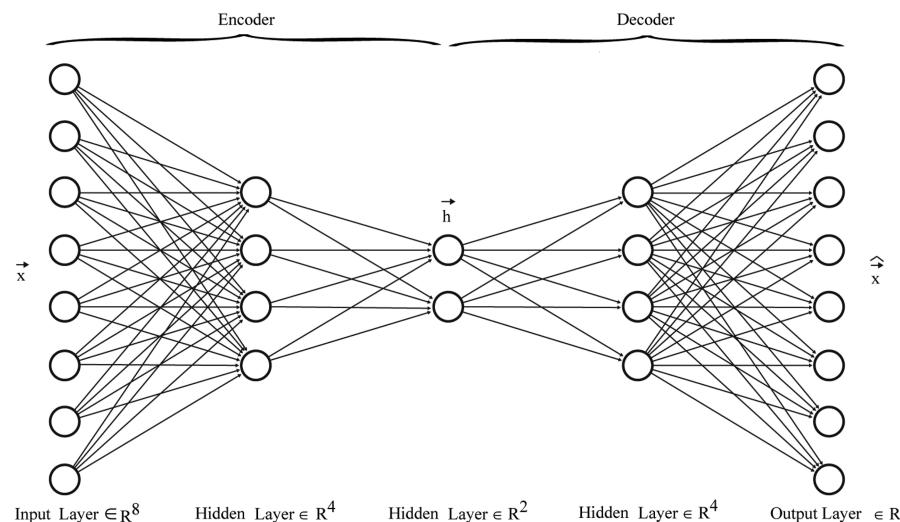


FIG. 1. Self-encoder structure.

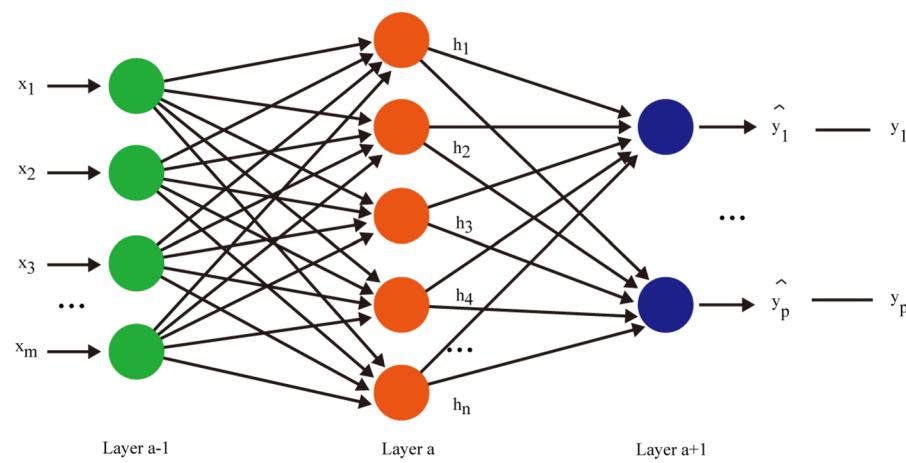


FIG. 2. Feedforward neural network structure.

As shown in Fig. 2, if the neurons in a certain layer are connected to all the neurons in the previous layer, this connection method is called a fully connected layer.

During the forward propagation process, assume that layer a of a neural network has n neurons and layer $a - 1$ has m neurons. When these two layers are fully connected, the output of the neurons in layer a can be represented by

$$h_j^{(a)} = f\left(\sum_{i=1}^M w_{ji}^{(a-1)} x_i^{(a-1)} + b_j^{(a)}\right). \quad (7)$$

Here, $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$, where $w_{ji}^{(a-1)}$ represents the weight between neurons and $b_j^{(a)}$ is the bias of the neuron.

In practice, neural network models typically contain a large number of weight parameters; therefore, iterative optimization methods are commonly employed to minimize the loss function. The gradient descent method from optimization theory is used to update the weights in the network, adjusting the parameters in the direction opposite to the gradient by a certain amount, as shown in the following equations:

$$w_{ji} = \left(w_{ji} - \eta \frac{\partial J}{\partial w_{ji}}\right), \quad (8)$$

$$b_j = \left(b_j - \eta \frac{\partial J}{\partial b_j}\right). \quad (9)$$

Here, η is the learning rate, also known as the step size. It is used to avoid getting stuck in local minima. A learning rate that is too small will slow down the learning process, while a learning rate that

is too large may cause oscillations in the learning process or even prevent convergence. Typically, at the beginning of network training, the learning rate is set relatively high, and it is gradually reduced as the training error decreases. To avoid local optima, a randomized gradient descent method can also be used.

In summary, the basic steps of the BP neural network training process can be summarized in the following three steps: (1) randomly initialize network weights and neuron biases; (2) perform forward propagation to calculate the outputs of the hidden layer neurons and the output layer neurons; and (3) perform backpropagation to adjust each weight parameter based on the loss function, deriving the error from the subsequent layer to the previous layer, and passing it backward layer by layer. In practice, this process is also known as an epoch.

The training process of a BP neural network model is the repeated iteration of the above process. By setting a certain number of training iterations or a certain loss function threshold, training stops when the conditions are met, at which point the gradient approaches zero, and all weight parameters in the model reach a state where the loss function for the training data is minimized. This is generally considered the completion of the model training process.

C. Dataset construction

To construct a training and testing dataset for the acoustic emission (AE) signal-to-noise classifier, numerous loading failure tests on concrete structures are required. Before starting the tests, it is necessary to set up the AE acquisition system, which includes AE collection instruments, preamplifiers, and sensors, and the strain measurement system for actual stress distribution measurement. During the tests, AE signals, strain, load, and displacement data need

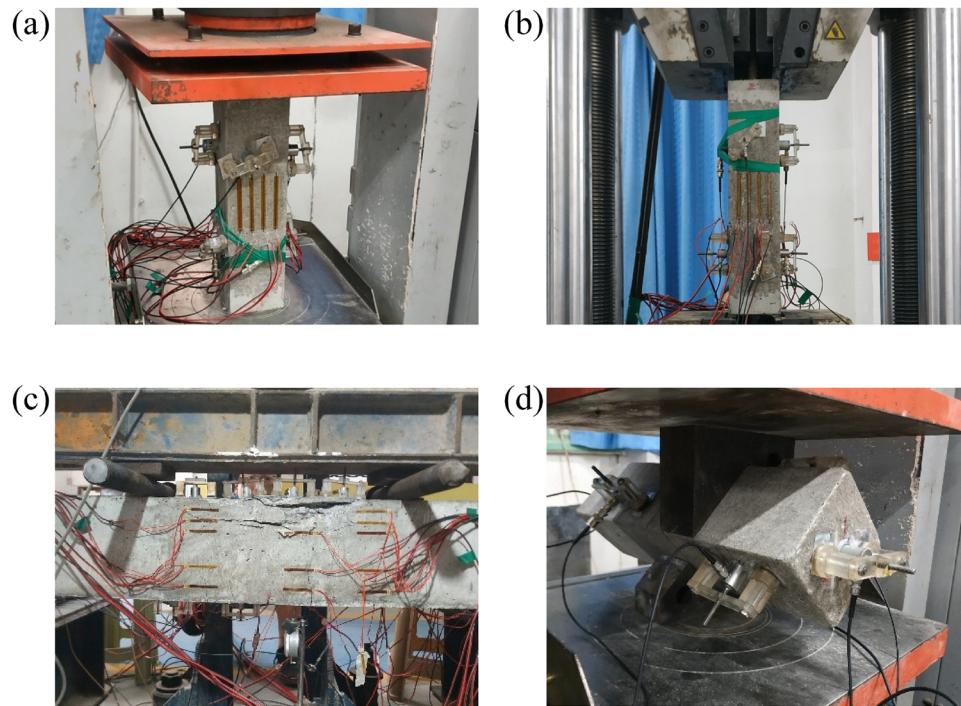


FIG. 3. (a) Axial compression test; (b) axial tensile test; (c) bending test; and (d) shear test.

