SCIENCE CHINA
Technological Sciences

•Article•

February 2023 Vol.66 No.2: 512–527
https://doi.org/10.1007/s11431-022-2218-9

A novel LSTM-autoencoder and enhanced transformer-based detection method for shield machine cutterhead clogging

QIN ChengJin[*], WU RuiHong, HUANG GuoQiang, TAO JianFeng & LIU ChengLiang

State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Received August 1, 2022; accepted September 22, 2022; published online January 18, 2023

Shield tunneling machines are paramount underground engineering equipment and play a key role in tunnel construction. During the shield construction process, the "mud cake" formed by the difficult-to-remove clay attached to the cutterhead severely affects the shield construction efficiency and is harmful to the healthy operation of a shield tunneling machine. In this study, we propose an enhanced transformer-based detection model for detecting the cutterhead clogging status of shield tunneling machines. First, the working state data of shield machines are selected from historical excavation data, and a long short-term memory-autoencoder neural network module is constructed to remove outliers. Next, variational mode decomposition and wavelet transform are employed to denoise the data. After the preprocessing, nonoverlapping rectangular windows are used to intercept the working state data to obtain the time slices used for analysis, and several time-domain features of these periods are extracted. Owing to the data imbalance in the original dataset, the *k*-means-synthetic minority oversampling technique algorithm is adopted to oversample the extracted time-domain features of the clogging data in the training set to balance the dataset and improve the model performance. Finally, an enhanced transformer-based neural network is constructed to extract essential implicit features and detect cutterhead clogging status. Data collected from actual tunnel construction projects are used to verify the proposed model. The results show that the proposed model achieves accurate detection of shield machine cutterhead clogging status, with 98.85% accuracy and a 0.9786 *F*1 score. Moreover, the proposed model significantly outperforms the comparison models.

shield tunneling machine, cutterhead clogging, fault diagnosis, autoencoder, multihead self-attention mechanism, imbalanced data

Citation: Qin C J, Wu R H, Huang G Q, et al. A novel LSTM-autoencoder and enhanced transformer-based detection method for shield machine cutterhead clogging. Sci China Tech Sci, 2023, 66: 512–527, https://doi.org/10.1007/s11431-022-2218-9

1 Introduction

Shield machines are major equipment urgently needed in the basic construction of subways, highways, railroads, water conservancies, and national defense facilities [1]. Shield machines have the advantages of fast excavation speed, high construction quality, and little disturbance to the surrounding environment and have been widely used in tunnel construction projects [2–5]. As shown in Figure 1, during a construction process, clay may get attached to the cutterhead of the shield tunneling machine, forming the so-called "mud cake" and causing the cutterhead to be clogged. Once the cutterhead is clogged, the load exerted on it will increase, making the excavation parameters change drastically and causing the construction efficiency to decrease [6].

There are four reasons for cutterhead clogging: adhesion of soil particles to the cutterhead surface, cohesion between clay particles, bridging effect of clay clusters in the opening of the cutterhead, and some soil particles' tendency to dissolve in water [7–10]. Based on scaled experiments, Fang et al. [10,11] suggested that the cutterhead clogging process can be divided into three stages. The first stage is called the

[*] Corresponding author (email: qinchengjin@sjtu.edu.cn)

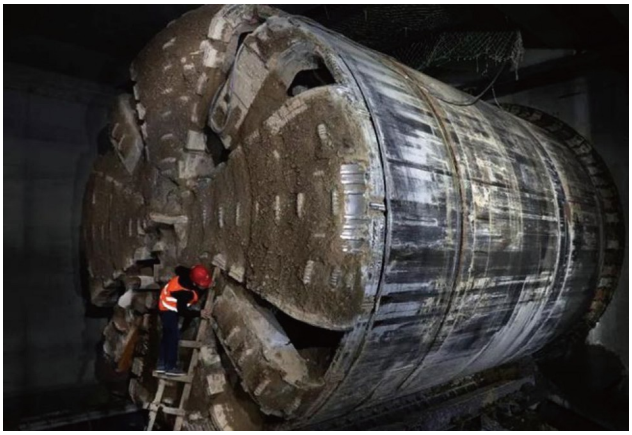

Figure 1 (Color online) A clogged cutterhead after excavation.

adhesion stage, during which clay clusters begin to adhere to the cutter, causing the excavation speed to decrease sharply. The second stage is called the adhesion development stage, during which the clay clusters adhered to the cutter in the first stage become larger and start to cover the cutter, making the excavation speed further decrease and the cutterhead torque slowly increase. The third stage is called the clogging stage, during which the clay adhesion begins to join together and extend from the cutterhead center to the edge, resulting in sharper excavation speed fluctuation and larger cutterhead torque. This theory has uncovered the relationship between cutterhead clogging and the excavation speed and cutterhead torque of shield tunneling machines, which provides some criteria for recognizing potential cutterhead clogging cases based on several excavation parameters of a shield tunneling machine. In early studies, real construction data also suggest that when the cutterhead is clogged, the cutterhead torque and total thrust tend to increase, whereas the excavation speed tends to decrease, and the earth pressure becomes unstable [6,12–14].

Traditionally, clogging risk is estimated by the geological environment of a shield tunneling machine. Hollmann et al. [15,16] studied the relationship between clogging potential and several geological environment characteristics, such as the liquid limit, plastic limit, and water content of the soil, and developed several diagrams to evaluate the clogging potential. Kang et al. [17] built an apparatus to assess clogging potential and drew similar results to those diagrams. Oliveira et al. [18] performed several experiments and further amended those diagrams based on the experiment results. However, because geological information cannot be acquired in real time, these methods may not detect cutterhead clogging in time; thus, measurements to remove clay clusters on the cutterhead may not be taken timely, and the tunneling efficiency may decrease. Fortunately, data-driven clogging potential detection methods provide a feasible alternative solution. These methods exploit the excavation data of a shield tunneling machine and build diagnostic models based on the data to detect clogging potential. With the development of hardware and data processing algorithms, data-driven methods are widely used in various engineering fields [19–21]. For cutterhead clogging, Zhou et al. [22–24] suggested that thrust and torque are the main loads acting on shielding machines and consume the most energy. Zhai et al. [25] used parameters such as penetration rate, thrust, specific energy, rotation speed, and torque of a shield tunneling machine to build a random forest (RF)-based model. Their work described another method to detect potential cutterhead clogging. However, as deep learning-based methods are frequently used in mechanical fault diagnosis fields, their diagnostic models are becoming deeper and achieving higher performances. The performance of shallow models such as RF may be less accurate than deep learning-based methods.

To improve the detection performance, we propose a novel long short-term memory (LSTM)-autoencoder and enhanced transformer-based model for cutterhead clogging status detection. First, the working state data of shield machines are collected from historical excavation data, and an LSTM-autoencoder neural network (NN) module is designed to remove outliers. Then, variational mode decomposition (VMD) and wavelet transform (WT) are employed to denoise the data after the removal of outliers. After preprocessing, nonoverlapping rectangular windows are used to intercept the working state data to obtain the time slices used for analysis, and several time-domain features of these periods are extracted. Considering that the original data set is imbalanced, we employ the k-means-synthetic minority oversampling technique (SMOTE) algorithm to oversample the extracted time-domain features of the clogging data in the training set, which balances the dataset and improves the model performance. Finally, we construct an enhanced transformer-based NN for extracting essential implicit features and detecting cutterhead clogging. The data collected from actual tunnel construction projects are used to verify the proposed model. The results show that the proposed model achieves accurate detection of shield machine cutterhead clogging status and significantly outperforms existing models.

The major contributions of this study are as follows.

(1) We propose a novel cutterhead clogging detection model for shield machines.

(2) The outlier removal and denoising techniques proposed in this study guarantee a good overall performance of the proposed model.

(3) The training set is oversampled and balanced by a k-means-SMOTE algorithm, which improves the model performance.

(4) The proposed model achieves an accuracy of 98.85% and an $F1$ score of 0.9786, which are significantly higher than those of existing methods.

The remainder of this article is organized as follows. Section 2 presents the main materials and methods used in this study. The performance of the proposed model is evaluated and discussed in Section 3. Several discussions on the experimental results are presented in Section 4. Finally, conclusions are presented in Section 5.

2 Materials and methods

The framework of the proposed LSTM-autoencoder and enhanced transformer-based cutterhead clogging detection model is presented in Figure 2. It contains four main steps. First, the original data are preprocessed by the LSTM-autoencoder NN module and a denoising method. Then, time-domain features are extracted as the input to the constructed enhanced transformer-based NN. In the dataset construction step, a *k*-means-SMOTE algorithm is adopted to oversample the extracted time-domain features of the clogging data in the training set to balance the dataset and improve model performance. Eventually, the model is trained on the balanced training set, and clogging results can be obtained.

2.1 Materials and data preparation

The original data of this study are from the excavation data on a shield tunneling machine from 10:05 on May 9, 2021, to 15:57 on June 23, 2021. During the construction, cutterhead clogging appeared twice, from 11:01 on June 7, 2021, to 20:08 on June 10, 2021, and from 14:21 on June 14, 2021, to 18:15 on June 15, 2021. The data are sampled at a rate of 1 piece/min, and because of possible failures of sensors, some data may be missing. In summary, there are 64633 pieces of data available in total, and each piece of data contains information from 25 different sensors installed on a shield tunneling machine, such as cutterhead speed and torque. However, as shield tunneling machines need to be shut down regularly for maintenance, the tunneling process only makes up for a small part of time [26–28]. In the case of clogging detection, only the working data of shield tunneling machines are useful; consequently, only the working state data should be retained for analysis. In this study, working state data are obtained by keeping the data with nonzero values of cutterhead speed, torque, and penetration rate. After selection, 7505 pieces of working state data are obtained. Combined with recommendations from the literature in Section 1 and observation of the distribution of the data, 19 of the 25 parameters are selected for analysis. Details of the selected parameters are listed in Table 1.

