Received: 17 April 2024 | Accepted: 3 September 2024

DOI: 10.1111/mice.13342

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Diagnosis of high-speed railway ballastless track arching based on unsupervised learning framework

Xueyang Tang[1] | Yi Wang[1] | Xiaopei Cai[1] | Fei Yang[2] | Yue Hou[3]

[1]School of Civil Engineering, Beijing Jiaotong University, Beijing, China

[2]Infrastructure Inspection Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing, China

[3]Department of Civil Engineering, Faculty of Science and Engineering, Swansea University, Swansea, UK

Correspondence

Yi Wang and Xiaopei Cai, School of Civil Engineering, Beijing Jiaotong University, Beijing, China.
Email: 22121245@bjtu.edu.cn and xpcai@bjtu.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Number: 52178405; Fundamental Research Funds for the Central Universities, Grant/Award Number: 2022JBQY009

Abstract

Vehicle-mounted detection methods have been widely applied in the maintenance of high-speed railways (HSRs), providing feasibility for diagnosing ballastless track arching. However, applying detection data faces several key limitations: (1) The threshold mostly requires manual setting, making recognition accuracy highly subjective; (2) the extensive workload of manual inspections makes it challenging to label detection data, hindering the application of supervised learning approaches. To address these problems, this paper utilizes the longitudinal level irregularity data obtained from vehicle-mounted detection, employing the concept of unsupervised learning for dimensionality reduction, combined with clustering algorithms and minimal label fine-tuning, to design two frameworks: the fully unsupervised framework (FUF) and the few-shot fine-tuned framework (FFF). Experiments on dynamic detection data from a Chinese HSR line were conducted, comparing the performance of data dimensionality reduction, clustering, and classification under different strategy combinations. The results show that the improved variational autoencoder significantly enhances the performance of the encoder in dimensionality reduction, facilitating better feature extraction; the FUF achieves effective clustering outcomes without any labeled samples and its adjusted rand index score exceeded 0.8, showcasing its robustness and applicability in scenarios with no prior annotations; the FFF requires only a small number of labeled samples (labeling ratio of 5%) and achieves excellent performance, with metrics such as accuracy exceeding 0.85, thus greatly reducing the reliance on labeled data. This study offers a novel method for solving engineering issues with limited labeled data, providing an efficient solution for identifying track arching defects and advancing railway infrastructure monitoring.

1 | INTRODUCTION

Ballastless track has been widely used in the construction of high-speed railway (HSR) all over the world due to its advantages of high stability, rigidity uniformity, and low maintenance (Cai, Tang, Pan, et al., 2023). However, in the process of long-term operation, affected by the high-frequency impact load of the train and complex environmental conditions, track structure has appeared with different degrees of disease, such as interlayer debonding of the track (Peng et al., 2019; Ren et al., 2020), void of mortar layer (Ren et al., 2016; H. Xu et al., 2013), wide and narrow joints defects (Z. Li et al., 2020; Liu et al., 2019), track arching deformation (Tang et al., 2023; Yu et al., 2018), and so on. Among these diseases, the arching of the track plate is the most common (Zhang et al., 2024). These diseases seriously affect the stability, durability, and smoothness of the track and bring great safety risks to the normal operation of HSR. Therefore, it is particularly important to efficiently and accurately achieve the detection and diagnosis of ballastless track diseases (Tang et al., 2023).

Scholars have conducted extensive research on the detection methods of ballastless track diseases. These methods include manual detection (Shao et al., 2013), impact echo method (J. Xu et al., 2018), ultrasonic method (Fan et al., 2019), image method (W. Li et al., 2018), ground-penetrating radar method (Liao et al., 2016), and infrared method (Z. Li et al., 2020). In a comprehensive analysis, the detection and assessment methods of ballastless track diseases still face several challenges. Current methods primarily focus on measuring specific arch sections to quantitatively assess their deformation. However, these approaches are costly and fail to provide rapid detection across the entire railway line. Moreover, they typically require railway engineering personnel to operate during limited skylight periods, resulting in a complex process with relatively slow speeds. Given the comprehensive detection vehicle's ability to collect a large amount of track irregularity and vehicle response data (Zhang et al., 2016), it is possible to identify arching diseases on the track slab by installing acceleration sensors on the train (Ngamkhanong et al., 2018), a method known as vehicle-mounted detection. This method has high efficiency (Tian et al., 2020; Yang et al., 2020). However, when using this method to identify arching diseases, it is necessary to artificially set the threshold for these diseases (Hong, 2020; Z. Li et al., 2019). Additionally, due to the difficulty of on-site reinspection, supervised identification methods are constrained by data labels, presenting certain challenges. Unsupervised learning, which does not rely on subjective experience and human judgment, emerges as a promising direction in the field of track detection.

To accurately identify arching defects on track slabs with minimal labeled data, this study proposes two frameworks for arch diagnosis on ballastless tracks using unsupervised learning: the fully unsupervised framework (FUF) and the few-shot fine-tuning framework (FFF). The FUF can uncover latent patterns from unlabeled data, making it ideal for HSR monitoring where labeled data are scarce. This approach enhances defect detection's flexibility and comprehensiveness, aiding decision-making. The FFF, on the other hand, is both flexible and efficient, enabling effective training with limited samples and quickly adapting to various problems. This ensures reliable model adjustment and prediction even in data-scarce environments, suitable for rapid analysis of emerging issues. These frameworks offer railway engineers the flexibility to choose based on actual situations and better meeting maintenance and operational needs. The flowchart of the research steps is shown in Figure 1.

The study's innovations include: (1) Optimization of the spatial structure of the variational autoencoder (VAE), significantly improving its dimensionality reduction performance. (2) An innovative FUF framework combined with a clustering algorithm, ensuring effective clustering even with entirely unlabeled training samples. (3) Enhancement of the encoder with several fully connected layers and fine-tuning with a small number of labeled samples to form the FFF. This allows the encoder to achieve significant performance improvements with minimal labeled data. These frameworks provide significant convenience to field engineers, which is crucial in engineering practice.

This study is organized as follows: Section 2 presents previous research on railway disease diagnosis. Section 3 introduces the dimensionality reduction methods and the disease diagnosis framework. Section 4 presents the principles of the track inspection system, the track inspection data, and the experimental setup. Section 5 showcases the experimental results of this study, while Section 6 provides the conclusions.

2 | RELATED WORK

Currently, dynamic track inspection, as a mature onboard detection method, has been adopted by many national railway maintenance departments and is widely applied in the diagnosis and evaluation of track arching defects. However, existing research mainly relies on the geometric features of the track for the qualitative detection of arching locations, rarely utilizing the dynamic response characteristics caused by the arching, and there is no research on the quantitative estimation of the actual arching displacement. The accuracy of identification methods based

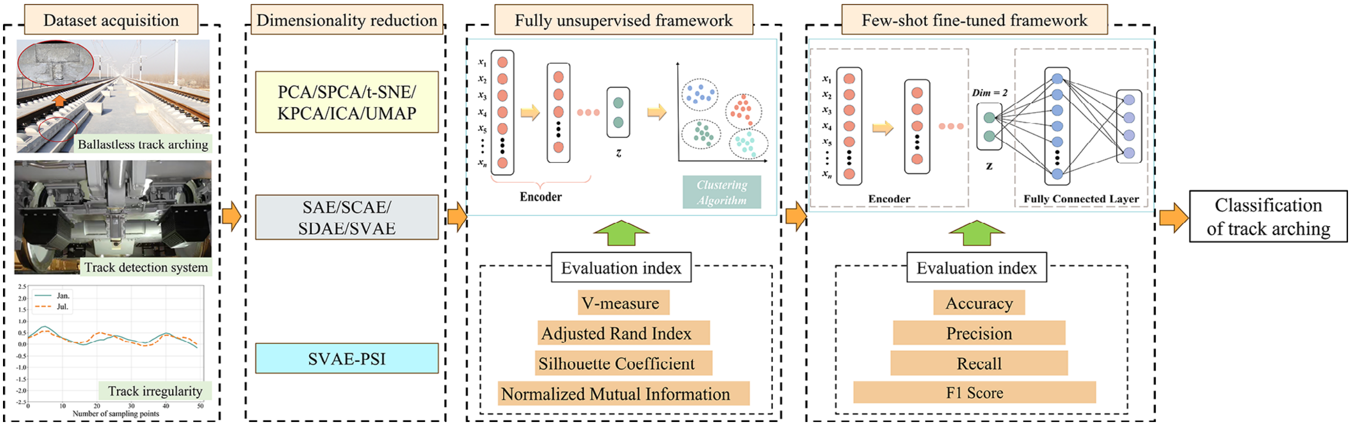


FIGURE 1 Flowchart of unsupervised learning-based framework for recognizing arches on high-speed railway ballastless track.

on track geometry needs further improvement, making it difficult to effectively detect minor arching defects.

With the rapid advancement of computer technology, deep learning techniques have seen significant progress (Hassanpour et al., 2019; Martins et al., 2020) and have been widely applied in the railway. These applications include railway maintenance plan optimization based on track segment conditions (Chang et al., 2023), railway track detection using drone imagery (Tong et al., 2023), 3D track optimization for dedicated HSRs (Song et al., 2022), and geographic information modeling for railway alignment optimization (Pu et al., 2023). Additionally, there is research on the rapid generation of railway lines using multi-touch gestures (Nie et al., 2023). The methods, however, are significantly data-dependent and require large amounts of labeled data for training, which is difficult to obtain in the field of railway track detection. In the field of structural defect detection, deep learning has demonstrated excellent generalization capabilities when faced with unknown datasets. Compared to traditional methods, it excels in learning complex categories and boundaries (Rafiei & Adeli, 2016). Although Adeli and Kamal (1989) did not directly address defect detection, their research laid the foundation for parallel processing in the field of structural engineering. Furthermore, Adeli (2001) provided a review of the application of neural networks in civil engineering, offering an important background for understanding the potential of deep learning.

For HSR infrastructure, manually labeling the track slabs is a challenging task due to the high-speed movement of trains and the limited window of opportunity (Shao et al., 2013). The data collected by detection or monitoring devices often include a large amount of unlabeled data, which cannot be fully utilized by supervised deep neural network models (Pan et al., 2023; Yang et al., 2021), resulting in the waste of useful information. Unsupervised learning can effectively address this issue (Zhao et al., 2021), including research on unsupervised profile evaluation of rail joints (Cong et al., 2023), unsupervised defect segmentation (Midwinter et al., 2023), a general unsupervised novelty detection framework (Soleimani-Babakamali et al., 2022), dynamic learning rates for continual unsupervised learning (Fernandez-Rodrıguez et al., 2023), and unsupervised domain adaptive dose prediction (Cui et al., 2023). Furthermore, in the field of railroad inspection, multidimensional inspection and monitoring data can significantly increase the complexity of data-driven models. Therefore, dimensionality reduction methods have become an effective means to deal with high-dimensional data, including the application of principal component analysis (PCA) methods to the prediction of the evolution of wheelset wear (Braga & Andrade, 2021), and the use of autoencoder (AE) for track injury monitoring in urban railways (J. Chen et al., 2024) versus the detection of obstacles and soiling in railroad tracks (Jahan et al., 2021), among others. These dimensionality reduction methods are combined with clustering algorithms such as Ordering Points to Identify the Clustering Structure to form an unsupervised deep learning framework, which is being used in the identification of injuries and damages to structures such as railway bridges (S. Li et al., 2024), helping to reduce the computational complexity and providing a new perspective for optimizing the model.

3 | METHODOLOGY

3.1 | Dimensionality reduction

In the realm of HSR detection and monitoring, the accumulation of massive high-dimensional data typically brings along redundancy and noise, potentially

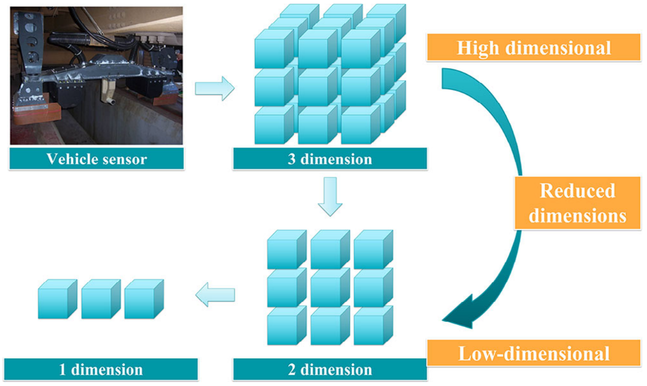


FIGURE 2 Schematic diagram of dimensionality reduction.