TABLE I. Statistics of various types of test parameters.

Type of test	Name of the test	Positioning mode	Number of sensors	Number of hits
Axial compression test	compress_1	3D	8	209 237
	compress_2	3D	8	195 645
	compress_new_1	3D	7	417 435
	compress_new_2	3D	7	376 759
	compress_new_3	3D	7	358 622
	compress_new_4	3D	7	441 218
	compress_new_5	3D	7	208 465
Axial tensile test	compress_new_6	3D	7	456 271
	tense_1	3D	8	13 462
Shear test	tense_2	3D	7	17 915
	shear_1	3D	7	21 748
Bending test	shear_2	3D	7	6347
	beam_1	2D	6	229 748
	beam_2	2D	6	98 667
	beam_3	2D	6	35 091
	beam_4	2D	6	186 839
	beam_5	2D	6	110 614
	beam_6	2D	6	129 667
	beam_long_1	2D	5	160 169
	beam_short_1	2D	7	974 062
	beam_short_2	3D	7	973 610
	beam_short_3	2D	8	544 376
	beam_short_4	2D	8	216 616

to be collected. After the tests, the collected data are summarized and organized into a user-friendly experimental database, laying a solid foundation for the subsequent construction of the training and testing datasets.

The equipment used in the test process includes the DNS300 uniaxial rheometer, resistance strain gauges, test3826 wireless strain boxes, an 8-channel AE MISTRAS system, preamplifiers (40 dB), R6D sensors (40–100 kHz), and related AEWIn software and computer workstations.

Since the stress conditions in actual engineering structures are relatively complex, using a single type of stress test mode lacks universality and persuasiveness. Therefore, this study uses multiple types of loading forms, including tension, compression, shear, and bending, to load the tested structures. The combination of various loading tests under different stress conditions forms a comprehensive test database, ensuring that the data sources in the database cover tensile, compressive, and shear stress states, thereby enhancing the generalization of the data.

As shown in Fig. 3, the on-site test diagrams include axial compression tests, axial tensile tests, shear tests, and beam bending tests of concrete. In particular, eight blocks were subjected to axial compression tests, two blocks were subjected to axial tensile tests, two blocks were subjected to shear tests, and 11 blocks were subjected to concrete bending tests, as detailed in Table I.

Based on the acoustic emission data obtained from the above experiments, a portion of representative data samples was selected and divided into three categories: training set, test set, and

validation set, with proportions of 60%, 20%, and 20%, respectively. This division is intended for subsequent model training and performance evaluation. The specific data allocation is shown in Table II.

D. Construction and training of autoencoder structure

1. Construction of autoencoder structure

After the training dataset is established, an appropriate autoencoder neural network model can be built. According to its principles, the model should compress the original data to the number of essential features as much as possible without losing the main features. Therefore, selecting the appropriate number of compressed features is particularly important. This paper uses experimental methods to determine the appropriate number.

To avoid overly complex models when combined with the BP neural network, the autoencoder adopts a single hidden layer. The number of neurons in this hidden layer is considered a multiple compression of the original data features. The original data have 1024 dimensions, and the number of neurons in the hidden layer is constructed using multiple compression methods, such as 512, 256, 128, 64, and 32. As the number of neurons in the hidden layer decreases, the proportion of lost main features increases, and the fitting effect gradually decreases. When the number of features is too small, it will not fit the original data well.

In summary, experimental methods can be used to determine the structure of the autoencoder model. First, construct the basic

TABLE II. Sources of the convolutional neural network dataset.

Dataset types	Experiment name	Sample quantity	Total quantity
Training dataset	beam_1	77	582
	beam_4	80	
	beam_6	77	
	beam_long_1	63	
	beam_short_2	75	
	beam_short_4	55	
Validation set	compress_1	89	194
	compress_new_1	66	
	shear_1	50	
	shear_2	43	
	tense_1	51	
	tense_2	50	
Testing dataset	beam_2	12	194
	beam_3	22	
	beam_5	17	
	beam_short_1	18	
	beam_short_3	12	
	compress_2	23	
	compress_new_2	15	
	compress_new_3	18	
	compress_new_4	18	
	compress_new_5	21	
	compress_new_6	18	

autoencoder model framework, with the number of neurons in the input layer and output layer set to 1024. The number of neurons in the hidden layer starts from 512 and then decreases in multiples. After each adjustment, the model is initially trained, stopping at three epochs. As observed in Table III, as the number of neurons in the hidden layer decreases, the loss value obtained by the loss function from the model output increases.

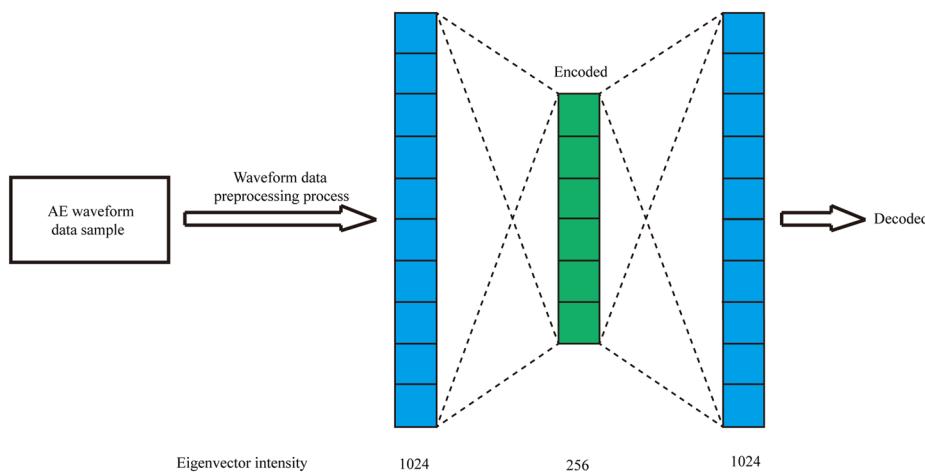
TABLE III. Number of neurons in the hidden layer and loss function values.