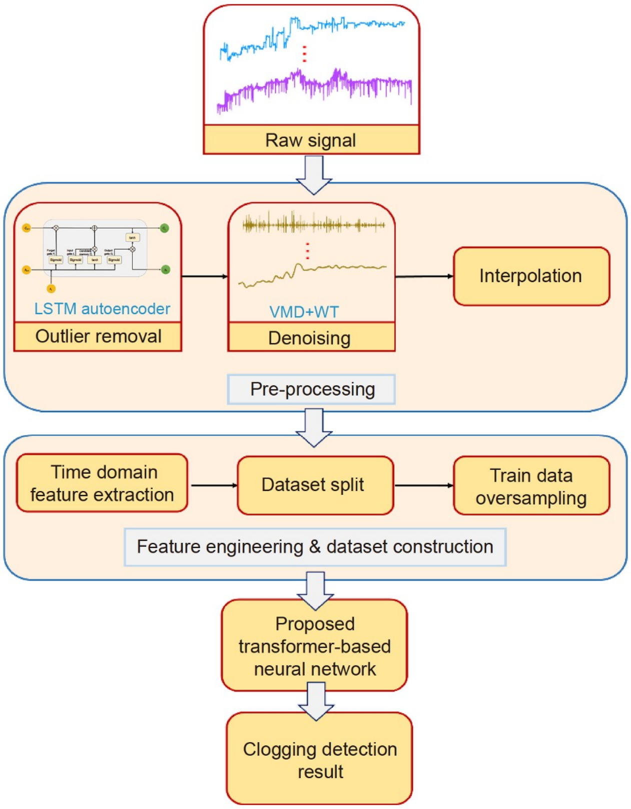
Raw signal
LSTM autoencoder
Outlier removal
VMD+WT
Denoising
Interpolation
Pre-processing
Time domain feature extraction
Dataset split
Train data oversampling
Feature engineering & dataset construction
Proposed transformer-based neural network
Clogging detection result

Figure 2 (Color online) Flowchart of the proposed model.

Table 1 Details of selected parameters

Number	Parameter
1	Cutterhead speed (r/min)
2	Cutterhead torque (kN m)
3–8	Propulsion pressure of cylinders groups A–F (bar)
9	Mean excavation speed (mm/min)
10	Penetration rate (mm/r)
11	Total thrust (kN)
12–17	Earth pressure of excavation soil bin No. 1–No. 6 (bar)
18–19	Earth pressure of working soil bin No. 1–No. 2 (bar)

2.2 LSTM-autoencoder-based outlier removal method

LSTM is an effective network architecture for analyzing sequences, which can make good use of both historical and current information on a given sequence [29]. The structure of an LSTM unit is shown in Figure 3.

The inputs to an LSTM unit at a specific time t are the memory unit and hidden state at time t−1 and the state input at time t. An LSTM unit contains three parts: forget gate, input gate, and output gate. The forget gate is used to determine whether historical information should be used. The output of the forget gate can be calculated as follows:

$$F_t = \text{Sigmoid}(X_t W_{xf} + H_{t-1} W_{hf} + b_f), \tag{1}$$

where F_t denotes the output of the forget gate, X_t denotes the input at time t, H_{t-1} denotes the hidden state at time t−1, and

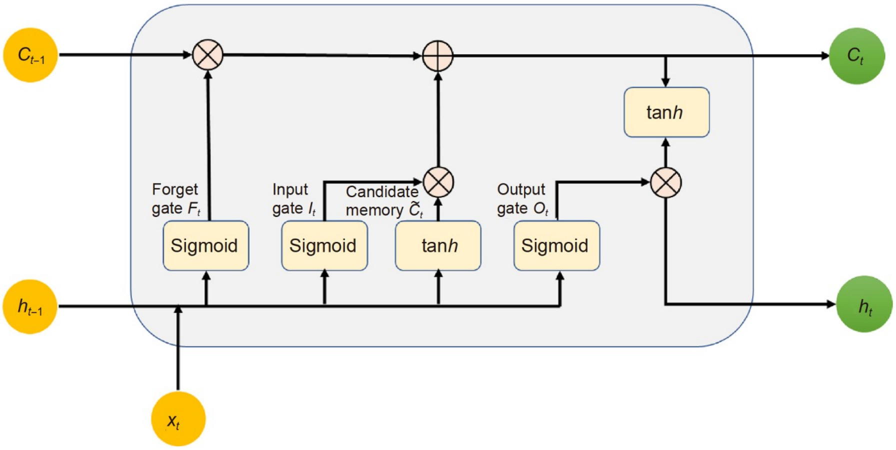

Figure 3 (Color online) Structure of an LSTM unit.

W_{xf}, W_{hf}, and b_f denote the corresponding weights and bias.

The input gate determines how much information about the current state should be retained, and its output can be calculated as follows:

$$I_t = \mathrm{Sigmoid}(X_t W_{xi} + H_{t-1} W_{hi} + b_i). \tag{2}$$

The output gate is directly related to the hidden state H_t, and its output is calculated as follows:

$$O_t = \mathrm{Sigmoid}(X_t W_{xo} + H_{t-1} W_{ho} + b_o). \tag{3}$$

C_t is called a memory unit, and $\widetilde{C}_t$ is the candidate memory; their values can be updated as follows:

$$\widetilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c), \tag{4}$$

$$C_t = F_t \circ C_{t-1} + I_t \circ \widetilde{C}_t, \tag{5}$$

where the operator ○ is the Hadamard product or elementwise product between matrices.

Finally, the hidden state output can be calculated as follows:

$$H_t = O_t \circ \tanh(C_t). \tag{6}$$

Autoencoder is an unsupervised learning method used to reconstruct an input [30]. Autoencoders use an encoder-decoder framework; the encoder can be seen as a feature extractor, which maps the input to a low-dimensional latent space, and the decoder can be treated as a reconstructor, used to recover the input from the latent space. To learn low-dimensional representations from an input, bottleneck network architectures are often used to build autoencoders [31]. For time sequence inputs, to minimize the reconstruction loss, the reconstructed sequence will try to learn the distribution of the inputs, and the reconstruction error of outliers will be higher because outliers are difficult to reconstruct. As LSTM is capable of extracting information from sequences and autoencoders can be used to detect outliers, their combination may have good performance in detecting sequence outliers. Srivastava et al. [32] proposed an LSTM-autoencoder model; the model was first developed to reconstruct and predict videos, but it turned out that the same model performs well in outlier detection.

A flowchart of the proposed LSTM-autoencoder-based outlier detection method is shown in Figure 4. First, the records from each sensor are standardized as follows:

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{7}$$

where x_i denotes the datapoint from the i-th sensor, and μ_i and σ_i denote the mean value and standard deviation of x_i, respectively. After standardization, the data of each parameter are intercepted by a series of sliding rectangular windows. The length of the window is 20; also, to better reconstruct the original signal, the window slides over one sample for each interception, meaning that, for each excavation parameter, there are 7486 pieces of small sequences to be reconstructed. Then, these sequences are used to train the autoencoder; some critical parameters of the autoencoder are shown in Figure 4. The model takes the small sequence both as the input and training label with Adam optimizer. Meanwhile, the loss function used for the model is the mean absolute error.

After training, the sequences are input into the model again to generate the corresponding reconstructed sequences. The reconstruction of the original signal is achieved using the mean value of the same point in different reconstructed small sequences. The autoencoders are trained separately for each parameter, and for each reconstructed data, the reconstruction error is defined as the absolute difference between the original and reconstructed data. For example, the reconstructed cutterhead speed and the distribution of its reconstruction error are shown in Figure 5.

From Figure 5, the reconstructed signal is much smoother than the original signal, and for most points, the reconstruction error is small, so it is reasonable to treat the data with large reconstruction errors as outliers. For each parameter, the outlier is defined by applying the 3σ criterion on the reconstruction error, which means that a threshold value

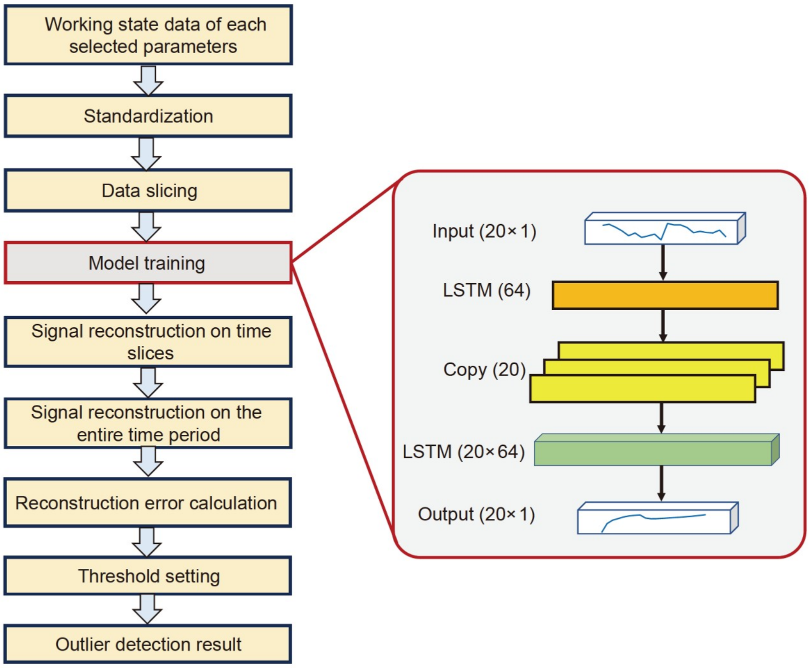
Working state data of each selected parameters
Standardization
Data slicing
Model training
Signal reconstruction on time slices
Signal reconstruction on the entire time period
Reconstruction error calculation
Threshold setting
Outlier detection result
Input (20× 1)
LSTM (64)
Copy (20)
LSTM (20× 64)
Output (20× 1)

Figure 4 Flowchart of the proposed outlier detection method.