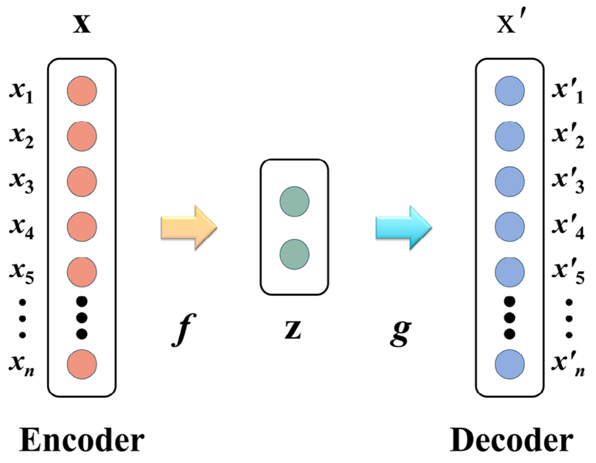


FIGURE 3 Structure of the autoencoder (AE).

leading to errors and decreased accuracy. Adeli and Wu (1998) initially highlighted the challenges of processing high-dimensional data in civil engineering. Generally, an excess of features will affect the effectiveness of classification and clustering tasks and significantly increase the computational time required for complex models. To address this issue, dimensionality reduction algorithms have become essential in machine learning (Kakarla et al., 2020; Rodriguez-Lozano et al., 2023). As illustrated in Figure 2, the high-dimensional spatial data obtained from the sampling of vehicle sensors can be effectively reduced to within the low-dimensional space after being processed by the dimensionality reduction algorithm. These algorithms facilitate the extraction of crucial information by minimizing data dimensions, which enhances the accuracy of downstream processes and boosts efficiency (Hinton & Salakhutdinov, 2006).

As shown in Equation (1) (Jiang et al., 2021), the essence of dimensionality reduction is to learn a mapping function f. It reduces the D-dimensional real number set $\mathbb{R}$ to d dimensions, where $1 \le d \le D$. The mapping function f may be linear or nonlinear, and common mapping functions (models) include PCA, t-distributed stochastic neighbor embedding (t-SNE), and so on.

$$f : \mathbb{R}^D \to \mathbb{R}^d \tag{1}$$

3.1.1 | AE

In 1986, Rumelhart et al. (1986) first introduced the concept of AE and applied it to the processing of high-dimensional complex data. As shown in Figure 3, the AE consists of two parts: an encoder and a decoder. The encoder transforms the high-dimensional input data $\boldsymbol{x}$ into a lower-dimensional code $\boldsymbol{z}$; meanwhile, the decoder attempts to reconstruct the input data from this low-dimensional encoding, generating the reconstructed input $\boldsymbol{x}'$. The specific representation of the encoder and decoder are as follows (Rifai et al. 2011):

$$\boldsymbol{z} = f(\boldsymbol{x}; \theta_e) = f(\boldsymbol{W}1\boldsymbol{x} + \boldsymbol{b}1) \tag{2}$$

$$\boldsymbol{x}' = \mathrm{g}(\boldsymbol{z}; \theta_{\boldsymbol{d}}) = \mathrm{g}(\boldsymbol{W}2\boldsymbol{z} + \boldsymbol{b}2) \tag{3}$$

where x and z are the high-dimensional input sample and lower-dimensional code, $x \in \mathbb{R}^D$, $z \in \mathbb{R}^d$, $D \ge d$. $\theta_{\boldsymbol{e}}$ is the encoder parameters, which consist of $W1$ and $b1$, and $f(\cdot)$ is the encoder activation function. x' is the reconstructed input sample, $x' \in \mathbb{R}^D$. $\theta_{\boldsymbol{d}}$ is the decoder parameter, which consist of $W2$ and $b2$, and $g(\cdot)$ is the decoder activation function.

The learning objective of the AE is to minimize the reconstruction error. Hence, its loss function can be defined as (Giglioni et al., 2023):

$$\boldsymbol{L}(\mathbf{x}, x') = \boldsymbol{L}(x, \mathrm{g}(z; \theta_{\boldsymbol{d}})) = \boldsymbol{L}(x, \mathrm{g}(f(x))) \tag{4}$$

where $g(f(x))$ is a composite function, and for a perfect AE, it will be a unit (identity) function $I(x) = x$. The parameters of the network can be effectively learned by minimizing the reconstruction error with the backpropagation algorithm.

The primary distinction between traditional neural network and AE lies in training supervision signals. For AE, the signal is not external labels $\mathbf{y}$ but the input data x. This allows AE to achieve an efficient representation of data through the nonlinear feature extraction capabilities of neural networks. Moreover, in comparison to linear methods, AE provides enhanced performance, enabling a more precise recovery of the input x. During this process, the encoder part of the AE is capable of learning a compressed representation of the input data, thereby facilitating effective data dimensionality reduction and feature extraction.

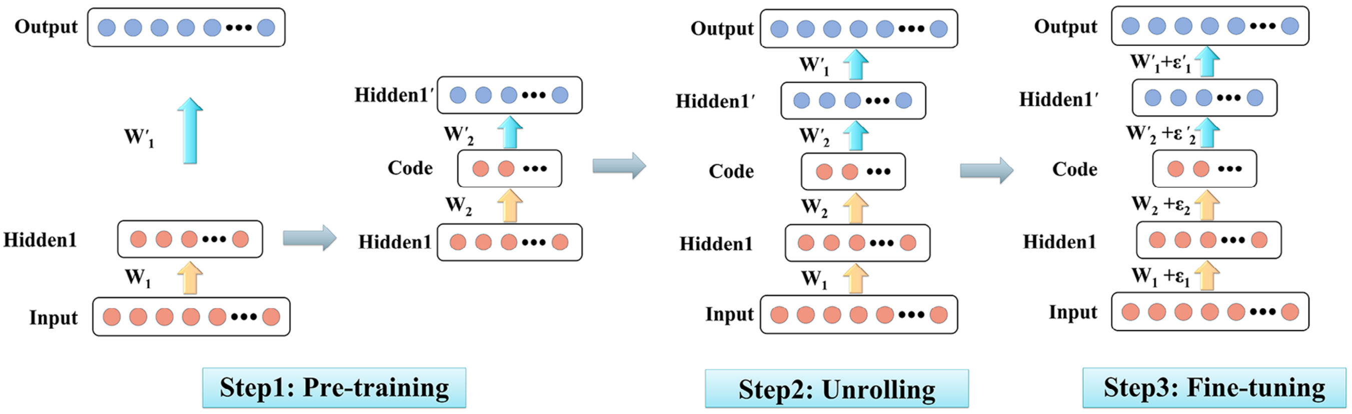


FIGURE 4 Greedy layer-wise pre-training.

3.1.2 | Stacked VAE (SVAE)

When training neural networks using gradient descent, the problem of vanishing gradients becomes significant as the number of network layers increases, which increases the difficulty of training deep network and limits its widespread application. To overcome this challenge, Hinton et al. (2006) improved upon the traditional AE proposing stacked AEs (SAEs), also known as deep AEs (DAEs). These models leverage multiple layers of encoding and decoding to learn more complex representations, effectively mitigating the vanishing gradient issue. Specifically, in SAE, the encoder and decoder usually have the same number of layers and are symmetrically arranged. To enhance efficiency further, the parameters of these symmetrical layers can have a transposed relationship, a design known as weight tying. This approach not only reduces the model parameters by half, speeding up the training process, but also significantly reduces the risk of overfitting.

Additionally, to effectively train deep networks, Hinton creatively introduced a unique training methodology: greedy layer-wise pre-training. That is, an initialized parameter $\mathbf{W_i}$ is formed by layer-by-layer individual training. Specifically, when training the 1 layer, the parameters of the 1~(L − 1) layers are frozen, thereby achieving independent training for each layer. Afterward, all layers are unrolled together, and the entire neural network undergoes parameter optimization adjustments using algorithms like backpropagation. This results in adjusted parameters $\mathbf{W_i} + \varepsilon_\mathbf{i}$. This training strategy, illustrated in Figure 4, effectively addresses the issue of the backpropagation algorithm easily getting trapped in local minima, thus enabling efficient training of deep neural networks (Bengio et al., 2007).

However, during the training phase, the encoder's search for a probability distribution within the latent space lacks specific constraints (such as ensuring Gaussian characteristics). This makes decoding samples from the latent space to produce meaningful outputs a challenging task (Sajedi & Liang, 2022). VAE addresses the aforementioned issue by employing variational inference, introducing a different perspective and methodology (Kingma & Welling, 2014). As a deep generative model, its primary objective is to establish a probabilistic model that can not only describe the data generation process but also learn the distribution of latent variables from observed data. Unlike traditional AEs that describe the latent space directly in a numerical form, VAEs observe the latent space in a probabilistic manner, encoding input data x into a standard Gaussian distribution. This constraint helps to regularize the latent space, thereby reducing the risk of model overfitting.

Based on the mentioned VAE and SAE, an SVAE model is constructed as illustrated in Figure 5a. In the SVAE, the role of the encoder is to output the key parameters of the latent variable distribution: mean and variance. They describe the Gaussian distribution of the latent space. And the reparameterization trick is applied to resample this distribution to obtain specific latent feature values, a process depicted in Equation (5) (Hoffman et al., 2013).

$$z = \mu + \sigma \cdot \varepsilon \tag{5}$$

where μ and σ are the mean and standard deviation for the latent variable $\mathbf{z}$. ε is a noise term following a standard normal distribution, $\varepsilon \sim \mathcal{N}(0, 1)$.

The loss function of SVAE comprises two components: reconstruction loss and Kullback–Leibler (KL) divergence. The optimization goal is to minimize the difference between the input x and the reconstructed output x' and to ensure that the distribution of the encoded latent variables z closely aligns with a predetermined prior distribution. This can be specifically represented as (Kingma & Welling, 2014):

$$\boldsymbol{L}(\mathbf{x}, \mathbf{x}') = -Eq\phi(\mathbf{z}|\mathbf{x})[logp\theta(\mathbf{x}|\mathbf{z})] + KL[q\phi(\mathbf{z}|\mathbf{x}) \,\||\, p(\mathbf{z})] \tag{6}$$

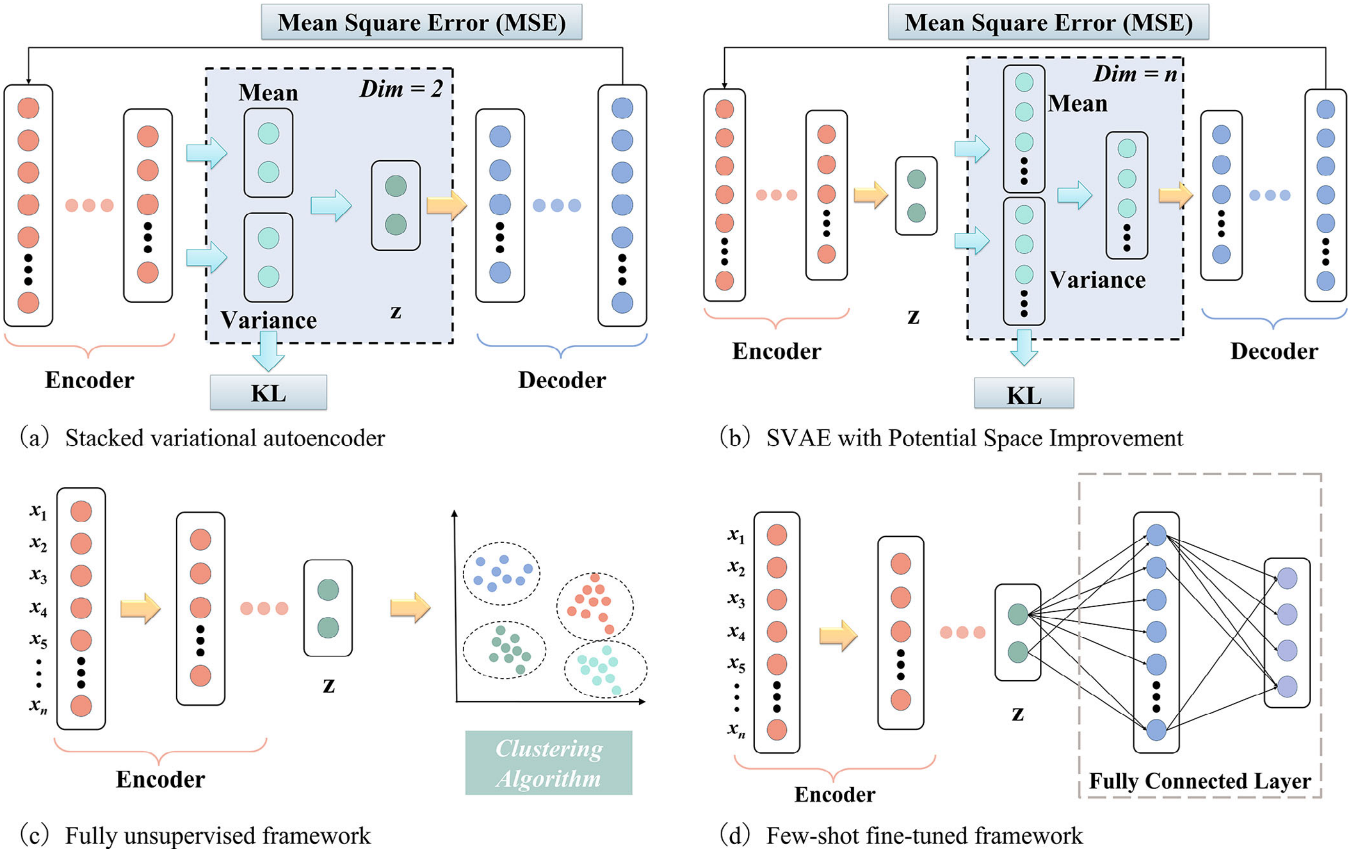


FIGURE 5 Structure of stacked variational AE and proposed framework (z is latent variable). KL, Kullback–Leibler; SVAE, stacked variational autoencoder.

where the first term is the reconstruction term, which encourages the model to reconstruct the input data x based on the latent variable z. The second term is the regularization term, which is the KL divergence between $q\phi(\mathbf{z}|\mathbf{x})$ and the prior distribution $p(\mathbf{z})$, ensuring that the distribution of the latent variables does not deviate from the prior distribution.