Number of intermediate layer neurons	Epoch	Loss function value
512	3	0.0047
256	3	0.0058
128	3	0.0107
64	3	0.0164
32	3	Unable to converge

As the number of neurons in the hidden layer decreases, when the number is still greater than 256, the loss function value increases slightly, indicating that irrelevant features and noise are largely removed with only a minimal loss of main features. When the number of neurons in the hidden layer decreases to 128, the loss function value increases sharply, indicating that a significant proportion of the main features are lost, causing the autoencoder to fail to adequately reconstruct the data. When the number of neurons in the hidden layer decreases to 32, the feature parameters are too few, resulting in the loss of almost all main information, making it impossible for the autoencoder to reconstruct the data after compression. Therefore, the number of neurons in the hidden layer is determined to be 256, establishing the structure of the autoencoder model. The structure of the autoencoder model is shown in Fig. 4. This model effectively filters out irrelevant features and retains the main features of the original data.

2. Autoencoder training

Once the structure of the autoencoder model is confirmed, the training process can begin. First, the dataset used for training needs to be loaded in tensor form, and the batch size for data training needs to be set. The batch size, which is the number of samples taken for each training session, affects the optimization and speed of the model. Given that the total number of samples is 23 000, the batch size is set to 100. This means that each epoch will randomly divide all samples into 230 batches and perform 230 training sessions.

**FIG. 4.** Self-encoder model structure.

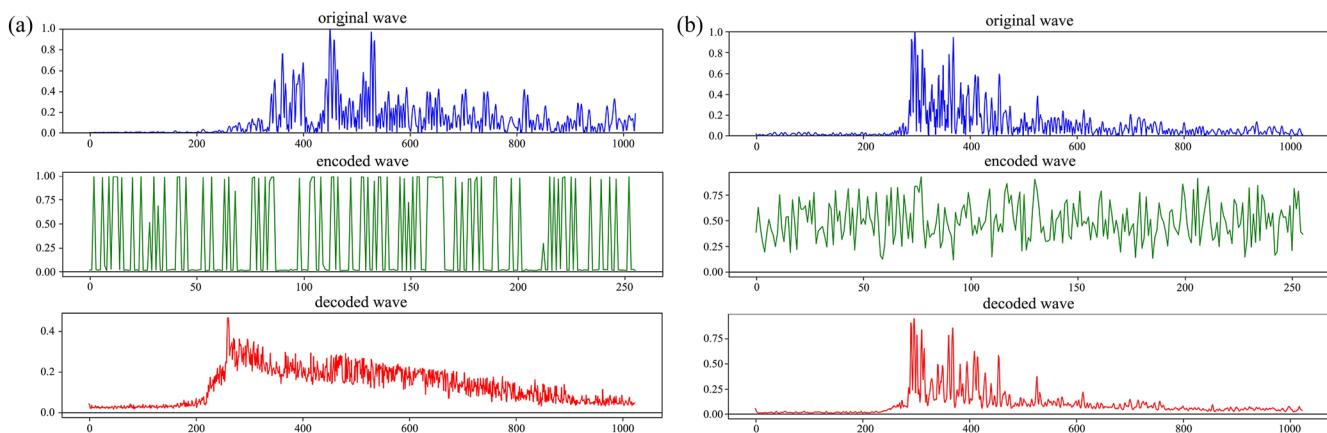


FIG. 5. Comparison of input and output data during training period. (a) Comparison of input and output data at the initial stage of training; (b) comparison of input and output data after 10 epochs of training.

Next, the optimizer choice in this context is the **Adam optimizer**. The Adam algorithm is a method for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The **learning rate is set to 0.005**. The loss function selected is the mean squared error loss (MSELoss). Finally, the stopping condition is manually set. With these preparations complete, training can commence.

At the initial stages of the training process, the model's fit to the data is relatively low, and the output data significantly differ from the original data in terms of overall shape characteristics. As shown in Fig. 5(a), the blue waveform in the top chart represents the 1024-dimensional original data, the green waveform in the middle chart represents the 256-dimensional compressed features, and the red waveform in the bottom chart represents the decoded output waveform data. At this stage, there is a noticeable difference between the top and bottom charts. As the number of training iterations increases, the model's fit improves gradually. The autoencoder transforms the input raw waveforms into more efficient internal representations, aiming to reconstruct the output data to match the input data. This process ensures that the efficient internal representations can be effectively utilized. When the training process reaches 10 epochs, the autoencoder model has essentially learned the main features of the original data. At this stage, the input data and output data of the model are highly similar. Through the compressed features within the model, the original data can be well reconstructed, as shown in Fig. 5(b).

By recording the loss function value after each training session during the training process, a loss curve of the model training process can be plotted, as shown in Fig. 6. The horizontal axis represents the number of training iterations, and the vertical axis represents the corresponding loss function value. As seen in Fig. 6, the loss curve is an oscillating line. This oscillation is due to the randomness of the optimizer, resulting in a non-smooth decrease in the loss function value. It can be observed that in the later stages of training, the amplitude of the loss curve oscillations is relatively smaller compared to the initial stages of training.

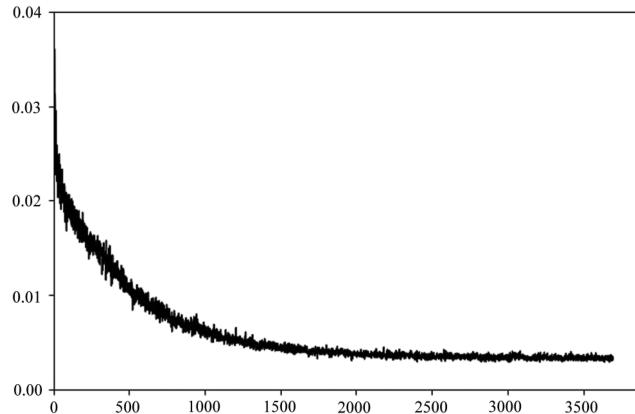


FIG. 6. Variation of loss function values during model training (i.e., loss curves).

If no stopping condition is added and the model is allowed to continue learning, the loss curve shows that the curve converges, with a relatively low loss value when the number of training iterations reaches 2000. Beyond this point, further learning does not significantly reduce the loss. Therefore, the stopping condition for training the autoencoder model in this study is set to 10 epochs, i.e., after 2300 training sessions, the training process stops, and the model is saved to a .pk1 file. This ensures that the autoencoder model has a high degree of data reconstruction accuracy while avoiding overfitting the training dataset.

E. BP neural network structure construction and training

1. BP neural network structure construction

The model structure of the BP neural network adopts the classic fully connected layers. The input layer contains 256 neurons, which receive the input data processed by the standard data