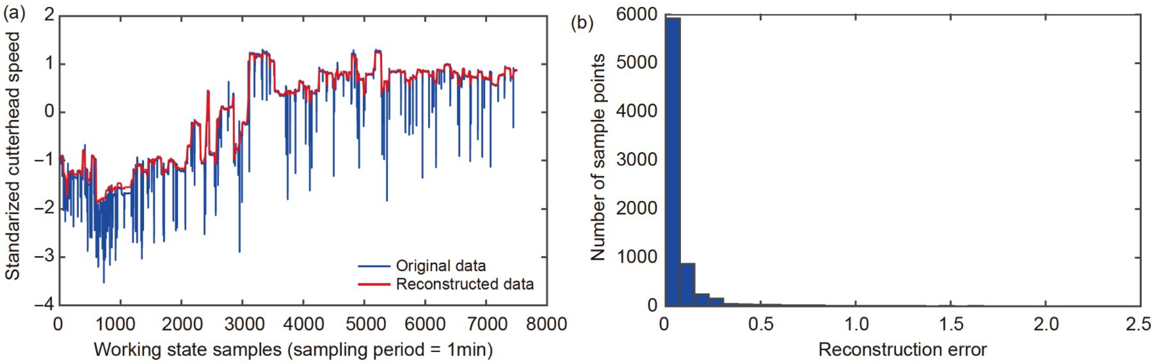

Figure 5 Reconstruction of cutterhead speed. (a) Comparison between the original and reconstructed data; (b) distribution of the reconstruction error.

is set as follows:

$$T = \mu + 3\sigma, \tag{8}$$

where T denotes the threshold, and μ and σ denote the mean value and standard deviation of the reconstruction error, respectively. For a given parameter, any data point that has a reconstruction error greater than the threshold will be treated as an outlier. For the working state data of shield tunneling machines, if all selected parameters are normal, the data will be treated as normal; otherwise, the data will be treated as outliers and discarded.

2.3 VMD-WT-based denoising

The purpose of outlier removal is to remove the data that are inconsistent with the global distribution of the original data, but the remaining data may still contain some noise because of measurement errors. Therefore, the data need to be further denoised. In this study, a combination of VMD and WT was used to denoise the data.

As an effective signal-processing method, VMD has been widely used in different engineering fields [33–35]. The idea of VMD is to decompose a signal into a series of band-limited amplitude-modulated-frequency-modulated signals called intrinsic mode functions (IMFs) [36]. An IMF can be expressed as follows:

$$u_k(t) = A_k(t)\cos(\phi_k(t)), \tag{9}$$

where $A_k(t)$ denotes the amplitude of the IMF, $\phi_k(t)$ denotes the phase, and $u_k(t)$ denotes the decomposed IMF. Considering a given signal f to be decomposed into k IMFs and the center frequency of each IMF u_k is ω_k, VMD can be achieved by solving the following optimization problem:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] \mathrm{e}^{-j\omega_k t} \right\|_2^2 \right\}, \qquad (10)$$

$$s.t. \sum_k u_k = f.$$

The solution to this optimization problem is equivalent to solving the saddle of the following augmented Lagrangian function L:

$$\begin{aligned} &L(\{u_k\}, \{\omega_k\}, \lambda) \\ &= \alpha \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] \mathrm{e}^{-j\omega_k t} \right\|_2^2 \\ &\quad + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_k u_k(t) \right\rangle, \end{aligned} \qquad (11)$$

where $\lambda(t)$ denotes the Lagrangian multiplier and α denotes a penalty factor. This function can be solved by an alternate direction method of multipliers. Given the initial values as u_k^1, w_k^1, and λ^1, the values can be updated using the following iterations:

$$u_k^{n+1}(\omega) = \frac{f(\omega) - \sum_{i \neq k} u_i(\omega) + \frac{\lambda(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2}, \qquad (12)$$

$$\omega_k^{n+1}(\omega) = \frac{\int_0^\infty \omega \left| u_k^{n+1}(\omega) \right|^2 \mathrm{d}\omega}{\int_0^\infty \left| u_k^{n+1}(\omega) \right|^2 \mathrm{d}\omega}, \qquad (13)$$

$$\lambda^{n+1}(\omega) = \lambda^n(\omega) + \tau\left(f(\omega) - \sum_k u_k^{n+1}(\omega)\right). \qquad (14)$$

The convergence condition is set to be

$$\frac{\sum_k \left\| u_k^{n+1} - u_k^n \right\|^2}{\left\| u_k^n \right\|^2} < \varepsilon. \qquad (15)$$

In this study, a signal is first decomposed by VMD to seven IMFs with the penalty factor α set to 1000. Based on the center frequency of the seven IMFs, the decomposed signals are labeled as IMF1–IMF7. IMF1 with the highest center frequency is considered a noise signal and is discarded. In addition, because the signal is not exactly the summation of those IMFs, there will also be a residual sequence representing the error between the original signal and the summation of IMFs:

$$Residual = f(t) - \sum_k u_k(t). \qquad (16)$$

Because the residual is just a reconstruction error and does not make many contributions to the original signal, it is also discarded in the denoising process.

The center frequencies of IMF5–IMF7 are low, and those signals are considered to represent the characteristics of the original signal and hence are kept unchanged. As for IMF2–IMF4, they are further decomposed by WT into five levels using sym4 as the wavelet function and are denoised with the empirical Bayes method with the threshold set to be the posterior median. After the wavelet threshold denoising, the denoised IMF2–IMF4 signals are recovered, and the denoised signal can be obtained by summing up IMF2 to IMF7. The flowchart of the denoising method is shown in Figure 6.

2.4 Time-domain feature extraction

To better detect cutterhead clogging, the working state data of a shield tunneling machine should be cut into smaller periods. Because the shield tunneling machine may be stopped between two working periods, two adjacent working state data may not come from the same working period. Therefore, the working state data can be split according to the following rule: for the data coming from the same working period, it is further split into smaller periods using a series of nonoverlapping rectangular windows with a length of 16 data points (i.e., each period contains the data generated in 5 min). If the cutterhead gets clogged at any time in those small sequences, the sequence should be annotated as clogging.

In the field of mechanical fault diagnosis, it is a common method to extract features of the sequences in time and frequency domains. However, as the sampling rate of a shield tunneling machine is quite low, severe aliasing problems will occur in the frequency domain, so it would be meaningless to extract features from the frequency domain. Therefore, only time-domain features were extracted for analysis. For feature selection, Wu et al. [37] used 11 time-domain features of vibration signals from rolling element bearings (REBs) to detect the degradation processes of REBs in electrical machines, Kuang et al. [38] performed a similar task using 13 time-domain features, Huang et al. [39] used root mean square, variance, maximum, skewness, kurtosis, and peak-to-peak value of signals from different sensors as time-domain features to predict tool wear in milling operations, and Bandyopadhyay et al. [40] used mean, standard deviation, skewness, kurtosis, shape factor, and crest factor of current signals to realize fault diagnosis of induction motors. Based on recommendations from the literature, 11 time-domain features are selected (Table 2).

2.5 Enhanced transformer block based on multihead attention mechanism

An attention mechanism can be described as finding an appropriate representation for a given query with a set of key-value pairs. The attention function can be calculated using different methods; nevertheless, considering the time consumption, dot-product attention is more often used and is given by

$$Attention(Q, K, V) = softmax(QK^T)V, \qquad (17)$$

where Q, K, and V denote the queries, keys, and values,

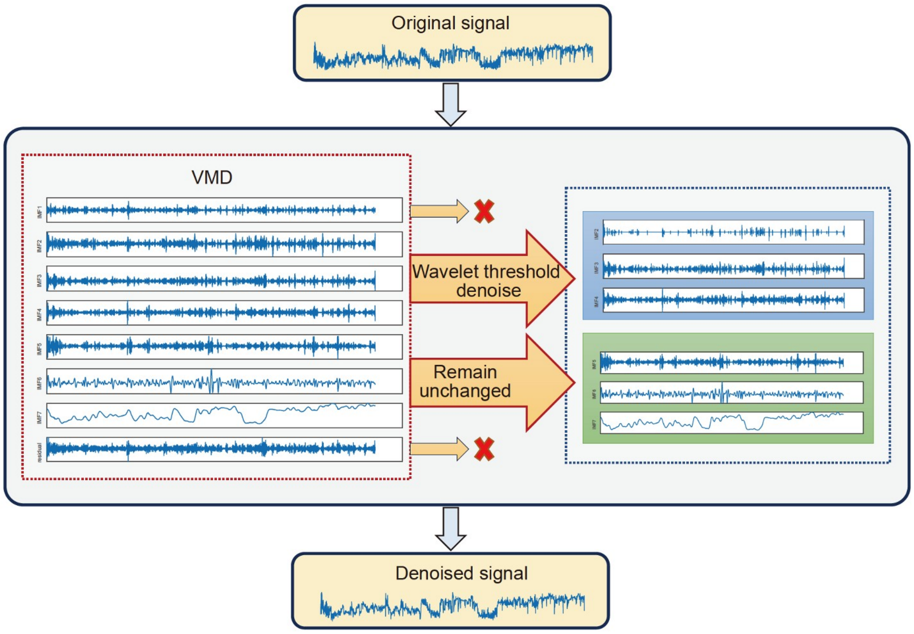

Figure 6 Flowchart of the denoising method.