Training the SVAE involves optimizing the parameters of the generative model θ and the variational parameters ϕ simultaneously. The objective is to maximize the Evidence Lower Bound Objective, which is equivalent to minimizing the loss function.

3.1.3 | Improved AE design

This study employs a dimensionality reduction algorithm to reduce high-dimensional detection data to low-dimensional potential space. However, when VAE is applied to the low-dimensional potential space, it is prone to problems such as loss of information and degradation of reconstruction quality, which is hypothesized to be that the data distribution of the potential space is not suitable for the dataset of this study. Based on this, this study proposes a new strategy.

As illustrated in Figure 5b, a high-dimensional latent space is introduced on the basis of the original model to assist in the computation of the KL scatter, while a two-dimensional latent space is reserved exclusively for extracting the downscaling results. These data are subsequently used for clustering and classification tasks. This strategy not only enhances the model's reconstruction quality and performance in subsequent tasks but also offers a new perspective in the analysis and dimensionality reduction of high-dimensional data. This enhanced version of the model focusing on potential spatial improvement (PSI) is named SVAE with PSI (SVAE-PSI).

3.2 | FUF design

The previous section explains the improvements made in this study in terms of dimensionality reduction. As shown in Figure 5c, the hidden layer coding z with dimension 2 was obtained after the dimensionality reduction process by using SVAE-PSI. Then, z was used as the input to the clustering algorithm, and the highly flexible clustering capability of methods such as the Gaussian mixture model (GMM) was utilized to successfully achieve a diagnosis of arches on the track structure. This training process is entirely independent of any label information, hence the name FUF. This framework not only significantly enhances the precision of clustering but also provides an effective solution for practical issues in the HSR domain, where labels are difficult to obtain.

Algorithm 1 details the training process of the FUF. First, the track inspection data collected by the onboard sensors is preprocessed to obtain the training and test sets. Next, the encoder network parameters are initialized, and a layer-wise greedy pre-training strategy along with a global fine-tuning strategy is employed to train the dimensionality reduction encoder network. After completing the network training, the number of clusters n is specified, and the dimensionality-reduced results are input into the GMM for clustering, ultimately returning the cluster labels for each group.

3.3 | Few-shot fine-tuned framework (FFF) design

Considering that railway workers routinely evaluate the condition of the track structure during regular inspections, thereby obtaining a small amount of labeled data samples, this study proposes the FFF. As shown in Figure 5d, the dimensionality reduction results are fed into a fully connected layer network to perform the supervised classification task. In addition, during the model fine-tuning process, there is flexibility to choose whether to freeze the encoder parameters or not, depending on the number of samples used (to be discussed further). This provides field workers with greater operational flexibility.

Algorithm 2 describes the training process of the FFF. The data preprocessing and dimensionality reduction steps are the same as in Algorithm 1. After obtaining the trained encoder network, the fully connected layers are added, and the encoder network parameters are frozen. The fully connected layers are trained separately using a small number of labeled samples. This training is conducted for 50 epochs to obtain the framework parameters. The classification results of the model are then validated based on the training set.

ALGORITHM 1 Fully unsupervised framework

```
Input:  Training and test datasets $X = [\mathbf{x1}, \dots, xm] \in \mathbb{R}^{D\times m}$
        and $T = [\mathbf{t1}, \dots, tn] \in \mathbb{R}^{D\times n}$   ▷$D$ is the dimension
        of original data, $m$ and $n$ are the number of training
        and test sets
Output: Cluster label $\hat{Y}$; encoder network $\theta_e$
Dimensionality reduction phase
    Initializes the encoder and decoder network, based
    on the Kaiming Uniform (He Uniform) initialization
    method
    // Pre-training
    (a) For $i = 1$ to the number of encoder layer (refer to
    Table 3 for setting the number of layers)
      Freeze encoder and decoder parameters $\theta_e, \theta_d$
      Unfreeze encoder layer $i$ and symmetric decoder
    layer $j$ parameters
      (b) For k = 1 to 50 epochs (number of pre-training)
        Input X in batches of size 32
        Forward propagation
        Calculate the reconstructed $X'$ and the mean
    and $\sigma$ variance $\mu$ of the latent spatial distribution.
        Calculate the loss function by Equation (6)
        Backpropagation to update parameters using
    Adam optimizer with learning rate 0.001
      End for
    End for
    // Fine-tuning
    Unfreeze all layer parameters
    (c) For $i = 1$ to 50 epochs (number of fine-tuning)
      Repeat (b)
    End for
    Return encoder network $\theta_e$
    (d) For $i = 1$: $n$
      ri =$encoder(\theta\boldsymbol{e}; \mathbf{ti})$   ▷$\mathbf{ti} \in \mathbb{R}^D, \mathbf{ri} \in \mathbb{R}^2$
    End for
    Return two-dimensional test datasets $R \in \mathbb{R}^{2\times n}$
Clustering phase (Take GMM for example)
    n_cluster = number of disease category
    Initialize the GMM based on random or k-means
    Fitting GMM to $R$ using the EM algorithm
    (e) For $i = 1$: $n$
      $\hat{\mathrm{yi}}=GMM(\mathbf{ri};n\_cluter)$
    End for
    Return test cluster label $\hat{Y} = [\widehat{\mathbf{y1}}, \dots, \widehat{\mathbf{yn}}] \in \mathbb{R}^n$
```

4 | EXPERIMENT DESIGN

4.1 | Data acquisition

The track irregularity detection data selected in this study comes from an operational HSR in China, with CRTS II ballastless track. Since the operation of the railway, the track irregularity detection system has regularly detected the status of the line and accumulated a large amount of detection data.

ALGORITHM 2 Few-shot fine-tuned framework

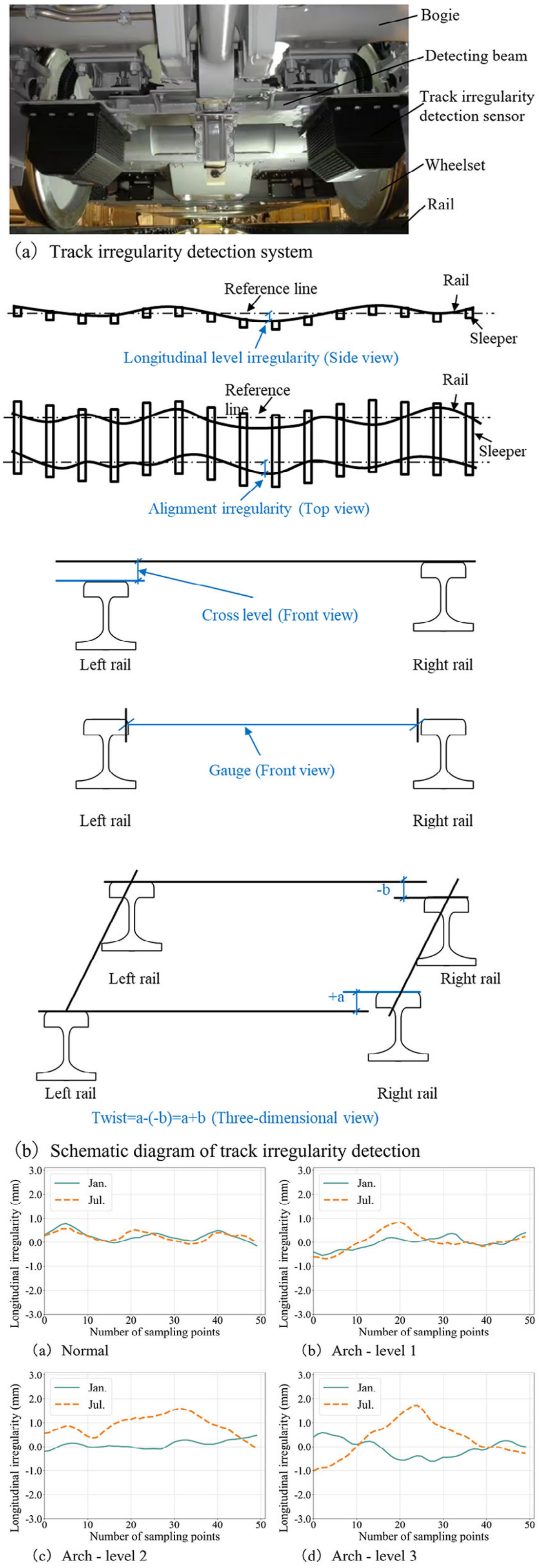
```
Input:   Training and test datasets $X = [\mathbf{x1}, \dots, xm] \in \mathbb{R}^{D\times m}$
         and $T = [\mathbf{t1}, \dots, tn] \in \mathbb{R}^{D\times n}$, small training sets label
         $S = [\mathbf{s1}, \dots, sp] \in \mathbb{R}^{p}$ ▷$D$ is the dimension of original
         data, $m$ and $n$ are the number of training and test
         sets, $p$ is the number of labeled training set, $p \le m$
Output:  Classify label $\hat{Y}$;
         Few-shot fine-tuned framework $\theta$
Pre-training phase
1:       Same as Algorithm 1 Steps 1 to 7
2:       Return encoder network $\theta_e$
Fine-tuning phase
3:       Extract the encoder network structure
4:       Add fully connected layers referring to Figure 5c
5:       Load parameters $\theta_e$ and freeze the encoder layer
6:       (a) For $i = 1$ to 50 epochs (number of fine-tuning)
             Input labeled set $XL$ forward propagation
             Calculate of classification results $\hat{S}$
             Calculate the loss function by loss $(\hat{S}, S)$
             Backpropagation to update parameters
7:       End for
Return few-shot fine-tuned framework $\theta$
Classification phase
8:       Load parameters $\theta$
9:       Input test datasets $T$
10:      Return classify label $\hat{Y}$=$FFF(\theta;T)$
```

4.1.1 | Track irregularity detection system

The track irregularity detection system mainly consists of a laser camera component and an inertial measurement component (Cai, Tang, Yang, et al., 2023). The laser camera component measures the lateral displacement and vertical displacement of the rail relative to the detection beam by processing the image information of the collected rail profile. The inertial measurement component is mainly composed of gyro platform, accelerometers, and so on. The main function of the inertial measurement component is to establish the inertial reference for track detection, which is shown in Figure 6a.

The detection items of the track irregularity detection system include gauge, alignments, longitudinal level, cross level, twist as shown in Figure 6b. The spatial sampling interval of the irregularity data is 0.25 m, and the detailed data acquisition process can be found in the literature (S. Wei et al., 2011). The description of various types of track irregularities is as follows:

1) Longitudinal level irregularity refers to the unevenness of the track in the vertical direction along the length of the rail.



(a) Track irregularity detection system

(b) Schematic diagram of track irregularity detection

(c) Measured data of track irregularity

FIGURE 6 Acquisition and presentation of track irregularity.

2) Alignment irregularity refers to the unevenness of the track in the transverse direction along the length of the rail.
3) Cross level refers to the difference in height between the top surfaces of the two rails at the cross-section of the track.
4) Gauge refers to the deviation of the minimum inner distance between the two rails at the same cross-section of the track, 16 mm below the top surface of the rails, from the standard gauge.
5) Twist refers to the algebraic difference in the cross-level amplitudes between two track cross-sections that are separated by a certain distance.