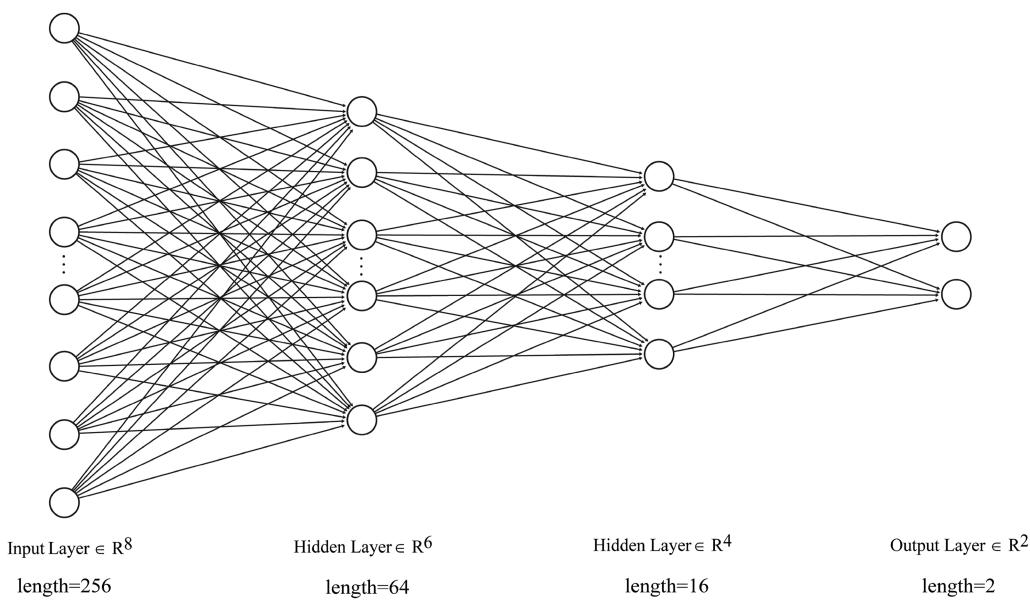


FIG. 7. BP neural network model structure.

preprocessing pipeline and the autoencoder. The middle part includes two hidden layers for feature extraction, with the number of neurons sequentially reduced by a factor of 1/4, having 64 and 16 neurons, respectively. The output layer contains two neurons for classification judgment output. The specific structure of the model is shown in Fig. 7.

Based on this model structure, the weight matrix between the input layer and the first hidden layer has a size of (64, 256), and there are 64 bias values for the neurons in the first hidden layer. The weight matrix between the first hidden layer and the second hidden layer has a size of (16, 64), with 16 bias values for the neurons in the second hidden layer. The weight matrix between the second hidden layer and the output layer has a size of (2, 16), with two bias values for the output layer neurons. The total number of weight parameters in this model can be calculated using the following equation:

$$(256 \times 64 + 64) + (64 \times 16 + 16) + (16 \times 2 + 2) = 17522. \quad (10)$$

This setup demonstrates that the preprocessing done by the autoencoder can significantly enhance the training efficiency of the BP neural network.

2. BP neural network training

The specific steps for training the BP neural network are as follows:

(1) Data preparation

The training data samples were categorized into two classes based on their characteristic features: one class consisted of background noise signals, labeled as class "0"; the other class comprised acoustic emission signals exhibiting features associated with physical events, labeled as class "1." The labeled samples are then placed back into the same directory to create a mixed dataset with classification labels.

The training dataset is randomly shuffled, while the test dataset is simply merged.

(2) Batch size and sampling

Set the batch size during training to 100, and use random sampling. In each epoch, the small batches used for training consist of 12 batches, each with 100 samples, obtained through random sampling. A DataLoader class object is constructed to facilitate this process, allowing small batch data to be retrieved during each training session.

(3) Data preprocessing

Preprocess the data in conjunction with the previously mentioned research. All waveform samples are subjected to a standard acoustic emission data preprocessing workflow, resulting in 1024-dimensional features that are mean-centered, absolute-valued, and normalized. These data are then input into the trained autoencoder for denoising and feature extraction, yielding 256-dimensional output feature vectors. These processed data serve as the input data for the BP neural network.

(4) Hyperparameter adjustment

Use the Adam optimizer with a learning rate set to 0.0005. Cross-entropy loss (CrossEntropyLoss) is chosen as the loss function. To determine when to stop training, no specific stopping condition is initially set, allowing training to continue indefinitely. After each training session, the current model state is used to make predictions on both the training and test datasets. The prediction scores are calculated, and the score variation with the number of training iterations is plotted as a function curve, as shown in Fig. 8.

Calculating the score value is the most basic method to measure the effectiveness of a classification problem. Its principle is to calculate the proportion of correctly classified samples to the total

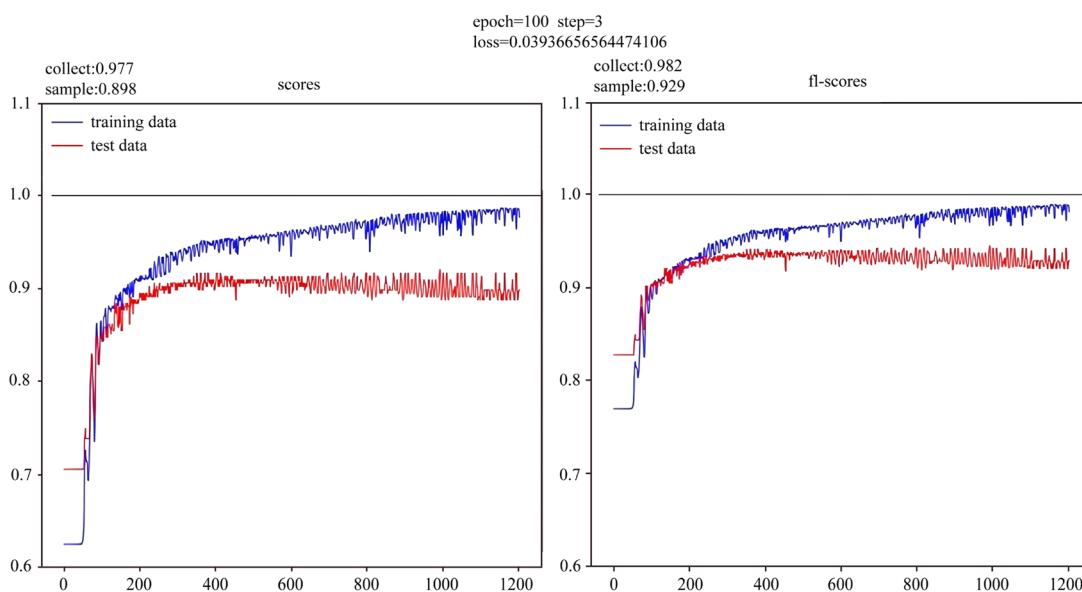


FIG. 8. Change pattern of training and test set scores.

number of samples. Let n be the number of correctly predicted samples and n_{total} be the total number of samples. The score value is given by (Fig. 9)

$$score = \frac{n}{n_{total}}. \quad (11)$$

However, the score value has certain limitations in evaluating classification results. When there is a significant disparity in the number of different types of samples in the dataset, the score value cannot accurately reflect the model's fit to the objective patterns. Therefore, in practical work, the $f1$ -score, which is a more reasonable calculation method, is often used as the final evaluation metric for classification problems.

The calculation of the $f1$ -score involves several basic concepts: TP (true positive): correctly predicted positive samples; FP (false positive): incorrectly predicted negative samples as positive; FN (false negative): incorrectly predicted positive samples as negative; and TN (true negative): correctly predicted negative samples.