Table 2 Extracted time-domain features

Feature number	Feature name	Equation
1	Absolute mean	$\left\|\overline{x}\right\| = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i\right\|$
2	Standard deviation	$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\overline{x})^2}$
3	Root mean square (RMS)	$x_{rms} = \frac{1}{N}\sum_{i=1}^{N}x_i^2$
4	Peak value	$x_{\text{peak}} = \max(\|x_i\|)$
5	Peak-to-peak value	$x_{p-p} = \max(x_i) - \min(x_i)$
6	Crest factor	$C_r = \frac{x_{\text{peak}}}{x_{\text{rms}}}$
7	Impulse factor	$I = \frac{x_{\text{peak}}}{\|\overline{x}\|}$
8	Shape factor	$S_f = \frac{x_{\text{rms}}}{\|\overline{x}\|}$
9	Clearance factor	$C_l = \frac{x_{\text{peak}}}{\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{x_i}\right)^2}$
10	Kurtosis	$K = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2\right)^2}$
11	Skewness	$S = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^3}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2\right)^{\frac{3}{2}}}$

respectively. If Q, K, and V are the same in attention score calculation, the attention is called self-attention. To gather more information from a given sequence, Vaswani et al. [41] proposed the multihead attention model. This model first linearly projects input queries, keys, and values h times to get h different lower dimensional queries, keys, and values, where h is called the model's head, and then performs scaled dot-product on those projected elements for attention calculation. Finally, the attention output of each head is concatenated and connected to a fully connected layer.

The scaled dot-product is similar to the original dot-product, except for scaling the input of the softmax function with a factor of $\frac{1}{\sqrt{d_k}}$ to avoid gradient vanishing, where d_k is the dimension of the keys. Combining this with the multihead attention mechanism, they further proposed a transformer block comprising a multihead attention layer and a feed-forward layer, which consists of two linear transformations with ReLU as the activation function. The layers are connected with residual connections, and the output of each residual block is normalized by layer normalization. The architecture of the transformer block is shown in Figure 7.

In this study, we modified the architecture to enhance its performance. The original multihead attention calculation requires the input dimension to be divisible by the key dimensions. To further improve the model, we ensure that the number of heads and dimensions of keys, values, and queries can be any value. To achieve this, we first map the input with a shape of (n_features, n_channels) into a latent space with the shape of (n_features, n_heads × key_dimensions) and then reshape it to different heads. In addition, unlike natural language processing problems, the input dimension of our model is a fixed number and is relatively small, so we used the dot-product attention without scaling, and instead of layer normalization, we still used batch normalization to normalize the outputs of the residual blocks.

As a linear transformation is applied to the input, the attention output and input do not have the same dimension. To make them summable for the residual connection, we need another fully connected layer to map the attention output to the input shape. To prevent overfitting, a dropout layer is also added between the two residual blocks of the transformer block.

In the feed-forward layer, we used a similar design as the original transformer block, with just changing the output dimension of the first fully connected layer to 32. The architecture of the modified transformer block is shown in Figure 8. For each modified transformer block, the main hyperparameters are the head numbers, key dimensions, and dropout rates, and the activation function for all linear transformations in the block is ReLU.

2.6 Architecture of the proposed model

The proposed model for clogging detection has four enhanced transformer blocks, and for each block, a dropout rate of 0.5 is applied. The first two blocks have two heads, with a key dimension of 10, and the following two blocks have four heads, and the key dimension for each head is 5. The reason is that our model first extracts the overall information with large key dimensions and few heads and then extracts more detailed information from the input with more heads and smaller key dimensions. The architecture of the proposed model is shown in Figure 9. Details of the proposed model are listed in Table 3.

3 Experiments and results

3.1 Data preprocessing

In this study, there were originally 7505 available working state data points. As the original working state data were contaminated by outliers, to better make use of the data, the outliers should be removed by the LSTM-autoencoder-based method. After outlier detection, there were 6469 data points available for further analysis.

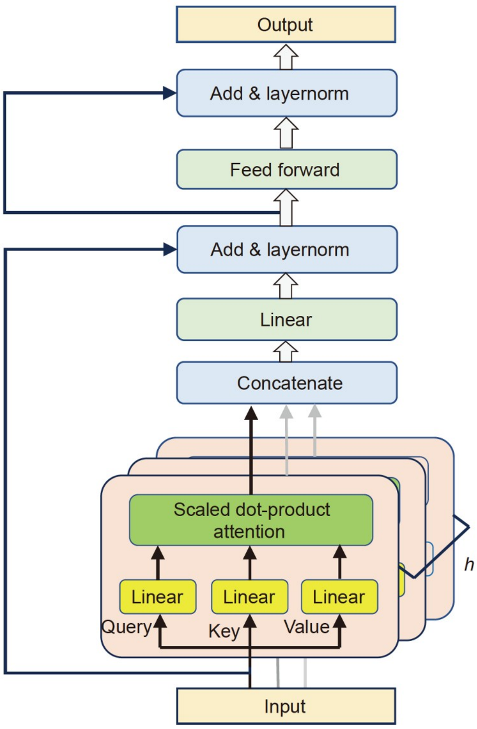

Figure 7 (Color online) Architecture of a transformer block.

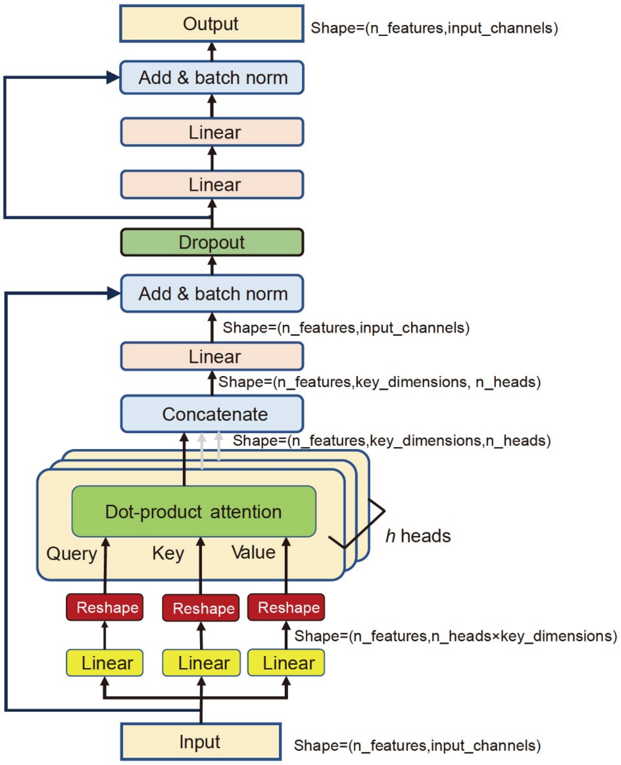
Output
Shape=(n_features,input_channels)
Add & batch norm
Linear
Linear
Dropout
Add & batch norm
Shape=(n_features,input_channels)
Linear
Shape=(n_features,key_dimensions, n_heads)
Concatenate
Shape=(n_features,key_dimensions,n_heads)
Dot-product attention
Query
Key
Value
h heads
Reshape
Reshape
Reshape
Shape=(n_features,n_heads×key_dimensions)
Linear
Linear
Linear
Input
Shape=(n_features,input_channels)

Figure 8 (Color online) Architecture of the modified transformer block.

To show the advantage of the proposed LSTM-autoencoder-based method in outlier detection, two other frequently used methods, namely isolation forest (IF) and local outlier factor (LOF), were used for comparison. The outlier results of the cutterhead speed signal are shown in Figure 10. Clearly, there are several time points where the cutterhead speed is significantly lower than their adjacent time points, and they should be treated as outliers. The proposed method successfully marked almost all of those data as outliers,