4.1.2 | Dataset

Longitudinal level irregularity refers to the unevenness of the track along the length of the rail, in the vertical direction. When the track is arched, the rail will have additional irregularity with the deformation of the track, which makes the longitudinal level irregularity appear as an arching feature (Z. Chen et al., 2024). As a regular item of the track detection system, the irregularity is obvious, and the feature position corresponds well with the position of the arching, which provides an opportunity for the rapid investigation and positioning of the arching disease.

This paper collects a large amount of inspection data from both normal sections and sections with arching. First, the data are classified using clustering algorithms, revealing that they can be divided into four clusters. These four clusters are then compared with the actual arching conditions recorded in the maintenance logs of HSRs. One cluster corresponds to normal sections, while the other three clusters correspond to slightly arching sections (Level 1), moderately arching sections (Level 2), and severely arching sections (Level 3), as shown in Figure 6c.

Arching is determined by differences in longitudinal level irregularity data. Damaged connections between track slabs cause arching at high temperatures, resulting in larger irregularity amplitudes, while low temperatures lead to smaller amplitudes. Significant differences in irregularities between low- and high-temperature months indicate arching, whereas minimal differences suggest no arching. It compares longitudinal-level irregularities in January (low temperature) and July (high temperature). In normal sections, irregularity amplitudes are consistent across months. In arched sections, however, significant differences in July compared to January indicate the severity of arching, with greater differences reflecting more severe conditions.

This paper detects arching defects based on the measured data of track irregularities. The mileage deviation between the identified arching defects and the actual arching defects depends on the deviation between the mileage of the track irregularity detection data and the actual mileage. In fact, the authors have conducted extensive work on correcting the mileage deviation in track irregularity detection data using the positioning index, confidence index (Qin et al., 2024), and dynamic time warping method (H. Wei, Yang, Wu, et al., 2022; H. Wei, Yang, Zhu, et al., 2022). After validation, the deviation between the detection data mileage and the actual mileage can be controlled within 0.5 m.

Due to the relatively weak connection between two track slabs, the maximum arching typically occurs at the junction of the two slabs. The arching magnitude gradually decreases as the distance from the junction increases, eventually reaching zero. Therefore, the arching of the track slab does not occur at a specific mileage point but rather forms within a mileage range. This study refers to this as the “track slab arching interval.” The length of the arching interval is typically 12.5 m, which includes 50 sampling points of track irregularity. The purpose of this research is to utilize detection data to identify whether a certain interval is an arching interval; if so, it is believed that there is an arching defect within the range of that interval. Based on the deviation between the detection data and the actual mileage, the deviation between the arching interval and the actual mileage can be derived.

Assume the starting point of the detected arching interval is $D1$ (detection data), and the actual starting point is $R1$. Therefore, it holds that:$|D1{-}R1|\leq 0.5$ m.

If the starting point of the arching interval is $D1$, then the endpoint $D2$ is: $D2 = D1 + 12.5$ m.

The actual endpoint $R2$ is: $R2 = R1 + 12.5$ m.

The deviation of the arching interval’s starting and ending points from the actual mileage is: $|D2{-}R2| = |(D1{+}12.5){-}(R1{+}12.5)| = |D1{-}R1|\leq 0.5$ m.

Thus, the deviations of both the starting and ending points of the arching interval from the actual mileage are within 0.5 m. This means that the overall deviation of the arching interval is also within 0.5 m, which is acceptable for railway maintenance personnel. In this study, data preprocessing mainly includes the alignment, cutting and division of 2 months’ track inspection data. The alignment process is as follows:

1) First, the track irregularity data for January and July are obtained, denoted as $f(x)$ and $g(x)$, respectively.
2) The maximum correlation method is employed to align the mileage of the data from the 2 months: for the two discrete irregularity datasets $f(x)$ and $g(x)$, let $g(x)$ be shifted by m units, denoted as $g(x{+}\mathrm{m})$. The correlation

TABLE 1 Sample composition of the dataset.

Name	Training set	Test set	All
Normal	900	100	1000
Arch-level 1	360	40	400
Arch-level 2	180	20	200
Arch-level 3	180	20	200
All	1620 (90%)	180 (10%)	1800

distribution between *f*(*x*) and *g*(*x*+m) is calculated for different values of m to find the value of m that corresponds to the maximum correlation coefficient. This *m* represents the phase difference between the irregularity data *f*(*x*) and *g*(*x*). By shifting *g*(*x*) by *m* units, the alignment of the two sets of irregularity data can be achieved.

After data preprocessing, 1800 sets of track detection data with track irregularity are used, each data segment contains 50 measuring points (measuring point spacing 0.25 m), and the longitudinal length along the railway is 12.5 m (covering the complete track slab size). Since the training set has a small amount of data, in order to avoid the risk of overfitting, this study used 10-fold cross-validation to fully evaluate the performance of the SVAE-PSI model. Therefore, in this study, the dataset was divided into a training set and a test set at a ratio of 9:1. The specific amount of data is detailed in Table 1.

4.2 | Comparative experiments

Corresponding to Section 2, three sets of comparison experiments are designed to focus on the visualization results of dimension reduction algorithms, the performance of unsupervised clustering, and the effects of classification after fine-tuning. They were all trained under the same conditions to reduce the error introduced by randomness. Both clustering and classification experiments were repeated three times, with the results averaged.

4.2.1 | Comparison of dimensionality reduction algorithms

As shown in Table 2, three sets of experiments, traditional dimensionality reduction algorithm, encoder-based dimensionality reduction algorithm, and improved encoder-based dimensionality reduction algorithm, were used for the comparison of dimensionality reduction algorithms. The test set was reduced to a two-dimensional space, and the dimension reduction effects of different methods were observed through visual scatter plots.

The AE configurations used in this study are shown in Table 3, including four types: SAE, stacked denoising AEs (SDAEs), stacked contractive AE (SCAE), and SVAE. Additionally, on the basis of the four SVAE, a latent space ranging from 10 to 100 dimensions (with a step size of 10) was added, resulting in a total of 40 groups of SVAE-PSI.

It is important to note that in this study, we set the dimensionality of the low-dimensional space to 2 and used the two-dimensional output as input for subsequent tasks. This parameter setting is based on early experimental attempts. As shown in Table 4, we employed two models, SVAE-1 and SVAE-PSI-1 (with a latent space dimension of 50), for the dimensionality reduction task and used the GMM method as the clustering algorithm. The table presents the adjusted rand index (ARI) under different dimensional outputs.

4.2.2 | Comparison of FUF

As shown in Table 5, after reducing the dimensionality of high-dimensional data using different algorithms, the two-dimensional samples were input into five types of clustering methods: KMeans, agglomerative hierarchical clustering, density-based spatial clustering of applications with noise, GMM, and fuzzy c-means, to achieve completely unsupervised classification of the arching degree of the track. External indices (using a labeled test set), such as the ARI, normalized mutual information, as well as internal indices (unlabeled test set), like the silhouette coefficient (SC) and v-measure, were used to evaluate the clustering performance, with ARI serving as the primary comparison metric.

To comprehensively compare the impact of introducing a high-dimensional latent space on model performance, the parameters, floating-point operations per second (FLOPs), and testing times of the SVAE and the SVAE-PSI were compared. Additionally, the clustering effects of incorporating identical fully connected layers in SAE, SDAE, and SCAE were compared. This comparison was aimed at validating that the improvements in clustering achieved by SVAE-PSI are not attributable to changes in the encoder structure settings.

4.2.3 | Comparison of FFF

The preliminary trials indicated that SVAE-PSI has significant advantages in dimensionality reduction and feature extraction. Therefore, the SVAE-PSI-1, with a latent space dimensionality of 50, was selected for the study.

TABLE 2 Comparison of different dimensionality reduction algorithms.

Algorithm type	Algorithm name	Characteristic
Traditional algorithms	PCA, KPCA, SPCA, ICA, UMAP, t-SNE	Less computationally intensive and suitable for linear relationships with simple datasets
Encoder-based algorithms	SAE, SDAE, SCAE, SVAE	Specializes in capturing complex non-linear relationships in high-dimensional data and can effectively work with structured data such as images and sequences
Improved encoder-based algorithms	SVAE-PSI	Improvements to the potential space reduce the loss of information in the dimensionality reduction process

Abbreviations: PCA, principal component analysis; SAE, stacked autoencoder; SCAE, stacked contractive autoencoder; SDAE, stacked denoising autoencoder; SVAE-PSI, stacked variational autoencoder with potential spatial improvement; t-SNE, t-distributed stochastic neighbor embedding.

TABLE 3 Different configurations of the autoencoder (AE) in this paper.

Name	Encoder structure	Activation function	Encoder out	Decoder out	Loss function
SAE-1	200-100-50-30-10-2	Relu	Linear	Linear	MSE
SAE -2	200-100-50-30-10-2	Relu	Tanh	Linear	MSE
SAE -3	200-500-200-30-20-2	Relu	Linear	Linear	MSE
SAE -4	200-500-200-30-20-2	Relu	Tanh	Linear	MSE
SDAE-1	200-100-50-30-10-2	Relu	Linear	Linear	MSE
SDAE-2	200-100-50-30-10-2	Relu	Tanh	Linear	MSE
SDAE-3	200-500-200-30-20-2	Relu	Linear	Linear	MSE
SDAE-4	200-500-200-30-20-2	Relu	Tanh	Linear	MSE
SCAE-1	200-100-50-30-10-2	Relu	Linear	Linear	MSE + CL
SCAE-2	200-100-50-30-10-2	Relu	Tanh	Linear	MSE + CL
SCAE-3	200-500-200-30-20-2	Relu	Linear	Linear	MSE + CL
SCAE-4	200-500-200-30-20-2	Relu	Tanh	Linear	MSE + CL
SVAE-1	200-100-50-30-10-2	Relu	Linear	Linear	MSE + KLD
SVAE-2	200-100-50-30-10-2	Relu	Tanh	Linear	MSE + KLD
SVAE-3	200-500-200-30-20-2	Relu	Linear	Linear	MSE + KLD
SVAE-4	200-500-200-30-20-2	Relu	Tanh	Linear	MSE + KLD
SVAE-PSI-1	200-100-50-30-10-2+latent space (10-100)	Relu	Linear	Linear	MSE + KLD
SVAE-PSI-2	200-100-50-30-10-2+latent space (10-100)	Relu	Tanh	Linear	MSE + KLD
SVAE-PSI-3	200-500-200-30-20-2+latent space (10-100)	Relu	Linear	Linear	MSE + KLD
SVAE-PSI-4	200-500-200-30-20-2+latent space (10-100)	Relu	Tanh	Linear	MSE + KLD

Abbreviations: CL, contrastive loss; KLD, Kullback-leibler divergence; MSE, mean squared error; Relu; rectified linear unit; SAE, stacked autoencoder; SCAE, stacked contractive autoencoder; SDAE, stacked denoising autoencoder; SVAE-PSI, stacked variational autoencoder with potential spatial improvement.

TABLE 4 Clustering results under different dimensions of low-dimensional space.

Dimensions	SVAE-1	SVAE-PSI-1
2	0.72	0.73
5	0.70	0.72
10	0.69	0.71
15	0.73	0.72
20	0.72	0.72

Abbreviations: SVAE, stacked variational autoencoder.

Additionally, fully connected layers with node counts of 50 and 100 were introduced to construct the FFF. Table 6 presents the outcomes of employing different training strategies. During the pre-training phase, two approaches were utilized: unsupervised greedy layer-wise pre-training for the encoder and proceeding without pre-training the encoder parameters. The fine-tuning phase explored the effects of either freezing or not freezing the pre-trained encoder parameters. The framework employs accuracy, precision, recall, and F1 score as evaluation metrics to

TABLE 5 Comparative settings of different clustering algorithms.

Strategy name	Clustering strategy-1	Clustering strategy-2	Clustering strategy-3	Clustering strategy-4	Clustering strategy-5
Dimensionality reduction	Traditional AE, improved AE				
Cluster method	KMeans	AHC	DBSCAN	GMM	FCM

Abbreviations: AHC, agglomerative hierarchical clustering; DBSCAN, density-based spatial clustering of applications with noise; GMM, Gaussian mixture model; FCM, fuzzy c-means.

TABLE 6 Comparative settings for different fine-tuning methods.