Precision and recall are two metrics calculated based on the prediction status. Precision is the proportion of true positive samples among the samples predicted as positive, while recall is the proportion of positive samples that are correctly predicted as positive. They can be calculated using the following equations:

$$precision = \frac{TP}{TP + FP}, \quad (12)$$

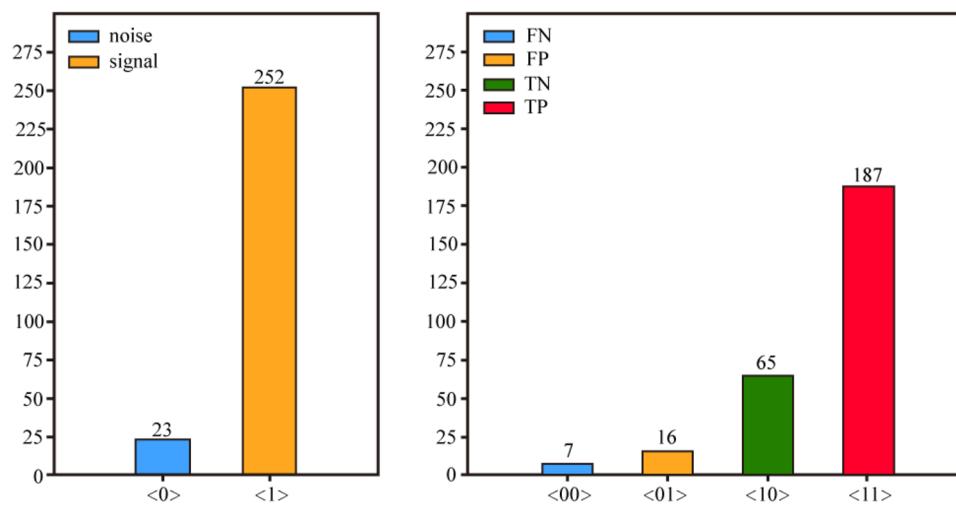


FIG. 9. Change pattern of training and test set scores.

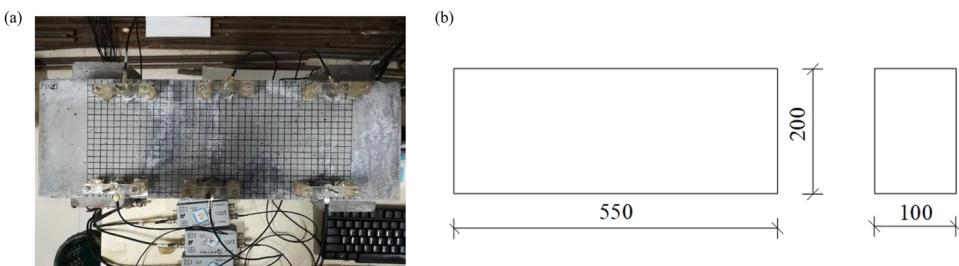


FIG. 10. Broken lead sample. (a) Lead break test site plan; (b) schematic diagram of the dimensions of the trabecular specimen.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (13)$$

The f_1 -score is the harmonic mean of precision and recall, described by Eq. (14). By transforming the equation, the calculation formula for f_1 -score is obtained as shown in Eq. (15). The f_1 -score considers precision and recall equally important, making it a more scientific and reasonable metric for evaluating the effectiveness of neural network model classification problems,

$$\frac{1}{f_1} = \frac{1}{2} \left(\frac{1}{\text{precision}} + \frac{1}{\text{recall}} \right), \quad (14)$$

$$f_1 = 2 \times \left(\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \right). \quad (15)$$

The x-axis in the score variation plot represents the number of training iterations, while the y-axis represents the prediction score on specific data. By observing the function curves in the plot, one can discern certain patterns in how the scores change with increased training iterations for different datasets.

In the initial stages of training, as the number of iterations increases, the model's performance on both datasets is lacking. However, due to continuous optimization during the training process, both scores show an increasing trend, with the training dataset score increasing more rapidly, reflecting the BP neural network model's self-fitting process. As the number of iterations continues to increase, the growth trend of both scores gradually levels off, approaching 1.0. When the number of iterations reaches around 500,

the model's score on the training dataset continues to grow, while the score on the test dataset reaches a stable peak. This indicates that the model has achieved its optimal state.

As training continues, the score on the training dataset gets closer to 1.0, while the score on the test dataset starts to decline, indicating overfitting. The model has excessively learned some characteristics of the training dataset that are not present in the test dataset, reducing its generalization performance and classification accuracy for data from other sources.

In summary, 42 epochs ($42 \times 12 = 504$ training sessions) are chosen as the stopping condition for the BP neural network training. When this condition is met, training stops to avoid overfitting and ensure good generalization performance. After training, the BP neural network model is used to predict the test dataset again. The results are shown in Fig. 10, with a calculated score of 0.916 and an f_1 -score of 0.942. The loss function value for the final training session is 0.107, indicating that the model has achieved a high level of classification accuracy and performance. The model is then saved to a .pkl file, completing the BP neural network training.

III. EXPERIMENTAL VERIFICATION

Using events containing noise hits for acoustic emission (AE) source localization lacks certain rationality and can lead to significant errors in the localization results. To verify the effectiveness of the AE signal-to-noise separation algorithm, a small beam lead break test was designed. The time difference localization algorithm was used for AE source localization to validate the effectiveness of the signal-to-noise separation algorithm.

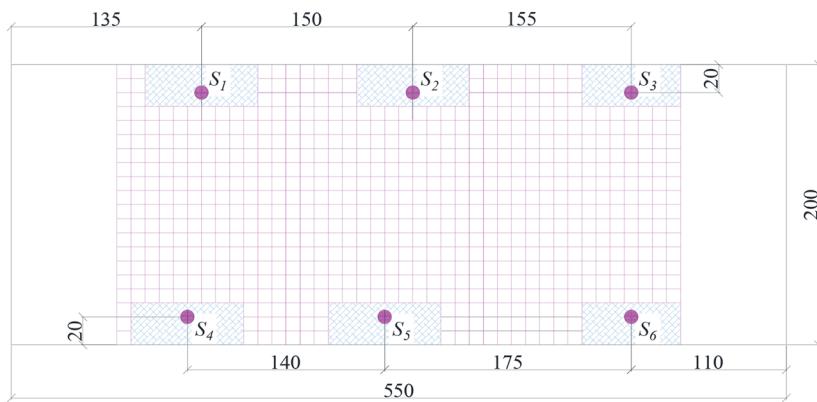


FIG. 11. Layout of gridlines and sensors.

A. Design and fabrication of small beam specimens

The specimens for the lead break test are small beams made of ultra-high performance concrete (UHPC) with a steel fiber content of 1%. The test setup is shown in Fig. 10(a). The dimensions of the specimens are designed to be 550 (length) \times 100 (width) \times 200 mm³ (depth), as shown in Fig. 10(b).

The casting process of the small beam specimens involves using wooden molds. After pouring the concrete into the molds, the specimens are compacted thoroughly using a vibration table to ensure proper densification. The specimens undergo steam curing for three days. After curing, the molds are removed. An angle grinder is used to smooth and level the surfaces of the specimens, removing any uneven edges. The surfaces are then cleaned to eliminate any residual dust and debris. Metal plates are attached to specific locations on the specimen surfaces using AB adhesive. This preparation step facilitates the subsequent placement of fixtures and installation of AE sensors. Finally, the specimens are left to dry until the surface is completely dry, completing the fabrication process of the small beam specimens.

B. Sensor arrangement and grid division for small beam

The arrangement of the acoustic emission (AE) sensors is shown in Fig. 11. A total of six sensors, S₁ – S₆, are set up. To avoid issues such as the symmetry of the AE source position concerning

the sensors causing the system of localization equations to be unsolvable or singular, the sensors are arranged in a staggered manner on the surface of the specimen, maintaining a certain distance between each other. In addition, to prevent excessive interference during the experiment from points where the lead break is too close to the sensors, a safety zone is established around each sensor within which no lead breaks are performed.