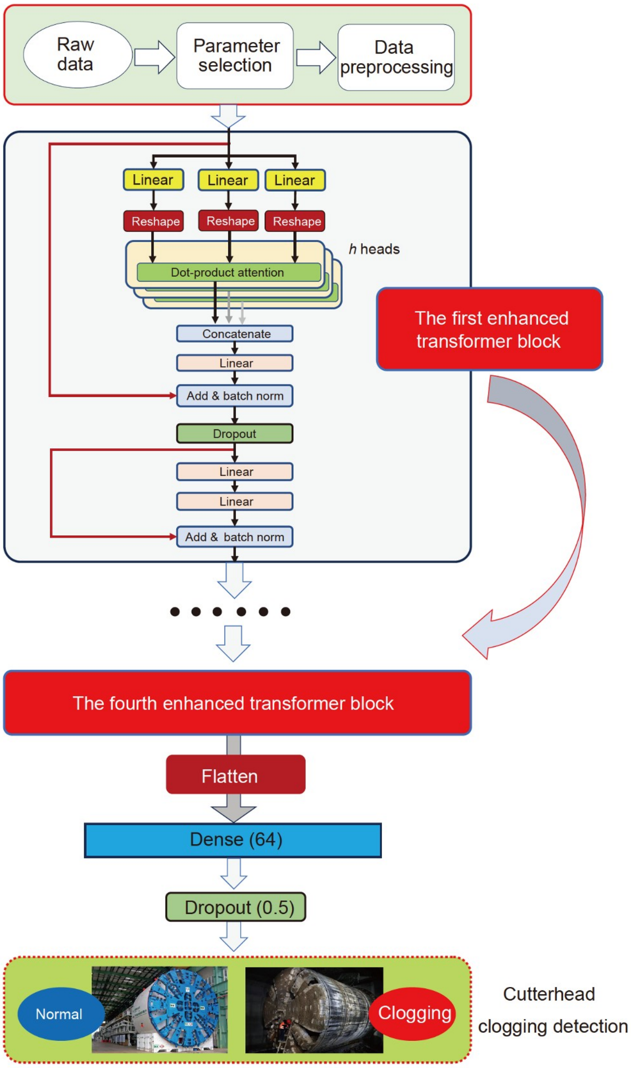

Figure 9 Architecture of the proposed model

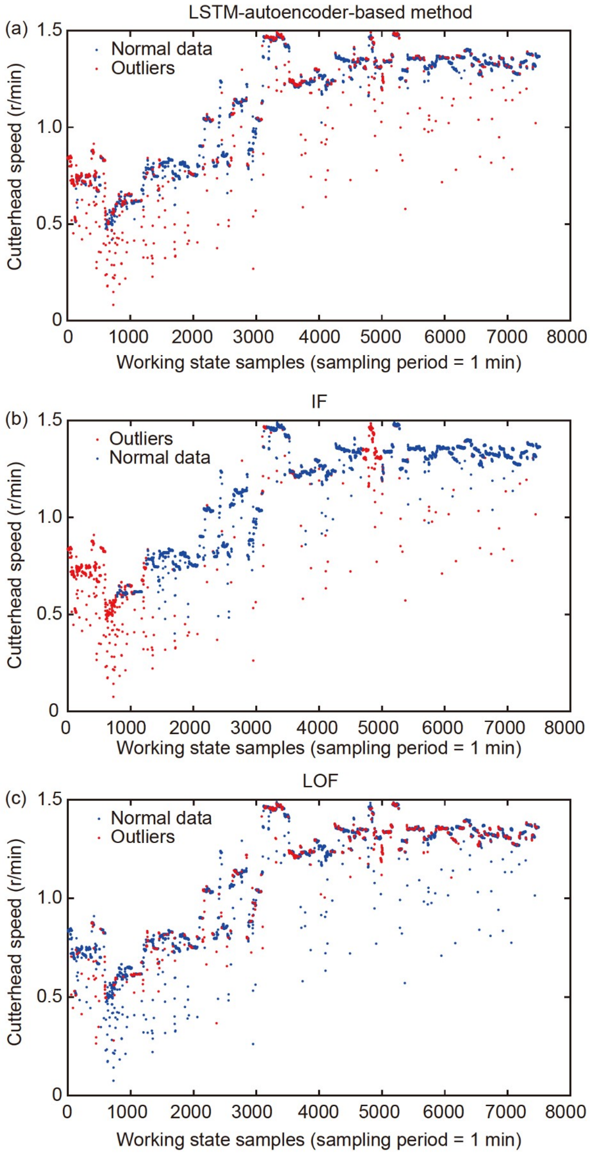

Figure 10 Outlier detection of cutterhead speed using different methods. (a) LSTM-autoencoder-based method; (b) IF; (c) LOF (n_neighbors = 35).

whereas IF only detected a small part of those data, and LOF did not recognize those data as outliers at all. As a result, the proposed outlier removal method is superior.

After outlier removal, the data were further denoised. Without loss of generality, considering the cutterhead speed signal as an example, the denoising result is shown in Figure 11.

Table 3 Details of the proposed model

Layer number	Description	Parameters	Output shape
Layer 1	Input	–	(11,19)
Layer 2	Enhanced transformer	n_heads = 2, key_dimension = 10, dropout = 0.5	(11,19)
Layer 3	Enhanced transformer	n_heads = 2, key_dimension = 10, dropout = 0.5	(11,19)
Layer 4	Enhanced transformer	n_heads = 4, key_dimension = 5, dropout = 0.5	(11,19)
Layer 5	Enhanced transformer	n_heads = 4, key_dimension = 5, dropout = 0.5	(11,19)
Layer 6	Flatten	–	209
Layer 7	Fully connected	units = 64, activation = ReLU	64
Layer 8	Dropout	rate = 0.5	64
Layer 9	Output (fully connected)	units = 1, activation = Sigmoid	1

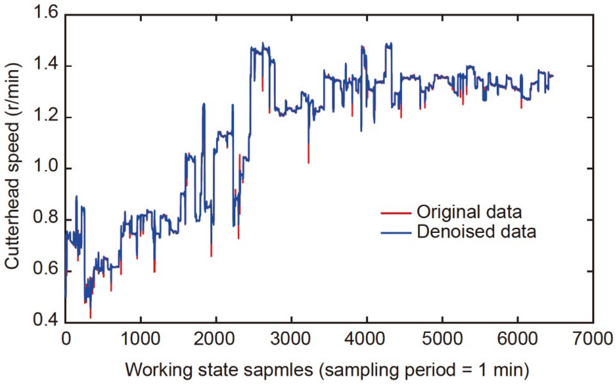

Figure 11 Denoising result of cutterhead speed.

Because the amount of data in this study is relatively small, particularly after denoising, linear interpolation was used to expand the dataset. The interpolation was performed in two steps; the first step was used to fill up the missing values caused by removal outliers or malfunction of the sensors. The method is to check the sampling time of every two adjacent data points; if the time difference is less than 10 min (i.e., 10 sample points), it is considered that the shield tunneling machine is still working, and the missing data between these two data points can be filled up by linear interpolation. Because the purpose of this step is only to make up for the missing values, it makes little contribution to expanding the dataset; the data should be further interpolated. In the second step of interpolation, for data sampled between every two consecutive time points, another two data points were interpolated between them. After the interpolation, there were 20429 available data points. Finally, the interpolated data were windowed, and their time-domain features were extracted to prepare the original data for further experiments.

In real-time applications, the LSTM-autoencoder model will first be trained with some historical data, and the main purpose of outlier detection will be to determine whether the current data are outliers. If the current data are outliers, instead of removing them, their values will be replaced by linear extrapolation of the two samples before them. After replacing the outliers, a rectangular window with a length of 6 will be applied to the current data and the data collected before. Unlike in the training process, interpolation will be performed before denoising in real-time applications to prevent the loss of useful information due to the denoising of short sequences. In addition, because in real-time applications, denoising is performed on windowed sequences and the length of the sequence is significantly shorter than in the training process, only VMD will be used to denoise the signal, meaning that the denoised signal will be the sum of IMF2–IMF7. After denoising, the prediction process will be similar to the training process: extract the time-domain features and use the pretrained transformer-based model to detect cutterhead clogging in real time.

3.2 Dataset split

As there are two periods of clogging data, the dataset could be split, using the first half as the training set and the second half as the test set. After splitting the dataset, there were 712 data points in the training set, among which 91 were clogging data points, and there were 713 data points in the test set, among which 120 were clogging data points. Given that the dataset is unbalanced with little clogging data available, the clogging data in the training set can be oversampled to better train the model.

The SMOTE is a simple and commonly used method for oversampling data in an imbalanced dataset for classification. The classic SMOTE algorithm can be achieved in three steps. First, a random data point in the minority class is selected, and then another data point from the minority class and its K-nearest neighbors (KNN) are selected. Finally, the convex linear combination of the two selected data points can be used to expand the dataset.

SMOTE is efficient in balancing imbalanced datasets, but if the minority class is not continuously distributed in space, the generated data may not have certain features of the minority class. Douzas et al. [42] improved the algorithm by using a *k*-means clustering algorithm before oversampling with SMOTE. In the improved algorithm, the data are first clustered into *k* groups, and then SMOTE is performed in the groups with a higher proportion of minority class samples. This algorithm can also handle an in-class imbalance in that it can decide how many samples should be interpolated into each cluster based on the sparsity of the clusters, and there will be more generated samples in sparse minority clusters than in dense ones.

The clogging data are oversampled until the training set is balanced (i.e., the amounts of normal data and clogging data are the same), the cluster number of *k*-means is set to 5, and the number of neighbors of the KNN in SMOTE is set to 2. To better visualize the results of the oversampling algorithm, the real and synthesized time-domain features in the training set were first reduced to 50 dimensions with principal component analysis and then reduced to 2 dimensions with *t*-distributed stochastic neighbor embedding (*t*-SNE) (Figure 12). The generated samples can effectively represent the clogging data and thus may help improve the performance of the proposed model.

3.3 Results and comparisons

Clogging detection is essentially a binary classification problem; therefore, several metrics for classification problems can be used to evaluate the performance of the proposed

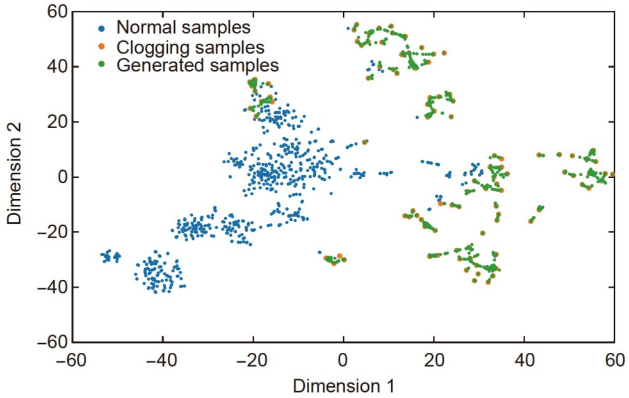

Figure 12 Visualization of the synthesized features.

model. The most commonly used metrics of classification problems are accuracy, precision, recall, and $F1$ score [43–47]. Consider a classification problem with n classes and N samples; for each class, according to the relationship between the predicted label y_{predict} and y_{real}, the classification results would be one of the four scenarios: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The metrics are defined as follows:

$$Acc = \frac{\sum y_{\text{predict}} = y_{\text{real}}}{N}, \tag{18}$$

$$P = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \tag{19}$$

$$R = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \tag{20}$$

$$F1 = \frac{2\times P\times R}{P+R}, \tag{21}$$

where Acc, P, R, and $F1$ denote the accuracy, precision, recall, and $F1$ score, respectively. In a classification problem with a balanced dataset, accuracy may be the best metric for model evaluation as it can reflect the proportion of the correctly classified samples. However, when the dataset is imbalanced, accuracy may not well-reflect performance. For example, in the clogging detection problem, if all data are predicted to be normal, an accuracy greater than 80% will still be achieved. In this case, the $F1$ score can reflect more information as both precision and recall are considered in this parameter. Consequently, to evaluate the performance of the proposed model, both the $F1$ score and accuracy are used, with the $F1$ score being a primary indicator.