Strategy name	Pre-training	Fine-tuning
Fine-tuning strategy-1	Unsupervised	Supervised + freezing encoder
Fine-tuning strategy-2	Unsupervised	Supervised + unfreezing encoder
Fine-tuning strategy-3	None	Supervised

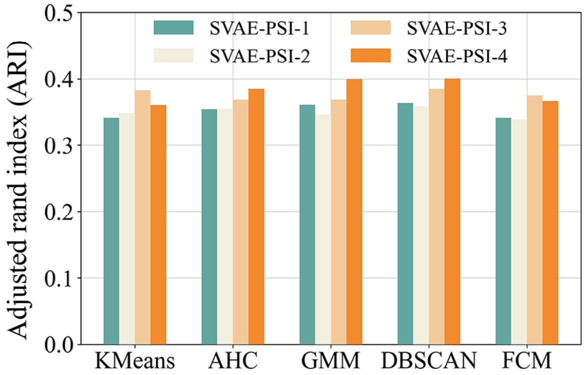


FIGURE 7 Adjusted rand index (ARI) under multiple track irregularity metrics. AHC, agglomerative hierarchical clustering; DBSCAN, density-based spatial clustering of applications with noise; FCM, fuzzy c-means; GMM, Gaussian mixture model; SVAE-PSI, stacked variational autoencoder with potential spatial improvement.

observe the performance when trained with only a small number of labeled samples.

5 | RESULTS AND DISCUSSION

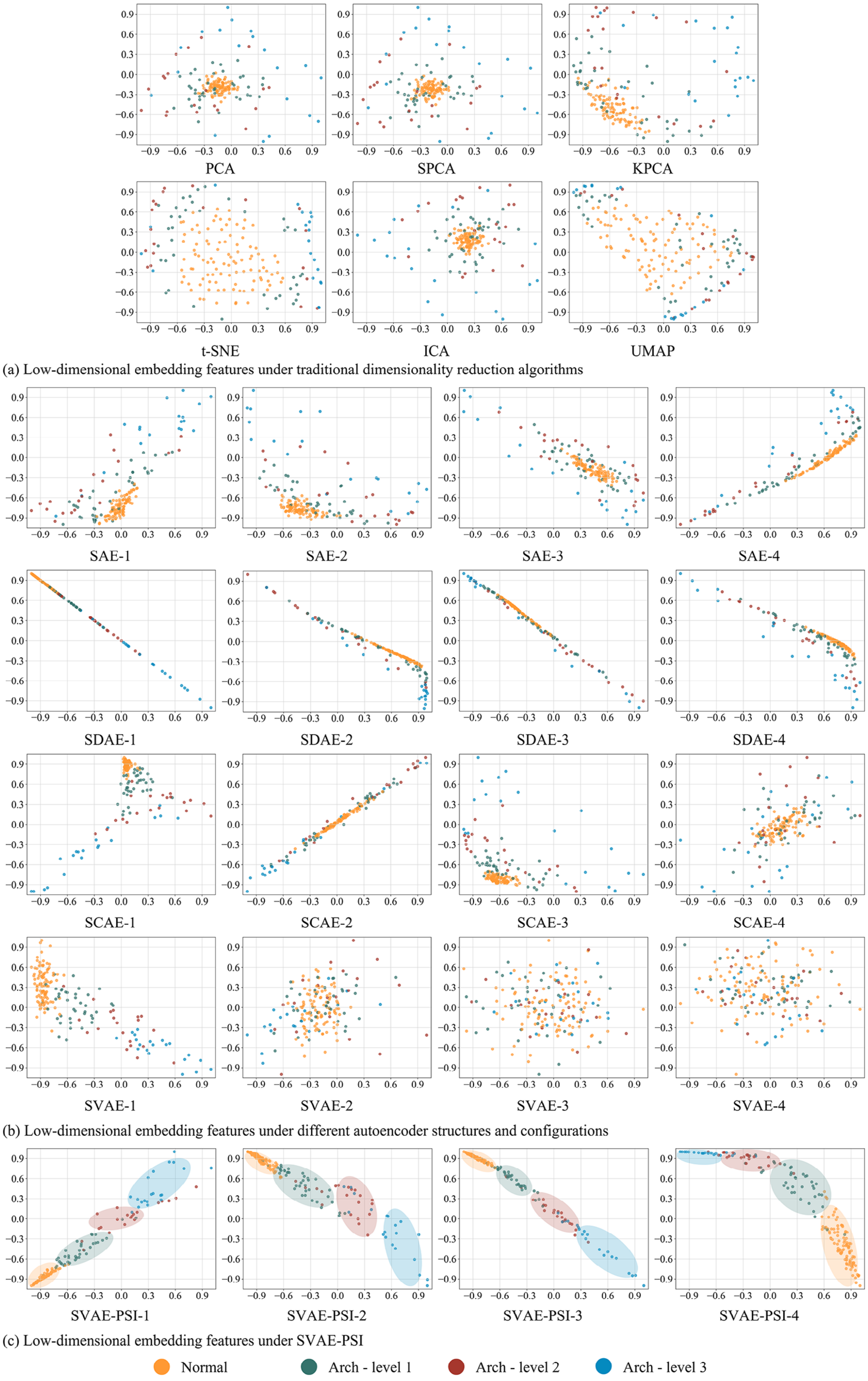
5.1 | Track irregularity type selection

As mentioned in the previous section, track irregularity includes multiple indicators. This study attempted to use various irregularity indicators, including gauge, alignment, longitudinal level, cross level and twist, as input for dimensionality reduction in the model. Figure 7 shows the ARI scores of the clustering after dimensionality reduction, indicating that the clustering performance is poor when multiple indicators are used as input together. Therefore, based on the findings of other studies, this research focuses on using longitudinal-level irregularity as the input indicators for subsequent studies.

5.2 | Visualization of dimensionality reduction results

The performance of dimensionality reduction algorithms is visualized by 2D scatterplots as shown in Figure 8. In the reduced-dimensional scatterplot, the x and y axes represent the projections of the original high-dimensional data in the new low-dimensional space. These two dimensions are generated by the algorithm with the aim of capturing and illustrating the main features and structure of the original data. Specifically, the x axis typically represents the first principal component or feature obtained during the dimensionality reduction process, which accounts for the largest variance in the data; whereas the y axis represents the second principal component or feature, which usually explains the largest portion of the remaining variance. The combination of these two dimensions allows us to intuitively observe the distribution of the data, clustering structures, and potential patterns.

Figure 8a illustrates the 2D scatterplot obtained based on traditional dimensionality reduction methods. From the figure, it can be observed that most of the dimensionality reduction methods fail to effectively distinguish between the normal category and different degrees of arch anomalies. In t-SNE and the uniform manifold approximation and projection (UMAP) algorithm based on flow learning, the scatter points show a uniform distribution, and the points representing normal are scattered in the central area; however, the distance between the points of different categories is the same, thus failing to form effective clusters. In contrast, the visualization results of PCA, SPCA and independent component analysis (ICA) are relatively consistent, showing a tendency of spreading from the central to the periphery, with the normal categories concentrated in the central of the scatterplot, but also intermixed with scatters from other categories. Kernel principal component analysis (KPCA), on the other hand, performs better, compared to the other methods, with the scatters of the normal category located in the lower left corner, which



(a) Low-dimensional embedding features under traditional dimensionality reduction algorithms

(b) Low-dimensional embedding features under different autoencoder structures and configurations

(c) Low-dimensional embedding features under SVAE-PSI

FIGURE 8 Visualization results of dimensionality reduction for different models. PCA, principal component analysis; SAE, stacked autoencoder; SCAE, stacked contractive autoencoder; SDAE, stacked denoising autoencoder; SVAE-PSI, stacked variational autoencoder with potential spatial improvement; t-SNE, t-distributed stochastic neighbor embedding.

is more clearly differentiated from the other categories, which is a more satisfactory result.

Figure 8b demonstrates the results of the dimensionality reduction based on the encoder model. From the figure, it can be observed that the performances of SAE and methods like SPCA are similar, both showing that the normal category scatter points are concentrated while the scatter points of other categories are dispersed. In contrast, the downscaling results of SCAE are relatively consistent, with all scatters distributed near the diagonal, suggesting that the addition of the CL term to the loss function causes the downscaled scatters to converge toward the diagonal. In the results of the SCAE-1, the normal category is concentrated in the upper left corner of the scatter plot and gradually concentrates toward the lower right corner as the degree of disease in the arch deepens, which is a satisfactory result. From the visualization results alone, the SDAE-1 showed the best results in terms of dimensionality reduction, with the scatters of the different classes largely unadulterated, which facilitated the subsequent clustering task. The SVAE, on the other hand, performed surprisingly well, and while SVAE-1 performed excellently, the remaining three models all performed extremely poorly as the model parameters were adjusted, with different categories of scatters confounded together. From the performance of the four AEs, it can be seen that each encoder performs better when the first parameter setting is used (nodes 200-100-50-30-10-2), but the performance decreases to varying degrees when the parameters are adjusted, especially for the SVAE. Therefore, it is hypothesized that for the dataset used in this study, increasing the number of hidden layer nodes is not an improved idea.

Figure 8c demonstrates the downscaling visualization results based on the SVAE-PSI for potential space dimension = 50, and it can be seen that the downscaling performances of the four model settings are satisfactory, and clusters with more distinct boundaries can be formed between different categories, which indicates that the improvement is effective.

5.3 | Analysis of clustering results

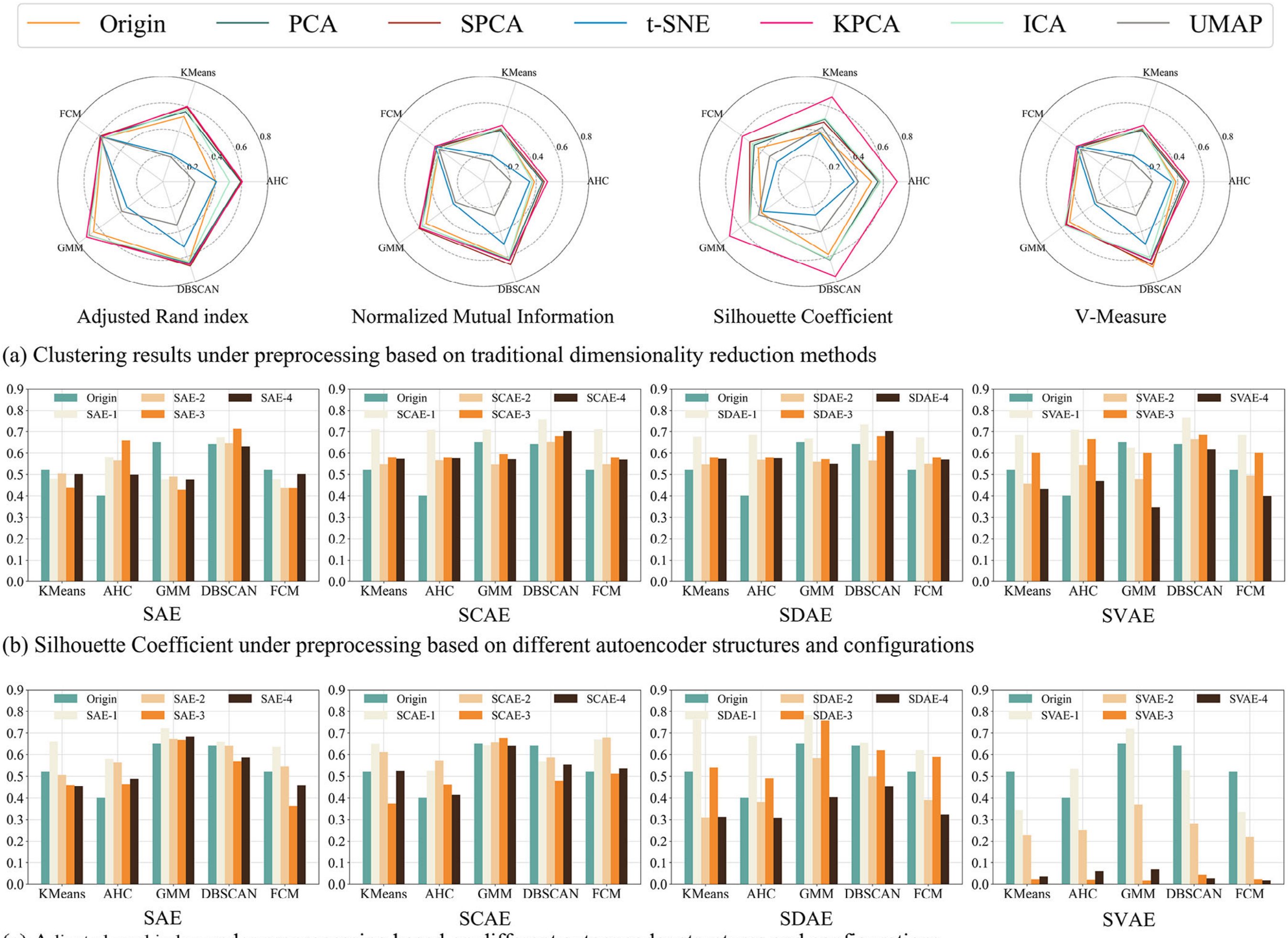
After achieving feature extraction by different dimensionality reduction algorithms, various clustering algorithms are accessed to evaluate the performance of FUF, and the detailed metrics are shown in Figures 9 and 10. The results in Figure 11 represent the average obtained through 10-fold cross-validation.