To compare the AE localization coordinates from the lead break test with the actual coordinates, a grid is designed on the surface of the specimen. The grid size is 10 \times 10 mm², with the coordinate origin at the lower-left corner of the specimen. Thus, the starting coordinates of the grid lines are (0.075, 0.0), representing the lower-left corner of the grid, and the ending coordinates are (0.475, 0.2).

The signal acquisition for the lead break test was performed using the PC-2 acoustic emission acquisition system from PAC (Physical Acoustics Corporation), USA. The sensor model used was R6a, as shown in Fig. 12.

C. Experimental steps

The steps for the lead break test are as follows:

- (1) Sensors were arranged as shown in Fig. 11, and the concrete specimen was tapped to check the sensitivity of sensor reception.

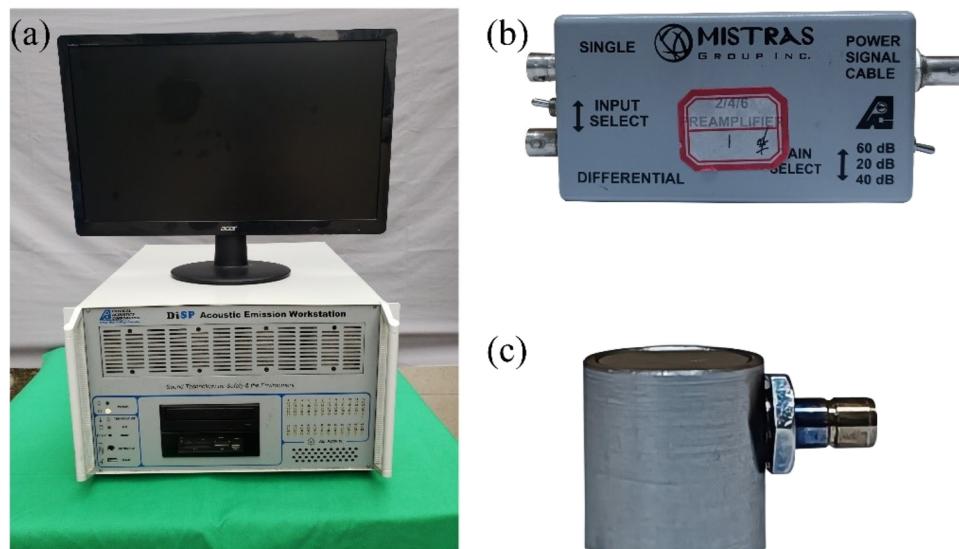


FIG. 12. Diagram of the experimental equipment: (a) acoustic emission equipment, (b) preamplifier, and (c) acoustic emission transducer.

TABLE IV. Classification results of the autoencoder and BP neural network model.

Signal category	Class label	Recognition result (M)	Recognition result (N)	Total number of signals	Misclassification rate (%)	Recognition accuracy (%)
Acoustic emission event signal	M	48	2	50	4	96
Noise signal	N	1	49	50	2	98

- (2) The amplification of the acoustic emission (AE) signal was set to 40 dB, and the noise threshold was also set to 40 dB. The values for peak definition time (PDT), hit definition time (HDT), and hit lockout time (HLT) were set to 150, 300, and 400 μ s, respectively.
- (3) The breakdown test was conducted at each grid point along the x-axis from left to right and along each vertical line from top to bottom. To minimize the influence of human factors, each grid point was tested five times. After completing the breakdown operation at a given grid point, a 60-s interval was observed before proceeding to the next point, until all grid points were tested.
- (4) After the test, the collected acoustic emission data were saved, and the actual coordinates of each lead breakpoint were recorded for subsequent localization analysis and error calculation.

D. Analysis of acoustic emission signal recognition effect

A total of 100 representative signal samples were constructed by randomly selecting 50 typical noise signals from pencil lead break (PLB) tests and 50 genuine acoustic emission (AE) event signals, serving as the dataset for model performance validation. The selection of 100 representative samples, comprising both noise signals and authentic acoustic emission (AE) events, was predominantly conducted through manual inspection. Each signal was individually evaluated and annotated by researchers according to

a set of waveform features, such as amplitude, signal duration, rise time, and spectral energy distribution. The previously trained classification model was employed, leveraging the deep feature extraction capability of the autoencoder and the strength of the BP neural network in nonlinear pattern recognition, to classify and identify the above samples. The classification results are summarized in Table IV. As shown in Table IV, the model demonstrated excellent performance in identifying genuine AE events, achieving an accuracy of 96%. Meanwhile, the identification accuracy for noise signals reached 98%. These results indicate that the constructed model exhibits strong stability and robustness in distinguishing genuine AE events from background noise. It not only achieves high overall classification accuracy but also demonstrates effectiveness and feasibility in intelligent discrimination of microseismic signals under complex conditions, providing technical support and methodological reference for subsequent AE signal processing and intelligent recognition.

E. Acoustic emission source localization accuracy and error analysis

After waveform filtering of the acoustic emission signals, the lead break test data include two different states: the raw, unprocessed data and the data filtered by the convolutional neural network model. This study will conduct acoustic emission hit selection, grouping, and event localization for both types of data. Localization accuracy and errors will be analyzed to validate the effectiveness of the neural network model in filtering acoustic emission signals.

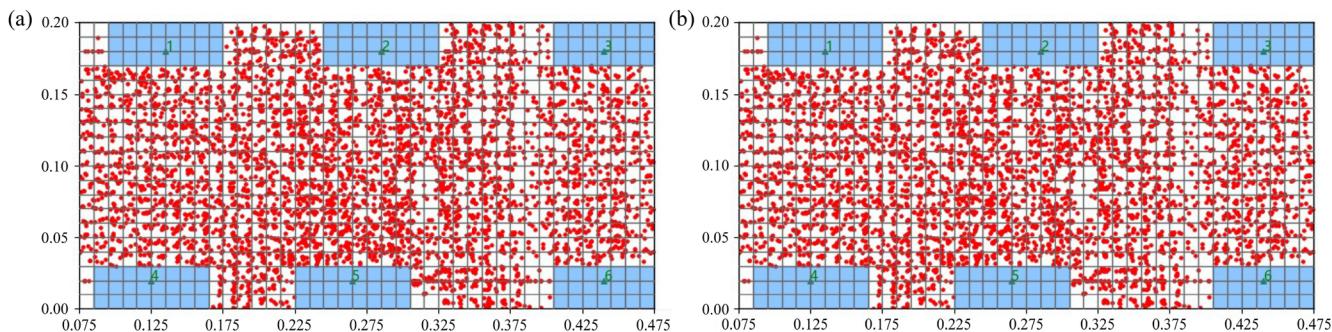


FIG. 13. Positioning scatter plot. (a) Raw data positioning results; (b) signal-to-noise separation data localization results.