The proposed model was trained on the training set, with all initial values randomly set. The optimizer used for training was Adam, and the momentum was set to 0.8. The loss function for the model was binary cross-entropy, and the learning rate was 0.0003. Keras package under the Tensorflow framework was used to build and train the model. As for hardware, a laptop equipped with an Intel(R) Core(TM) i7-11800H CPU and a 6GB NVIDIA GeForce RTX 3060 Laptop GPU was used.

In the training process, 4-fold cross-validation was used; for each fold of cross-validation, 75% of the data in the training set was used as training data, and the other 25% was used as validation data. The model was trained on each fold for 100 epochs with a batch size of 32. To save the training time and avoid possible overfitting, early stopping was also used in the training, i.e., when the validation loss did not decrease for 10 epochs, the training loop would stop for this fold of cross-validation. The model we used is the one with the smallest validation loss.

Because of the randomness in the training process, the performance of the model will not be the same in each training process. To evaluate the stability of the proposed model, we performed five repeated experiments; the results are shown in Table 4.

The results show that the proposed model can achieve high performance in clogging detection with good stability. To visualize the classification result of the proposed model, the confusion matrix of the fifth experiment on the expanded training and test sets is shown in Figure 13.

To show the advantage of the proposed model, several other well-known models were also used as comparisons, including KNN, support vector machine (SVM), RF, extreme gradient boosting classifier (XGBC), and deep NN (DNN). Besides, some effective fault diagnosis-related methods were also used as comparisons. For example, Ke et al. [48] considered the synchrosqueezing transform (SST) spectra as inputs and built a deep convolutional NN model (SST-DCNN) to detect possible faults in modular multilevel converters, Yuan et al. [49] and Zou et al. [50] proposed a general convolutional NN (CNN) based end-to-end diagnostic framework for manufacturing systems. Some key parameters of the comparison models are listed in Table 5.

All comparison models were trained on the expanded training set and evaluated under the same test set as the proposed model. The performance of these models is shown in Figure 14. The results show that our model outperforms the comparison models, attributable to the proposed processing method and designed enhanced transformer block. After preprocessing, the data were well-denoised and were no longer contaminated by outliers so that the features of the data could be recognizable to our model. Particularly, the proposed model achieves accurate detection of shield machine cutterhead clogging status, with 98.85% accuracy and a 0.9786 $F1$ score. KNN obtained 97.35% accuracy and a 0.9512 $F1$ score. SVM obtained 95.45% accuracy and a 0.9078 $F1$ score. RF obtained 92.55% accuracy and a 0.8347 $F1$ score. XGBC obtained 93.06% accuracy and a 0.8484 $F1$ score. DNN obtained 96.34% accuracy and a 0.9281 $F1$ score. SST-DCNN obtained 72.93% accuracy and a 0.5581 $F1$ score, whereas CNN obtained 92.57% accuracy and a

Table 4 Performance of our model in five repeated experiments

Experiment number	Accuracy (%)	Precision	Recall	$F1$ score
1	99.16	0.9950	0.9750	0.9847
2	99.02	0.9873	0.9775	0.9823
3	99.44	0.9966	0.9833	0.9898
4	97.34	0.9845	0.9208	0.9491
5	99.29	0.9958	0.9792	0.9872
Mean	98.85	0.9918	0.9672	0.9786

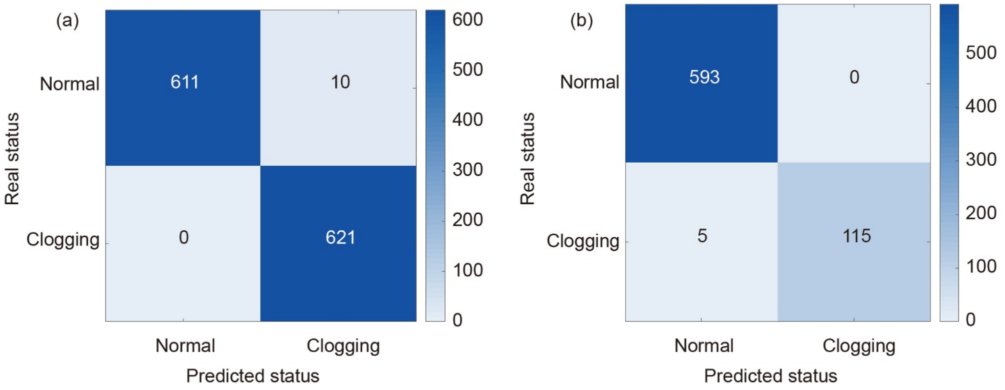

Figure 13 (Color online) Confusion matrix of the proposed model on the following. (a) Expanded training set and (b) test set.

Table 5 Key parameters of the comparison models [a)]

Name	Key parameters
KNN	N_neighbors = 12
SVM	Kernel = RBF, C = 10, gamma = 0.003
RF	Max_depth = 3, n_estimators = 12
XGBC	L2_penalty_factor = 0.01, learning_rate = 0.1, gamma = 1, max_depth = 3

DNN	Input_shape = 209,
	structure = {200-128-64-32-dropout(0.5)-1}

SST-DCNN	Input_shape = 96 × 16@19, padding = “the same”

CNN	Input_shape = 16@19, kernel_size = 3, stride = 1,
	padding = “the same,” structure = {conv-maxpool-conv-maxpool-conv-globalMaxpool-fc(1)}

a) in the parameters of CNN, “conv” means the convolutional layer, “maxpool” means the maxpooling layer, “globalMaxpool” means the global maxpooling layer, “fc(1)” means a fully connected layer with a single neuron. All convolutional layers use the same hyperparameters as stated in the table.

0.8478 $F1$ score. Therefore, it could be concluded that the proposed model can achieve high precision in cutterhead clogging detection of shield machines and hence help realize intelligent fault diagnosis on shield machines.

To have a better understanding of the clogging results of the different models, their confusion matrices are shown in Figure 15. All employed models can correctly recognize the normal data; the main difference is in detecting clogging data. Compared with other models, only five clogging data points were wrongly detected by our model. As shown in Figure 12, some clogging data may share a similar feature with the normal data, and it is those data that affect the accuracy of clogging detection models. Compared with other models, the proposed model can dig more information from the data and hence better distinguish clogging data from normal data and have a better performance.

Notably, the performance of end-to-end models and time-frequency-domain feature-based models is worse than that of time-domain feature-based models. For CNN-based models, although CNNs can effectively extract information from raw timeseries data, their performance will be better for longer sequences. Because the window size in this study is relatively small, CNNs may not outperform time-domain feature-based methods. As for SST-DCNN, because of the extremely low sampling frequency, frequency-domain and time-frequency-domain features are highly contaminated by aliasing; thus, it would be difficult to extract useful information from those features. Specifically, several SST

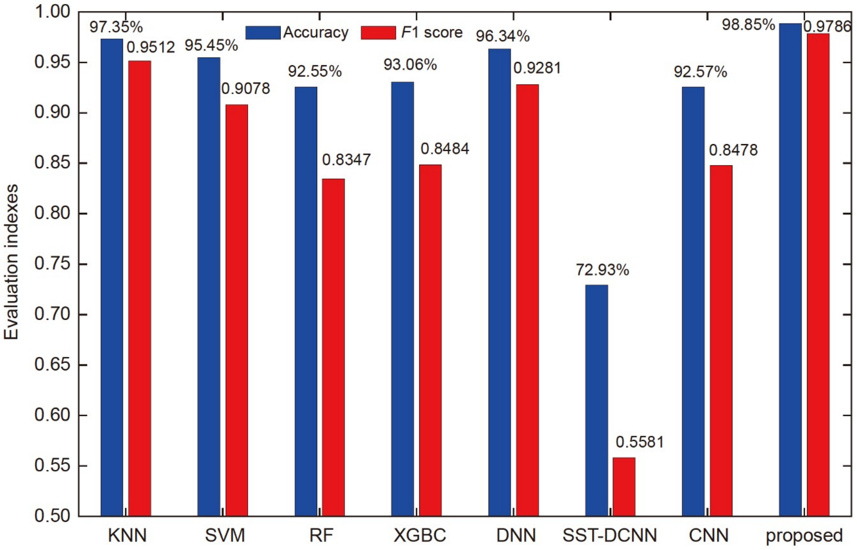

Figure 14 Performance of different methods on the test set.