Figure 9a shows the results of clustering after preprocessing by traditional dimensionality reduction methods such as PCA and direct clustering of the original data. The figure shows that the performance of the original data when clustering directly is more general, which is only better than t-SNE and UMAP methods, which indicates that the dimensionality reduction operation of the data helps to improve the performance of clustering, and verifies that the idea of this study is correct. Among the traditional dimensionality reduction algorithms, KPCA is the best performer, which indicates that the introduction of kernel function for constructing complex nonlinear classifiers can effectively cope with the nonlinear data in this study. Among the clustering algorithms, GMM performs most satisfactorily with the best performance in several evaluation metrics, which indicates that the GMM method as a soft clustering method can better deal with the ambiguity and uncertainty in the dataset of this study.

Figure 9b shows the model performance under different clustering strategies using SC as the evaluation metric. It can be seen that all models have achieved good performance, indicating that these models have high consistency and good clustering quality. Figure 9c shows the ARI scores, and it can be seen that GMM achieved the best scores in all combinations. However, except for SVAE-1, the scores of other SVAE models have a sharp decline, indicating that the dimensionality reduction results are unstable when the latent space dimensionality is 2. This is speculated to be due to the over-regularization problem and the limitations of the prior distribution in traditional VAE (Dilokthanakul et al., 2017). This might lead to overly simplified learned latent representations, failing to sufficiently capture the complexity and diversity of the data, thus limiting the model's ability to capture multimodal data distributions.

Moreover, despite the good performance shown by the SC scores under different clustering strategies, some models perform poorly in ARI scores, reflecting that model performance may vary under different evaluation metrics. The performance of SVAE in ARI scores further indicates the need to consider latent space dimensionality and regularization strategies in model design to enhance model robustness and adaptability to complex data structures.

The SVAE is improved by adding 10–100 dimensions (in steps of 10) of latent space, and the results are shown in Figure 10. As can be seen from the figure, the clustering performance of the framework is significantly improved with the introduction of high dimensional latent space followed by dimensionality reduction, especially the ARI scores are in the range of 0.8–0.9 when using GMM for clustering, which is satisfactory. Table 7 gives the model complexity of different SVAEs when the potential space dimension 50/100 is introduced. The comparison of the number of parameters, FLOPs, and test time shows that the introduction of the potential space only slightly increases the complexity of the model and has almost no effect on its actual operation and deployment.



(a) Clustering results under preprocessing based on traditional dimensionality reduction methods

(b) Silhouette Coefficient under preprocessing based on different autoencoder structures and configurations

(c) Adjusted rand index under preprocessing based on different autoencoder structures and configurations

FIGURE 9 Clustering results under preprocessing based on different dimensionality reduction methods. AHC, agglomerative hierarchical clustering; DBSCAN, density-based spatial clustering of applications with noise; FCM, fuzzy c-means; GMM, Gaussian mixture model; PCA, principal component analysis; SAE, stacked autoencoder; SCAE, stacked contractive autoencoder; SDAE, stacked denoising autoencoder; SVAE-PSI, stacked variational autoencoder with potential spatial improvement; t-SNE, t-distributed stochastic neighbor embedding.

TABLE 7 The parameter quantity, floating-point operations per second (FLOPs) and test time before and after AE improvement.

	Params			FLOPs			Testing time		
Name	**Initial**	**Dim = 50**	**Dim = 100**	**Initial**	**Dim = 50**	**Dim = 100**	**Initial**	**Dim = 50**	**Dim = 100**
SVAE-1	74,494	75,252	76,052	73.7 K	74.3 K	75.0 K	5.3 ms	5.2 ms	5.6 ms
SVAE-2	74,494	75,252	76,052	73.7 K	74.3 K	75.0 K	6.2 ms	5.1 ms	5.9 ms
SVAE-3	435,274	436,492	437,792	433.3 K	434.4 K	435.6 K	6.6 ms	6.5 ms	6.1 ms
SVAE-4	435,274	436,492	437,792	433.3 K	434.4 K	435.6 K	6.7 ms	6.6 ms	6.0 ms

Abbreviation: SVAE, stacked variational autoencoder.

Figure 11 shows the clustering performance after adding fully connected layers of dimensions 20/50/80 to the other AE models. As seen in the figure, the introduction of these fully connected layers in these models did not improve the ARI score, indicating that the change in SVAE is due to the change in the latent space and not a tweak in the structural settings.

5.4 | Analysis of classification results

The results in the previous section show that the SVAE-PSI has advantages in dimensionality reduction and can achieve superior results in a fully unsupervised clustering framework, so in this section, the fine-tuning results are computed with the SVAE-PSI-1, with latent

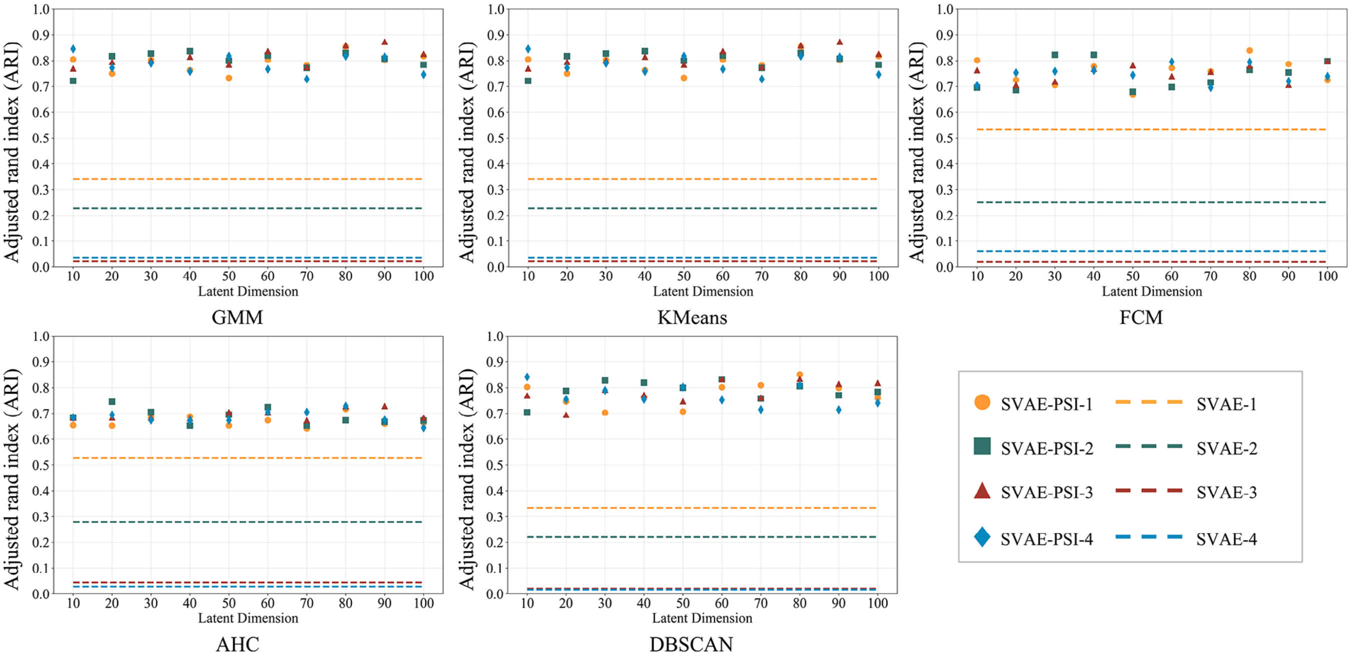


FIGURE 10 Comparison of clustering results after AE improvement. AHC, agglomerative hierarchical clustering; DBSCAN, density-based spatial clustering of applications with noise; FCM, fuzzy c-means; GMM, Gaussian mixture model; SVAE-PSI, stacked variational autoencoder with potential spatial improvement.

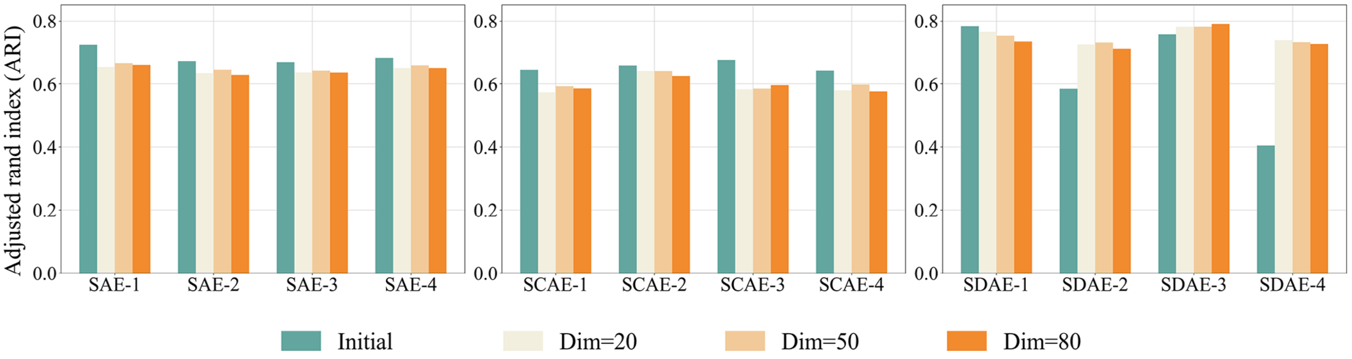
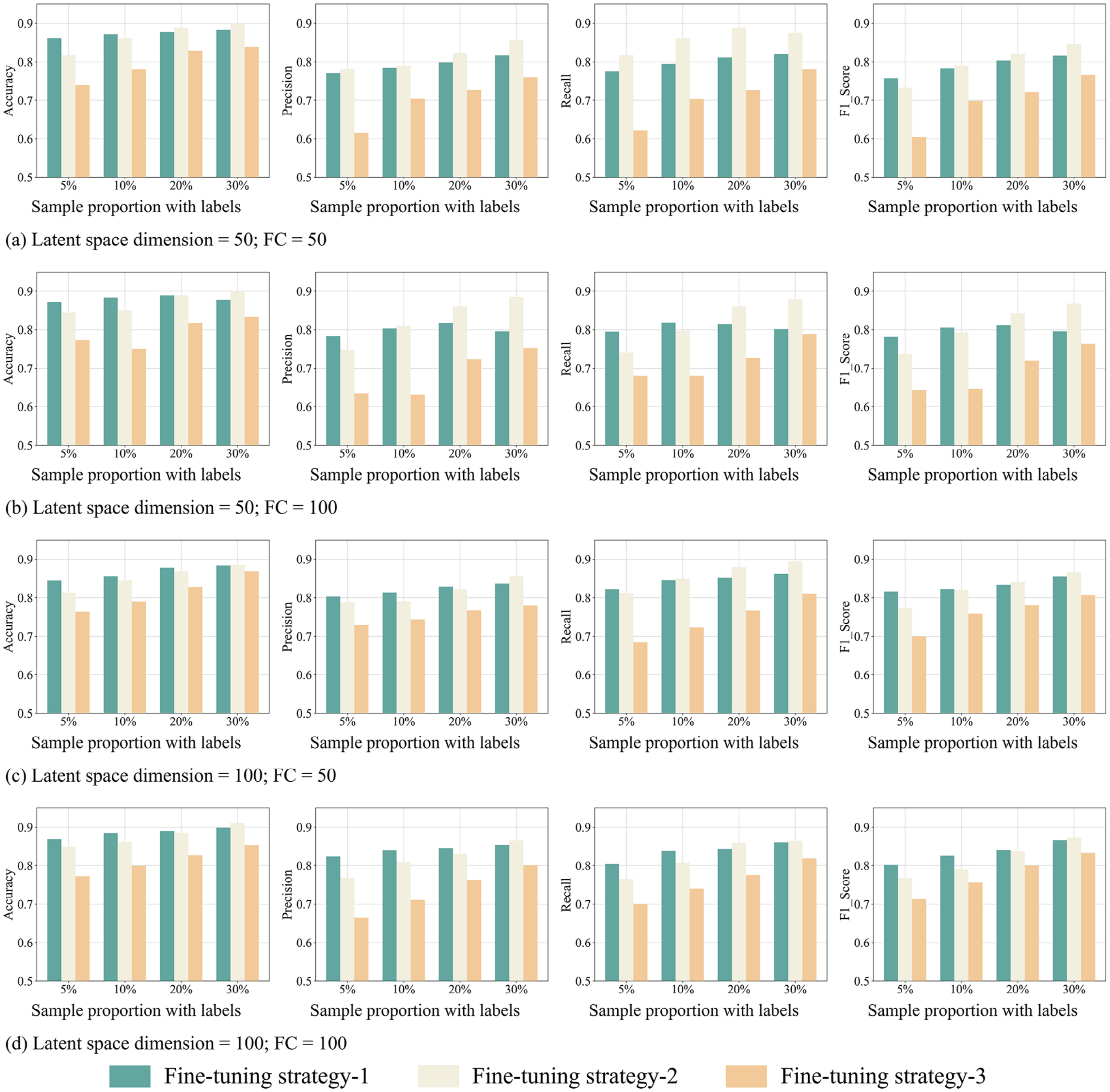


FIGURE 11 Clustering results of other AEs after structural adjustment (with Gaussian mixture model [GMM] as the clustering algorithm). SAE, stacked autoencoder; SCAE, stacked contractive autoencoder; SDAE, stacked denoising autoencoder.

space = 50/100, and with the incorporation of fully connected (FC) = 50/100.