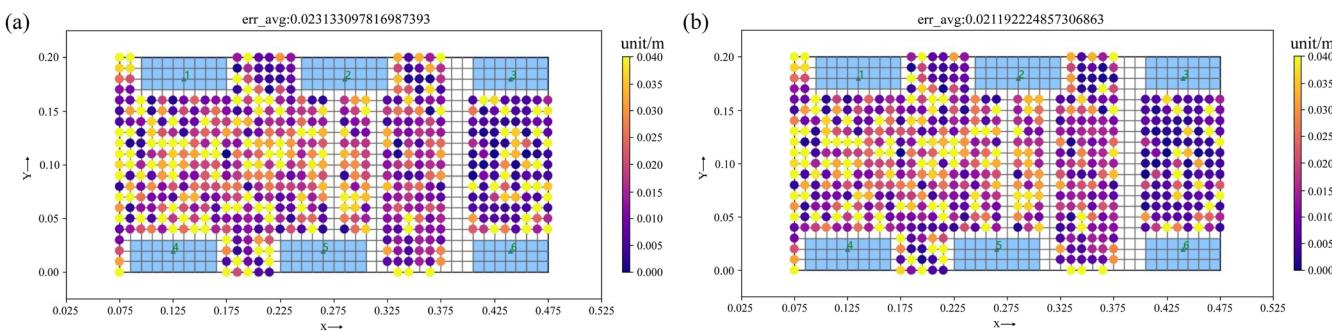


FIG. 14. Positioning error distribution map. (a) Distribution of errors in raw data; (b) distribution of errors in signal-to-noise separation data.

To qualitatively illustrate the effectiveness of the neural network model, the distribution of the acoustic emission source localization results at various stages can be plotted, as shown in Figs. 13(a) and 13(b).

From the time difference localization results, it can be seen that the distribution of localization points for the raw data is relatively chaotic, with many points located in areas between the vertical lines, which does not align with the actual lead break test procedure along the vertical lines, resulting in relatively large errors. After signal-to-noise separation, the number of localization points in the filtered data decreases compared to the raw data, and the overall level of chaos is reduced. The data processed by the neural network model show smaller errors compared to the original lead break test data, more closely matching the actual situation.

The average errors of the localization results for the raw data and the signal-to-noise separated data are calculated. A grid is drawn on the surface of the small beam specimen, and the localization error at each lead breakpoint is indicated by the color intensity in the grid, forming an error distribution map of the data. These are shown in Figs. 14(a) and 14(b), respectively. For ease of comparison, the upper limit of the color intensity variation is set to twice the mean of the average errors of the three data stages, with the error range (0, 0.04) linearly mapped to the color range (0, 1.0). Lead break points with errors greater than or equal to 0.04 m are marked in yellow.

From the error distribution maps, it is evident that the raw data have a relatively high average localization error, with a significant number of yellow points, and the overall color of the error distribution map is lighter. After signal-to-noise separation, the average localization error of the data shows a slight decrease compared to the raw data, with some light-colored lead break points transitioning to darker colors due to the model's noise filtering, indicating an improvement in localization accuracy. Looking at the overall data processing workflow, the neural network model significantly reduces the average error of the lead breakpoints, achieving good results and substantially improving the localization accuracy of the

acoustic emission sources, demonstrating the effectiveness of the neural network model.

To better summarize and compare the optimization effects of the neural network model on the data at each stage, an error parameter analysis statistics table was created using the calculation results, as shown in Table V. From the data in Table IV, it can be seen that the average localization error of the lead break test decreased after signal-to-noise separation compared to the raw stage. However, the maximum localization error did not decrease compared to the raw data because the lead break point with the maximum error did not lose its hit during the signal filtering process. The variance parameter reflects the fluctuation degree of the data errors. The variance of the signal-to-noise separated data decreases, indicating that the fluctuation of the errors also gradually decreases, and the distribution of the errors becomes closer to the average localization error.

Failure points are defined as lead break points with errors greater than 0.02 m. With the processing by the neural network model, the number of failure points for AE sources gradually decreases. The proportion of failure points to the total number of lead break points shows a significant reduction, indicating a substantial improvement in the localization accuracy of the lead break points. To more intuitively display the distribution pattern of localization errors in terms of numerical values, this study considers the maximum error value of 0.146 m from the raw data. The error range (0, 0.15) is divided into 100 small intervals, each with a width of 0.0015 m. The number of lead break points with errors within each interval is then counted. The error distribution statistics for the raw data and the signal-to-noise separated data are plotted, as shown in Figs. 15(a) and 15(b).

In Fig. 15, the horizontal axis represents the localization error of the lead breakpoints in meters, while the vertical axis represents the number of lead breakpoints within each error interval.

From the error distribution statistics of the raw data, it can be observed that most lead breakpoint errors are below 0.06 m,

TABLE V. Statistics of error parameter analysis.

Comparative	Average error (m)	Maximum error (m)	Variance	Number of failure points	Percentage of failure points (%)
Raw stage data	0.023 133	0.146 256	0.000 450	256	44.3
Signal-to-noise separation data	0.021 192	0.146 256	0.000 415	230	39.8

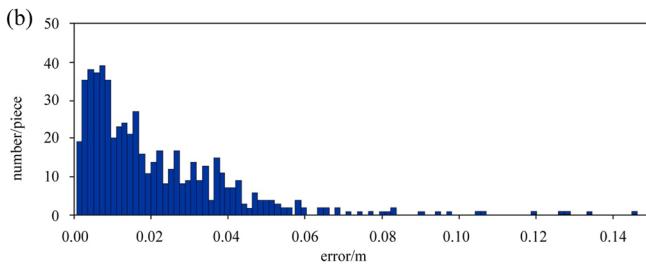
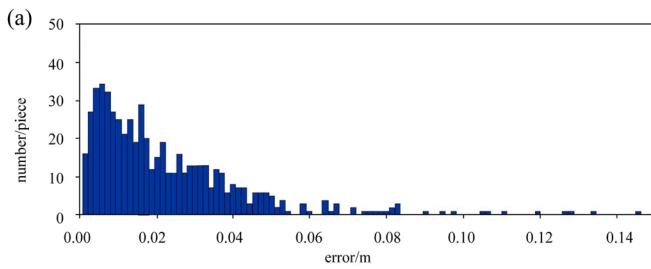


FIG. 15. Positioning error columnar distribution map. (a) Statistical results of raw data errors; (b) error statistics for signal-to-noise separation data.

although there are a few points with errors greater than 0.06 m. As the error value increases, the number of lead break points decreases. Comparing the localization results of the signal-to-noise separated data with the raw data, there is a slight increase in the number of lead break points with small errors below 0.01 m. The number of lead break points with errors in the range of 0.01–0.06 m slightly decreases, and a few points with errors greater than 0.06 m are eliminated, resulting in an overall reduction in average error. In summary, the signal-to-noise separation processing of acoustic emission signals by the neural network model has achieved good results and has played a positive role in improving the localization accuracy of acoustic emission sources.