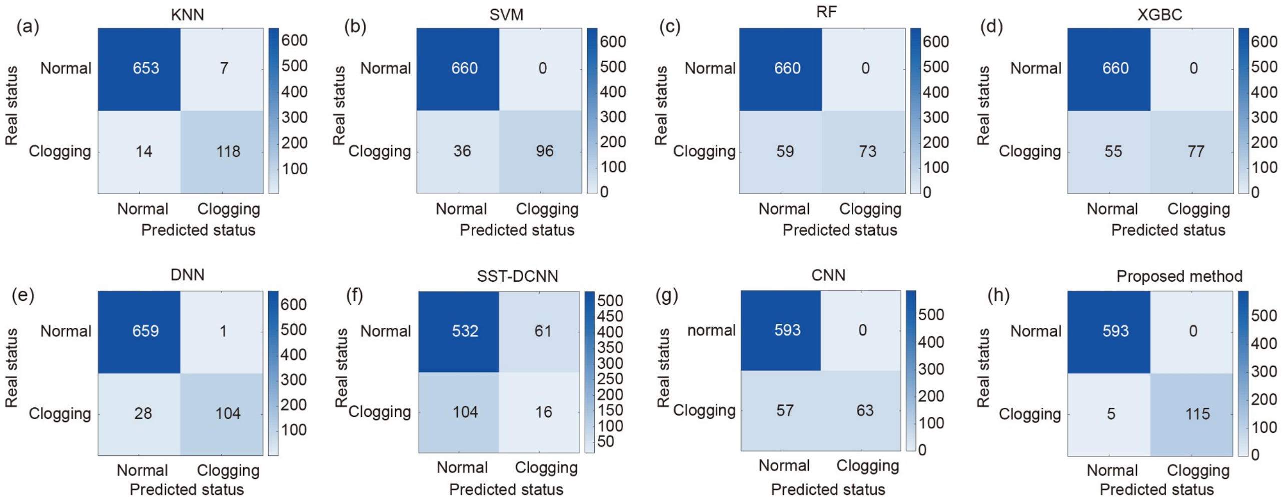

Figure 15 Confusion matrices of different models. (a) KNN; (b) SVM; (c) RF; (d) XGBC; (e) DNN; (f) SST-DCNN; (g) CNN; (h) Proposed.

spectra of the cutterhead torque under normal and clogging conditions are shown in Figure 16. From the figure, the difference between normal and clogged samples is unclear from SST spectra. Therefore, time-frequency-domain feature-based models may not have a good performance.

4 Discussion

In the training process, a k-means-SMOTE algorithm was used to oversample clogging data in the training set, and the dataset was made balanced. However, as balancing the dataset will require enormous synthesized data, it may not be the best choice for synthesizing data on such a large scale. To evaluate the proposed model with different amounts of oversampled data, the ratio of clogging data to normal data at the percentages of 30%, 50%, and 80%, as well as the original data without oversampling, were evaluated for comparison. The distribution of data under these conditions can be visualized in Figure 17.

The training processes for these models are the same as described in Section 3.2; the average accuracy and $F1$ score of the model with different numbers of oversampled data are shown in Figure 18. From the figure, oversampling helps the model to achieve higher performances. Notably, with more synthesized data, the model performance improved gradually until the amount of clogged data was 80% of the normal data. With this proportion of clogging data to normal data, the accuracy improved by about 0.7% and the $F1$ score improved by about 0.12. The absolute value of the improve-

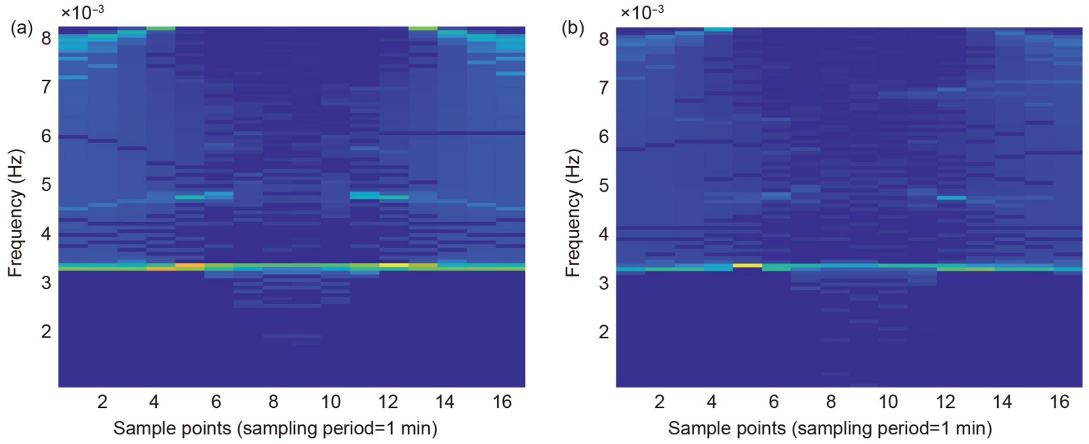

Figure 16 SST spectra of (a) normal samples and (b) clogged samples.

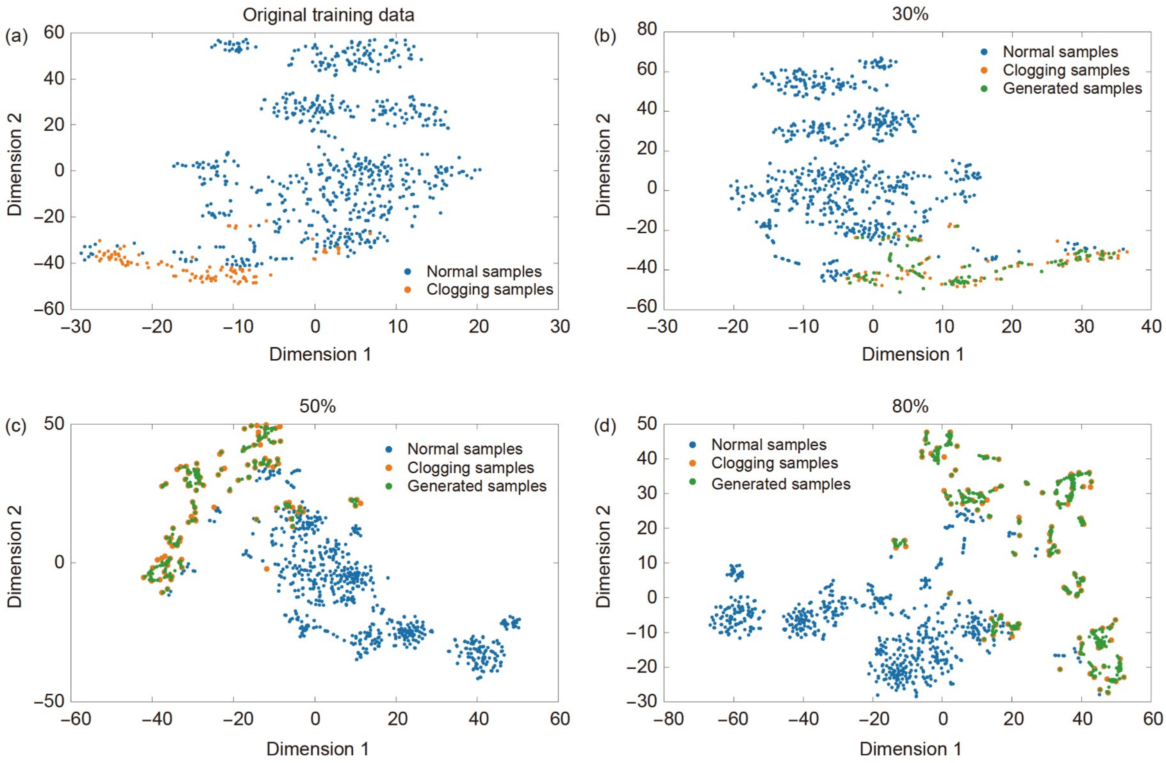

Figure 17 Visualization of data with different ratios of clogging data to normal data. (a) 14.63% (original training data); (b) 30%; (c) 50%; (d) 80%.

ment may seem small, but as the model performed well on the test set, a 0.7% improvement in the classification accuracy can reduce the wrongly classified samples by 46.85%, which is a huge improvement.

However, when the dataset is balanced by oversampling, the model performance drops, which is because there are too many generated samples, and thus, their distribution is too dense in this situation (Figure 12). Because the oversampling algorithm presumes that the clogging data follows a certain distribution, and with too much oversampled data, the randomness of the distribution may be less, and some clogging features that do not appear in the training set may not be correctly detected because of overfitting caused by oversampling. Nevertheless, as there are few clogging data available, the oversampled data can still be used to generate more useful data for training; thus, the model will still have a better performance than training with the original training set.

The proposed preprocessing method can also help improve performance. To examine the effect of the proposed preprocessing method, several comparisons were made. The proposed model was trained under four conditions: without preprocessing, with only removing the outliers, with only denoising, and with both preprocessing techniques. All training data in these situations were balanced by the *k*-means-SMOTE algorithm. The results are shown in Figure 19. Compared with using the time-domain features extracted from the original working state data, the proposed preprocessing method improves the clogging detection accuracy by about 1.5% and the *F*1 score by about 0.026, meaning that the wrongly detected samples became less than 50%. In addition, both the outlier removal and denoising techniques can make certain improvements to the model performance. On this specific dataset, the effect of each of these techniques seems similar, and using a combination of both can further improve the model performance and make it more precise.