Figure 12 illustrates the performance after fine-tuning with different labeled training sets. As can be seen from the figure, Fine-tuning strategy-1 achieves better performance when using a very small number of labeled training sets (only 5% of the total number of samples), while Fine-tuning strategy-2 gradually overtakes it as the number of samples increases. This suggests that better feature extraction can be achieved by retaining the original encoder parameters when using a very small number of samples and that unfreezing the parameters at this point may cause overfitting due to too little data. As the number of samples increases, the risk of model overfitting is greatly reduced, so Fine-tuning strategy-2 gradually outperforms Fine-tuning strategy-1. The performance of Fine-tuning strategy-3 is lower than that of other strategies, which indicates that pre-training of the encoder is necessary.

Additionally, to explore the impact of using pre-trained encoders as feature extraction layers on the performance of fully connected networks, the number of labeled samples was gradually increased during the training process of the FFF framework. The results are shown in Figure 13. In the figure, the dashed lines represent the results for the FFF framework with FC = 100, data_scale = 5% and FC = 50, data_scale = 5%. The specific procedure is as follows: first, the SVAE-PSI-1 encoder structure was pre-trained using unlabeled training set samples to extract data features and perform dimensionality reduction. Then, the pre-trained encoder was connected to fully connected layers (with 50 and 100 hidden node), and the encoder network parameters were frozen. Next, the FFF framework was trained



(a) Latent space dimension = 50; FC = 50

(b) Latent space dimension = 50; FC = 100

(c) Latent space dimension = 100; FC = 50

(d) Latent space dimension = 100; FC = 100

FIGURE 12 Classification results of few-shot fine-tuned framework with different numbers of hidden nodes and proportion of labeled samples.

using 5% of the labeled training set samples, and the results were obtained from the test set. Figure 13 shows that without pre-training the encoder structure, 50% of the labeled samples are required to achieve the same effect as using 5% of the labeled samples with pre-training.

To ensure line safety in practical engineering, railway engineers also conduct regular inspections to check the track structure status. To estimate the workload of this study dataset only, each piece of data is about 12.5 m, so 45% (810 labels) of the labels need to be checked by the staff inspecting a railway section of about 10 km, which is an extremely large amount of workload and can be carried out only during the daytime window period, so the sample-less fine-tuning framework of this paper is of significant engineering significance.

5.5 | Generalization performance validation

To verify the generalization performance of the proposed framework, we collected track inspection data from another HSR line. The specific data collection process

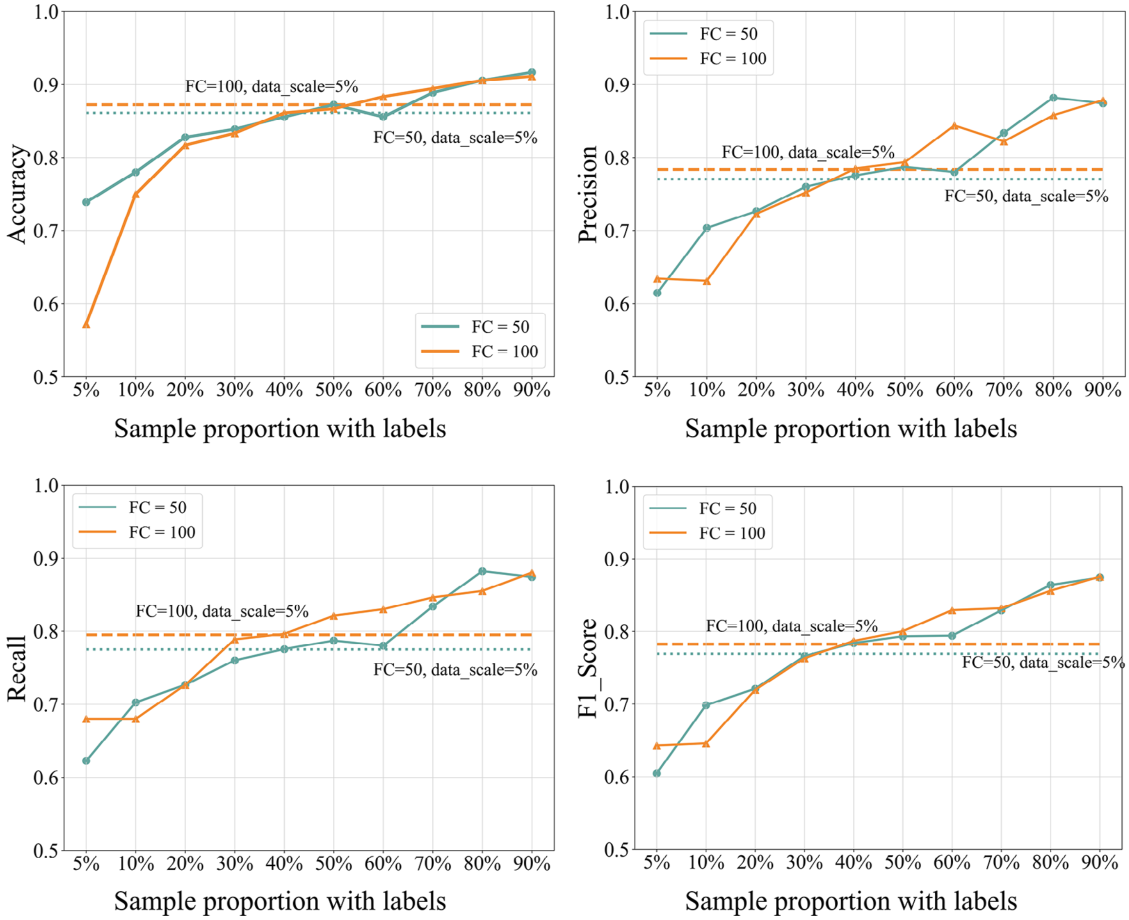


FIGURE 13 Comparison of classification results for pre-trained and un-pretrained encoder parameters (where the dashed and dotted lines line is the classification result when freezing the encoder parameters, latent space dimension = 50).

TABLE 8 Sample composition of the dataset.

Name	Training set	Test set	All
Normal	540	60	600
Arch-level 1	180	20	200
Arch-level 2	90	10	100
Arch-level 3	90	10	100
All	900	100	1000

is detailed in Section 4.1, and the data composition is shown in Table 8. Figure 14 presents the clustering results obtained using the FUF framework.

As can be seen from the figure, applying the SVAE-PSI series model for dimensionality reduction before clustering significantly improved the results, compared to direct clustering, achieving an ARI score of approximately 0.8. This indicates that the proposed framework performs well on other operational lines and can accurately diagnose arching defects.

In addition, regarding the generalization to other types of track defects, although this study primarily focuses on the identification of arch defects in ballastless tracks, the proposed framework demonstrates broad applicability due to the similarities among various track diseases and faults encountered in HSR systems. With appropriate adjustments to the model's input and output structures, it is anticipated that this framework can be extended to identify other types of track defects in the future. This direction will be an important focus of our subsequent research.

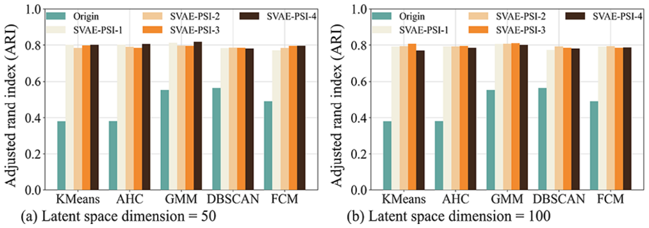


FIGURE 14 ARI under preprocessing based on different AE structures and configurations. AHC, agglomerative hierarchical clustering; DBSCAN, density-based spatial clustering of applications with noise; FCM, fuzzy c-means; SVAE-PSI, stacked variational autoencoder with potential spatial improvement.

6 | CONCLUSION

This study introduces two frameworks for identifying track slab arch, FUF and FFF, which are based on unsupervised algorithms. These frameworks provide railway maintenance personnel with the flexibility to choose based on actual conditions, thereby better meeting the maintenance and operational needs of railway engineering sites. The results indicate that both frameworks can enhance the diagnostic effectiveness of track slab arch defects, which holds significant practical engineering implications. Specific details are as follows:

1. The structure of the latent space in the VAE has been modified to create SVAE-PSI, enhancing the encoder's performance in dimensionality reduction tasks.
2. The FUF framework is designed in combination with the clustering algorithm. It incorporates the feature extraction benefits of the dimensionality reduction algorithm along with the superior clustering performance of methods such as GMM, ensuring the effective clustering of various data categories even in the complete absence of labeled training samples.
3. The FFF framework is designed in combination with the idea of model fine-tuning. Results show that this framework can achieve significant performance improvements with very few labeled samples. It can significantly reduce the demand for labeled samples. In practical railway applications, it greatly reduces the workload of field personnel.

It should be noted that this study has some limitations. First, there remains significant potential to enhance the performance of the proposed model. Strategies such as clustering integration, data augmentation techniques, and hyperparameter optimization can be employed to improve the model's effectiveness in clustering and classification tasks. To further enhance the generalizability of the proposed frameworks, it is essential to incorporate data from a broader range of routes and to conduct a thorough evaluation of model performance in handling large-scale on-site data. Specifically, implementing strategies such as dynamic adjustments to model complexity, along with batch and stream processing, will be vital for achieving an optimal balance between model complexity and performance. Additionally, exploring advanced methodologies such as the neural dynamic classification algorithm (Rafiei & Adeli, 2017), dynamic integrated learning algorithm (Alam et al., 2020), finite element machine for fast learning (Pereira et al., 2020), and self-supervised learning (Rafiei et al., 2023) represents promising avenues for the future development.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China No. 52178405, the Fundamental Research Funds for the Central Universities No. 2022JBQY009.

REFERENCES

Adeli, H., & Kamal, O. (1989). Parallel structural analysis using threads. *Computer-Aided Civil and Infrastructure Engineering*, *4*(2), 133–147.

Adeli, H., & Wu, M. (1998). Regularization neural network for construction cost. *Journal of Construction Engineering and Management*, *124*(1), 18–24.

Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering*, *16*(2), 126–142.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications*, *32*(10), 8675–8690.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems 19 (NIPS'06)*, British Columbia, MA (pp. 153–160).

Braga, J. A. P., & Andrade, A. R. (2021). Multivariate statistical aggregation and dimensionality reduction techniques to improve monitoring and maintenance in railways: The wheelset component. *Reliability Engineering and System Safety*, *216*, 107932.

Cai, X., Tang, X., Pan, S., Wang, Y., Yan, H., Ren, Y., Chen, N., & Hou, Y. (2023). Intelligent recognition of defects in high-speed railway slab track with limited dataset. *Computer-Aided Civil and Infrastructure Engineering*, *39*(6), 911–928. https://doi.org/10.1111/mice.13109

Cai, X., Tang, X., Yang, F., Wang, T., & Sun, J. (2023). Estimation of turnout irregularities using vehicle responses with improved Bi-LSTM and Gaussian process regression. *Measurement*, *221*, 113513.

Chang, Y., Liu, R., & Tang, Y. (2023). Segment condition-based railway track maintenance schedule optimization. *Computer-Aided Civil and Infrastructure Engineering*, *38*(2), 160–193.