IV. CONCLUSION

Based on the problem of acoustic emission (AE) signal and noise superposition, this study constructs an AE waveform screening model using an autoencoder and BP neural network, grounded on AE data obtained from laboratory experiments. The following conclusions are drawn:

- (1) The autoencoder achieved dimensionality reduction of features, thereby reducing the number of neurons and weight parameters in the subsequent neural network layers and significantly accelerating the training process. In addition, dimensionality reduction and extraction of key features contribute to improving the accuracy of subsequent classifiers or reducing the error of regressors.
- (2) The autoencoder and BP neural network model achieved a recognition rate of 96% for acoustic emission (AE) events and 98% for noise signals, demonstrating strong performance in classifying AE signals and noise.
- (3) After noise filtering, the localization error of acoustic emission (AE) sources in pencil lead break tests is significantly reduced, leading to improved localization accuracy. The number of AE events with large localization errors decreases noticeably, while the number of AE events with smaller errors increases accordingly. Conversely, the accuracy of AE source localization can serve as an important indicator for evaluating the effectiveness of noise filtering in AE signal processing.

ACKNOWLEDGMENTS

This research did not receive funding support. The authors would like to thank the anonymous reviewers for their valuable recommendations and comments, as they led to considerably enhanced paper quality.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Shunli Jiang: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Methodology (equal); Writing – original draft (equal). **Kangpei Zheng:** Software (equal); Supervision (equal);

Validation (equal); Visualization (equal); Writing – review & editing (equal). **Jinghui Jv:** Investigation (equal); Validation (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹M. Chai, X. Hou, Z. Zhang, and Q. Duan, “Identification and prediction of fatigue crack growth under different stress ratios using acoustic emission data,” *Int. J. Fatigue* **160**, 106860 (2022).
- ²G. Ciaburro and G. Iannace, “Machine-learning-based methods for acoustic emission testing: A review,” *Appl. Sci.* **12**(20), 10476 (2022).
- ³N. Ullah, Z. Ahmed, and J.-M. Kim, “Pipeline leakage detection using acoustic emission and machine learning algorithms,” *Sensors* **23**(6), 3226 (2023).
- ⁴H. Wei, H. Zhang, J. Li, J. Zheng, and J. Ren, “Effect of loading rate on failure characteristics of asphalt mixtures using acoustic emission technique,” *Constr. Build. Mater.* **364**, 129835 (2023).
- ⁵X. Xu, Z. Jin, Y. Yu, and N. Li, “Damage source and its evolution of ultra-high performance concrete monitoring by digital image correlation and acoustic emission technologies,” *J. Build. Eng.* **65**, 105734 (2023).
- ⁶X. Chiementin, D. Mba, B. Charnley, S. Lignon, and J. P. Dron, “Effect of the denoising on acoustic emission signals,” *J. Vib. Acoust.* **132**, 031009 (2010).
- ⁷M. G. Droubi, R. L. Reuben, and J. I. Steel, “Flow noise identification using acoustic emission (AE) energy decomposition for sand monitoring in flow pipeline,” *Appl. Acoust.* **131**, 5–15 (2018).
- ⁸Q. Hao, Y. Shen, Y. Wang, and J. Liu, “An adaptive extraction method for rail crack acoustic emission signal under strong wheel-rail rolling noise of high-speed railway,” *Mech. Syst. Signal Process.* **154**, 107546 (2021).
- ⁹K. Kuang, D. Li, and C. Koh, “Acoustic emission source location and noise cancellation for crack detection in rail head,” *Smart Struct. Syst.* **18**(5), 1063–1085 (2016).
- ¹⁰C. Zhou and Y. Zhang, “Particle filter based noise removal method for acoustic emission signals,” *Mech. Syst. Signal Process.* **28**, 63–77 (2012).
- ¹¹M. A. Abdelrahman, M. K. ElBatanouny, J. R. Rose, and P. H. Ziehl, “Signal processing techniques for filtering acoustic emission data in prestressed concrete,” *Res. Nondestr. Eval.* **30**(3), 127–148 (2018).
- ¹²F. Ren, C. Zhu, Z. Yuan, M. Karakus, S. Tang, and M. He, “Recognition of shear and tension signals based on acoustic emission parameters and waveform using machine learning methods,” *Int. J. Rock Mech. Min. Sci.* **171**, 105578 (2023).
- ¹³K. Ohno and M. Ohtsu, “Crack classification in concrete based on acoustic emission,” *Constr. Build. Mater.* **24**(12), 2339–2346 (2010).
- ¹⁴D. G. Aggelis, “Classification of cracking mode in concrete by acoustic emission parameters,” *Mech. Res. Commun.* **38**(3), 153–157 (2011).
- ¹⁵R. Huiyan, N. Jianguo, S. Shuizhou, and W. Zonglian, “Investigation on crack growth in concrete by moment tensor analysis of acoustic emission,” *Chin. J. Theor. Appl. Mech.* **51**(6), 1830–1840 (2019).
- ¹⁶X. Liu, Z. Liu, X. Li, F. Gong, and K. Du, “Experimental study on the effect of strain rate on rock acoustic emission characteristics,” *Int. J. Rock Mech. Min. Sci.* **133**, 104420 (2020).
- ¹⁷L. Dong, Y. Zhang, S. Bi, J. Ma, Y. Yan, and H. Cao, “Uncertainty investigation for the classification of rock micro-fracture types using acoustic emission parameters,” *Int. J. Rock Mech. Min. Sci.* **162**, 105292 (2023).
- ¹⁸A. K. Das, D. Suthar, and C. K. Y. Leung, “Machine learning based crack mode classification from unlabeled acoustic emission waveform features,” *Cem. Concr. Res.* **121**, 42–57 (2019).
- ¹⁹A. Thirumalaiselvi and S. Sasimal, “Pattern recognition enabled acoustic emission signatures for crack characterization during damage progression in large concrete structures,” *Appl. Acoust.* **175**, 107797 (2021).

- ²⁰A. Behnia, H. K. Chai, M. GhasemiGol, A. Sepehrinezhad, and A. A. Mousa, "Advanced damage detection technique by integration of unsupervised clustering into acoustic emission," *Eng. Fract. Mech.* **210**, 212–227 (2019).
- ²¹S. Ju, D. Li, and J. Jia, "Machine-learning-based methods for crack classification using acoustic emission technique," *Mech. Syst. Signal Process.* **178**, 109253 (2022).
- ²²Z.-H. Zhang and J.-H. Deng, "A new method for determining the crack classification criterion in acoustic emission parameter analysis," *Int. J. Rock Mech. Min. Sci.* **130**, 104323 (2020).
- ²³J. Jiang, G. Su, Z. Yan, Z. Zheng, and X. Hu, "Rock crack type identification by Gaussian process learning on acoustic emission," *Appl. Acoust.* **197**, 108926 (2022).
- ²⁴G. Siracusano, F. Lamonaca, R. Tomasello, F. Garesci, A. L. Corte, D. L. Carni, M. Carpentieri, D. Grimaldi, and G. Finocchio, "A framework for the damage evaluation of acoustic emission signals through Hilbert–Huang transform," *Mech. Syst. Signal Process.* **75**, 109–122 (2016).
- ²⁵J. Dai, J. Liu, L. Zhou, and X. He, "Crack pattern recognition based on acoustic emission waveform features," *Rock Mech. Rock Eng.* **56**(2), 1063–1076 (2023).
- ²⁶D. Li, Y. Wang, W.-J. Yan, and W.-X. Ren, "Acoustic emission wave classification for rail crack monitoring based on synchrosqueezed wavelet transform and multi-branch convolutional neural network," *Struct. Health Monit.* **20**(4), 1563–1582 (2021).