Chen, J., Li, Q., Zhang, S., Lin, C., & Wei, S. (2024). Convolutional autoencoder-based damage detection for urban railway tracks using an ultra-weak FBG array monitoring system. *IEEE Sensors Journal*, Advance online publication. https://doi.org/10.1109/JSEN.2024.3411652

Chen, Z., Cai, X., Wang, T., Wang, M., Liu, W., & Zhang, Q. (2024). Mapping the relationship between the temperature gradient of CRTS III slab track on bridge and rail deformation in high-speed railway. *Structures*, *59*, 105777.

Cong, J., Yan, X., Chen, R., Gao, M., An, B., Tang, H., Wang, Y., & Wang, P. (2023). Profile evaluation of rail joint in a 3 m wavelength based on unsupervised learning. *Computer-Aided Civil and Infrastructure Engineering*, *38*(13), 1834–1856.

Cui, J., Xiao, J., Hou, Y., Wu, X., Zhou, J., Peng, X., & Wang, Y. (2023). Unsupervised domain adaptive dose prediction via cross-attention transformer and target-specific knowledge preservation. *International Journal of Neural Systems*, *33*(11), 2350057.

Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., Lee, M. C. H., Salimbeni, H., Arulkumaran, K., & Shanahan, M. (2017). *Deep unsupervised clustering with Gaussian mixture variational* *autoencoders*. arXiv: 1611.02648. https://doi.org/10.48550/arXiv.1611.02648

Fan, G., Zhang, H., Zhu, W., Zhang, H., & Chai, X. (2019). Numerical and experimental research on identifying a delamination in ballastless slab track. *Materials*, *12*(11), 1788.

Fernandez-Rodrıguez, J. D., Palomo, E. J., Ortiz-de-Lazcano-Lobato, J. M., Ramos-Jimenez, G., & Lopez-Rubio, E. (2023). Dynamic learning rates for continual unsupervised learning. *Integrated Computer-Aided Engineering*, *30*(3), 257–273.

Giglioni, V., Venanzi, I., Poggioni, V., Milani, A., & Ubertini, F. (2023). Autoencoders for unsupervised real-time bridge health assessment. *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 959–974.

Hassanpour, A., Moradikia, M., Adeli, H., Khayami, S. R., & Shamsinejad, P. (2019). A novel end-to-end deep learning scheme for classifying multiclass motor imagery EEG signals. *Expert Systems*, *36*(6), e12494.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural network. *Science*, *313*(5786), 504–507.

Hinton, G. E., Osinder, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14*(1), 1303–1347.

Hong, J. (2020). Automatic assessment method of ballastless track slab-expanding defect section. *Railway Standard Design*, *64*(04), 12–17.

Jahan, K., Umesh, J. P., & Roth, M. (2021). Anomaly detection on the rail lines using semantic segmentation and self-supervised learning. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Orlando, FL (pp. 1–7).

Jiang, K., Han, Q., Du, X., & Ni, P. (2021). A decentralized unsupervised structural condition diagnosis approach using deep autoencoders. *Computer-Aided Civil and Infrastructure Engineering*, *36*(6), 711–732.

Kakarla, S., Gangula, P., Sai Charan Singh, C., & Hitendra Sarma, T. (2020). Dimensionality reduction in hyperspectral images using auto-encoders. *Advances in Computational Intelligence and Informatics*, *119*, 101–107.

Kingma, D. P., & Welling, M. (2014). *Auto-encoding variational Bayes*. arXiv:1312.6114. https://doi.org/10.48550/arXiv.1312.6114

Li, C., Wang, P., Gao, T., Wang, J., Yang, C., Liu, H., & He, Q. (2020). Spatial–temporal model to identify the deformation of underlying high-speed railway infrastructure. *Journal of Transportation Engineering, Part A: Systems*, *146*(8), 4020084.

Li, C., Li, L., Wang, J., Feng, X., Wang, Q., Huang, C., Wang, Y., & He, Q. (2022). Deformation recognition and prediction of track slabs based on track inspection data. *Journal of Southwest Jiaotong University*, *57*(2), 306–313.

Li, S., Cao, Y., Gdoutos, E. E., Tao, M., Faisal Alkayem, N., Avci, O., & Cao, M. (2024). Intelligent framework for unsupervised damage detection in bridges using deep convolutional autoencoder with wavelet transmissibility pattern spectra. *Mechanical Systems and Signal Processing*, *220*, 111653.

Li, W., Zhang, M., Shen, Z., Hu, W., & Li, P. (2018). Track crack detection method in complex environment. *Proceedings of 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China (pp. 356–359).

Li, Y., Chen, J., Wang, J., Shi, X., & Chen, L. (2020). Analysis of damage of joints in CRTSII slab track under temperature and vehicle loads. *KSCE Journal of Civil Engineering*, *24*(4), 1209–1218.

Li, Z., Liu, X., & He, Y. (2019). Identification of temperature-induced deformation for HSR slab track using track geometry measurement data. *Sensors*, *19*(24), 5446.

Li, Z., Liu, X., Lu, H., He, Y., & Zhou, Y. (2020). Surface crack detection in precasted slab track in high-speed rail via infrared thermography. *Materials*, *13*(21), 4837.

Liao, H., Zhu, Q., Zan, Y., Xie, Y., & Sun, J. (2016). Detection of ballastless track diseases in high-speed railway based on ground penetrating radar. *Journal of Southwest Jiaotong University*, *51*(1), 8–13.

Liu, X., Zhang, W., Xiao, J., Liu, X., & Li, W. (2019). Damage mechanism of broad-narrow joint of CRTSII slab track under temperature rise. *KSCE Journal of Civil Engineering*, *23*(5), 2126–2135.

Martins, G. B., Papa, J. P., & Adeli, H. (2020). Deep learning techniques for recommender systems based on collaborative filtering, *Expert Systems*, *37*(6), e12647.

Midwinter, M., Al-Sabbag, Z. A., & Yeum, C. M. (2023). Unsupervised defect segmentation with pose priors. *Computer-Aided Civil and Infrastructure Engineering*, *38*(17), 2455–2471.

Pan, S., Yan, H., Liu, Z., Chen, N., Miao, Y., & Hou, Y. (2023). Automatic pavement texture recognition using lightweight few-shot learning. *Philosophical Transactions of the Royal Society A*, *381*(2254), 20220166.

Peng, H., Zhang, Y., Wang, J., Liu, Y., & Gao, L. (2019). Interfacial bonding strength between cement asphalt mortar and concrete in slab track. *Journal of Materials in Civil Engineering*, *31*(7), 1–11.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Pu, H., Wan, X., Song, T., Schonfeld, P., Li, W., & Hu, J. (2023). A geographic information model for 3-D environmental suitability analysis in railway alignment optimization. *Integrated Computer-Aided Engineering*, *30*(1), 67–88.

Qin, H., Liu, J., Liu, W., Yang, F., & Sun, J. (2024). Identification method for minor changes in track geometry in turnout areas based on location index and confidence coefficient. *Journal of the China Railway Society*, *46*(5), 110–115.

Rafiei, M. H., & Adeli, H. (2017). NEEWS: A novel earthquake early warning system using neural dynamic classification and neural dynamic optimization model. *Soil Dynamics and Earthquake Engineering*, *100*, 417–427.

Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Construction Engineering and Management*, *142*(2), 04015066.

Rafiei, M. H., Gauthier, L., Adeli, H., & Takabi, D. (2023). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 1457–1471.

Ren, J., Li, X., Yang, R., Wang, P., & Xie, P. (2016). Criteria for repairing damages of CA mortar for prefabricated framework-type slab track. *Construction and Building Materials*, *110*, 300–311.

Ren, J., Wang, J., Li, X., Wei, K., Li, H., & Deng, S. (2020). Influence of cement asphalt mortar debonding on the damage distribution and mechanical responses of CRTS I prefabricated slab. *Construction and Building Materials*, *230*, 116995.

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11). Omnipress, Madison, WI, USA (pp. 830–844).

Rodriguez-Lozano, F. J., Gámez-Granados, J. C., Palomares, J. M., & Olivares, J. (2023). Efficient data dimensionality reduction method for improving road crack classification algorithms. *Computer-Aided Civil and Infrastructure Engineering*, *38*(16), 2339–2354.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Sajedi, S., & Liang, X. (2022). Deep generative Bayesian optimization for sensor placement in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1109–1127.

Shao, P., Li, H., Wu, S., Wang, T., Liu, Z., & Li, H. (2013). Measurement and research on temperature warping of CRTS I track slab and crack between track slab and cement asphalt mortar cushion. *China Railway Science*, *34*(2), 5.

Song, T., Pu, H., Schonfeld, P., Zhang, H., Li, W., & Hu, J. (2022). Simultaneous optimization of 3-D alignments and station locations for dedicated high-speed railways. *Computer-Aided Civil and Infrastructure Engineering*, *37*(4), 405–426.

Soleimani-Babakamali, M. H., Sepasdar, R., Nasrollahzadeh, K., Lourentzou, I., & Sarlo, R. (2022). Toward a general unsupervised novelty detection framework in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1128–1145.

Ngamkhanong, C., Kaewunruen, S., & Costa, B. J. A. (2018). State-of-the-art review of railway track resilience monitoring. *Infrastructures*, *3*(1), 3.

Nie, L., Zhang, R., Hu, T., Tang, Z., Fang, M., Lv, X., & Zhang, R. (2023). A fitting algorithm based on multi-touch gesture for rapid generation of railway line. *Integrated Computer-Aided Engineering*, *30*(2), 135–150.

Tang, X., Chen, rZ., Cai, X., & Wang, Y. (2023). Ballastless track arching recognition based on one-dimensional residual convolutional neural network and vehicle response. *Construction and Building Materials*, *408*(8), 133624.

Tian, X., Gao, L., Yang, F., Sun, X., Wei, H., & Sun, J. (2020). Management standard for cyclic irregularity of ballastless track slab based on dynamic short chord. *China Railway Science*, *41*(6), 30–38.

Tong, L., Jia, L., Geng, Y., Liu, K., Qin, Y., & Wang, Z. (2023). Anchor-adaptive railway track detection from unmanned aerial images. *Computer-Aided Civil and Infrastructure Engineering*, *38*(18), 2666–2684.

Wei, H., Yang, F., Wu, S., Zhu, L., Yao, H., & Ren, X. (2022). Historical data matching method of static track geometry measurement and its performance evaluation. *Journal of Railway Engineering Society*, *39*(9), 19–25.

Wei, H., Yang, F., Zhu, H., Zhang, M., & Yin, H. (2022). A matching method for dynamic and static inspection data of track based on dynamic time warping. *Journal of Railway Science and Engineering*, *19*(1), 78–86.

Wei, S., Liu, L., Zhao, Y., Li, Y., & Wang, H. (2011). GJ-6 Track Inspection System. *Railway Engineering*, *11*, 98–101.

Xu, H., Ping, W., & Xu, J. (2013). Analysis of cement emulsified asphalt mortar gap on CRTSII slab track under train loading. *Applied Mechanics and Materials*, *405–408*, 1834–1838.

Xu, J., Wang, P., An, B., Ma, X., & Chen, R. (2018). Damage detection of ballastless railway tracks by the impact-echo method. *Proceedings of the Institution of civil Engineers-Transport*, *171*(2), 106–114.

Yang, F., Wang, X., You, M., Yang, J., & Shi, S. (2020). Research on the state evaluation method of CRTS II track slab based on track irregularity. *Journal of Railway Engineering Society*, *37*(7), 29–34.

Yang, F., Zhao, G., You, M., Liao, X., Ji, H., & Li, D. (2021). Research on identification and early warning model for track slab arch based on variation characteristics of longitudinal irregularity. *Journal of the China Railway Society*, *43*(8), 106–116.

Yu, C., Xiang, J., Mao, J., Gong, K., & He, S. (2018). Influence of slab arch imperfection of double-block ballastless track system on vibration response of high-speed train. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *40*(2), 109.

Zhang, Y., Gao, L., Zhong, Y., Jiang, H., & Huang, Y. (2024). Study on arching pattern of continuous ballastless slab track induced by oblique crack in concrete base under high temperature in summer. *Journal of the China Railway Society*, *46*(3), 165–175.

Zhang, Y., O'Connor, S. M., van der Linden, G. W., Prakash, A., & Lynch, J. P. (2016). SenStore: A scalable cyber infrastructure platform for implementation of data-to-decision frameworks for infrastructure health management. *Journal of Computing in Civil Engineering*, *30*(5), 04016012.

Zhao, Z., Zhang, Q., Yu, X., Sun, C., Wang, S., Yan, R., & Chen, X. (2021). Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–28.

How to cite this article: Tang, X., Wang, Y., Cai, X., Yang, F., & Hou, Y. (2024). Diagnosis of high-speed railway ballastless track arching based on unsupervised learning framework. *Computer-Aided Civil and Infrastructure Engineering*, 1–21. https://doi.org/10.1111/mice.13342