5 Conclusion

In this study, we propose a novel LSTM-autoencoder and enhanced transformer-based detection model for cutterhead clogging status detection. To better train the model, after selecting the working state data of shield machines from historical excavation data, we designed an LSTM-autoencoder NN module to remove outliers and employed a VMD-WT-based denoising method to further denoise the data. Then, the *k*-means-SMOTE algorithm was adopted to oversample the extracted time-domain features of the clogging data in the training set to balance the dataset and improve model performance. On this basis, an enhanced transformer-based NN was constructed to extract essential implicit features and detect cutterhead clogging status. Comprehensive comparisons with existing models were performed to verify the proposed model, indicating that the proposed model outperformed the comparison models. Meanwhile, the proposed model can achieve an accuracy of 98.85% and an *F*1 score of 0.9786, indicating that the proposed model can achieve high precision in the cutterhead clogging detection of shield tunneling machines and hence can help realize intelligent fault diagnosis on shield tunneling machines. In the future, efforts will be made to combine the excavation parameters, geological environment of shield tunneling machines, and ideas in other fields [51–54] to further improve the generalizability of data-driven clogging detection models.

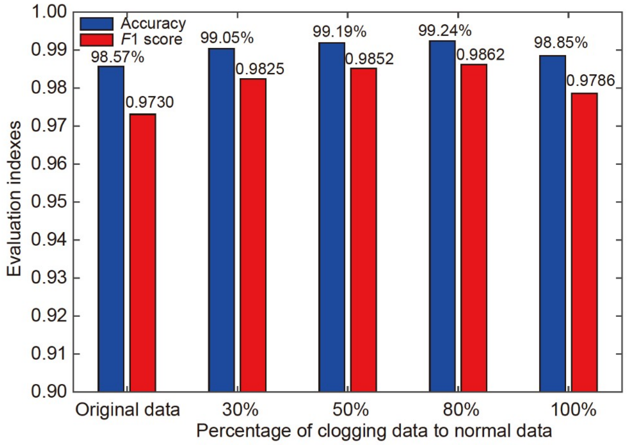

Figure 18 Performance of the proposed model with different amounts of synthesized data.

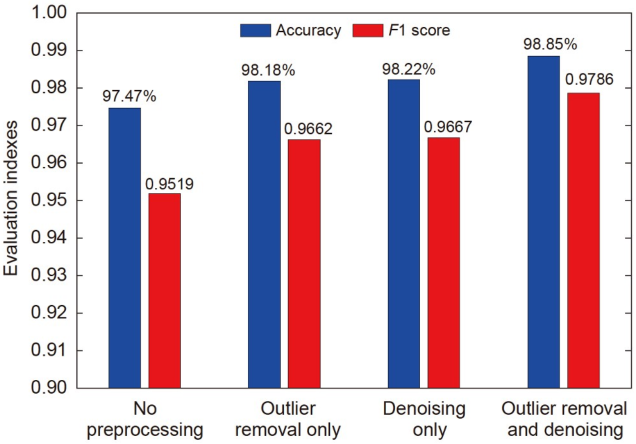

Figure 19 Evaluation of the effect of the proposed preprocessing method.

This work was supported by the National Key R&D Program of China (Grant No. 2018YFB1702503), Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102), and the State Key Laboratory of Mechanical System and Vibration (Grant No. MSVZD202103).

1 Wang L T, Sun W, Gong G F, et al. Method to avoid the structural interference of the thrust system of a shield tunneling machine. Sci China Tech Sci, 2017, 60: 701–712
2 Zhang Q, Su C X, Qin Q H, et al. Modeling and prediction for the thrust on EPB TBMs under different geological conditions by considering mechanical decoupling. Sci China Tech Sci, 2016, 59: 1428–1434
3 Miao J B, Lu D C, Lin Q T, et al. Time-dependent surrounding soil pressure and mechanical response of tunnel lining induced by surrounding soil viscosity. Sci China Tech Sci, 2021, 64: 2453–2468
4 Qin C, Shi G, Tao J, et al. Precise cutterhead torque prediction for shield tunneling machines using a novel hybrid deep neural network. Mech Syst Signal Pr, 2021, 151: 107386
5 Liu S L, Yang K H, Cai J, et al. Prediction of geological parameters during tunneling by time series analysis on in situ data. Comput Intel Neurosc, 2021, 2021: 3904273
6 Yang Y, Li X. Review on the issue of clogging occurred in shield tunneling (in Chinese). Chin J Undergr Sp Eng, 2020, 16: 1030–1038
7 Alberto-Hernandez Y, Kang C, Yi Y, et al. Mechanical properties of clayey soil relevant for clogging potential. Int J Geotech Eng, 2018, 12: 529–536
8 Alberto-Hernandez Y, Kang C, Yi Y, et al. Clogging potential of tunnel boring machine (TBM): A review. Int J Geotech Eng, 2018, 12: 316–323
9 Avunduk E, Copur H. Effect of clogging on EPB TBM performance: A case study in akfirat waste water tunnel, Turkey. Geotech Geol Eng, 2019, 37: 4789–4801
10 Fang Y, Wang K, Tao L, et al. Experimental study on clogging of cutterhead for panel earth-pressure-balance shield tunneling in cohesive strata (in Chinese). Chin J Geotech Eng, 2020, 42: 1651–1658
11 Chen Z, Bezuijen A, Fang Y, et al. Experimental study and field

validation on soil clogging of EPB shields in completely decomposed granite. Tunn Undergr Sp Tech, 2022, 120: 104300

12 Li Z, Zhai Z, Zhao K. Causes of mud cake formation on cutter head of slurry shield and its control technology (in Chinese). Chin J Undergr Sp Eng, 2014, 10: 1866–1871

13 Li X, Yang Y, Li X, et al. Criteria for cutting head clogging occurrence during slurry shield tunneling. Appl Sci, 2022, 12: 1001

14 Chen Y, Zhang R, Li M. Prevention and cure of cutter-head mud cake of composite EPBM in compound stratum (in Chinese). J Munic Technol, 2015, 33: 165–169

15 Hollmann F S, Thewes M. Assessment method for clay clogging and disintegration of fines in mechanised tunnelling. Tunn Undergr Sp Tech, 2013, 37: 96–106

16 Hollmann F, Thewes M. Evaluation of the tendency of clogging and separation of fines on shield drives/Bewertung der Neigung zur Verklebungsbildung und zur Feinkornfreisetzung bei Schildvortrieben. Geomechanik Tunnelbau, 2012, 5: 574–580

17 Kang C, Yi Y, Bayat A. Performance evaluation of TBM clogging potential for plain and conditioning soil using a newly developed laboratory apparatus. Int J Geotech Eng, 2020, 14: 463–472

18 de Oliveira D G G, Thewes M, Diederichs M S, et al. EPB tunnelling through clay-sand mixed soils: Proposed methodology for clogging evaluation. Geomechanik und Tunnelbau, 2018, 11: 375–387

19 Yu H, Tao J, Huang S, et al. A field parameters-based method for real-time wear estimation of disc cutter on TBM cutterhead. Automat Constr, 2021, 124: 103603

20 Tao J, Yu H, Qin C, et al. A gene expression programming-based method for real-time wear estimation of disc cutter on TBM cutterhead. Neural Comput Applic, 2022, 34: 20231–20247

21 Qin C, Xiao D, Tao J, et al. Concentrated velocity synchronous linear chirplet transform with application to robotic drilling chatter monitoring. Measurement, 2022, 194: 111090

22 Zhou S, Liu S, Kang Y, et al. Physics-based machine learning method and the application to energy consumption prediction in tunneling construction. Adv Eng Inf, 2022, 53: 101642

23 Qi W, Wang L, Zhou S, et al. Total loads modeling and geological adaptability analysis for mixed soil-rock tunnel boring machines. Undergr Space, 2022, 7: 337–351

24 Zhou S, Kang Y, Xie H, et al. An approach integrating dimensional analysis and field data for predicting the load on tunneling machine. KSCE J Civ Eng, 2019, 23: 3180–3187

25 Zhai J, Wang Q, Yuan D, et al. Clogging risk early warning for slurry shield tunneling in mixed mudstone-gravel ground: A real-time self-updating machine learning approach. Sustainability, 2022, 14: 1368

26 Yu H, Tao J, Qin C, et al. Rock mass type prediction for tunnel boring machine using a novel semi-supervised method. Measurement, 2021, 179: 109545

27 Qin C, Huang G, Yu H, et al. Geological information prediction for shield machine using an enhanced multi-head self-attention convolution neural network with two-stage feature extraction. Geosci Front, 2023, 14: 101519

28 Yu H, Qin C, Tao J, et al. A multi-channel decoupled deep neural network for tunnel boring machine torque and thrust prediction. Tunnll Undergr Sp Technol, 2023, 133: 104949

29 Jin Y, Qin C, Tao J, et al. An accurate and adaptative cutterhead torque prediction method for shield tunneling machines via adaptative residual long-short term memory network. Mech Syst Signal Pr, 2022, 165: 108312

30 Yu H, Tao J, Qin C, et al. A novel constrained dense convolutional autoencoder and DNN-based semi-supervised method for shield machine tunnel geological formation recognition. Mech Syst Signal Pr, 2022, 165: 108353

31 Pang G, Shen C, Cao L, et al. Deep learning for anomaly detection. ACM Comput Surv, 2022, 54: 1–38

32 Srivastava N, Mansimov E, Salakhudinov R. Unsupervised learning of video representations using LSTMs. In: Proceedings of the 32nd International Conference on Machine Learning. Lille, 2015. 843–852

33 Qin C, Shi G, Tao J, et al. An adaptive hierarchical decomposition-based method for multi-step cutterhead torque forecast of shield machine. Mech Syst Signal Pr, 2022, 175: 109148

34 Shi G, Qin C, Tao J, et al. A VMD-EWT-LSTM-based multi-step prediction approach for shield tunneling machine cutterhead torque. Knowl-Based Syst, 2021, 228: 107213

35 Kaur C, Bisht A, Singh P, et al. EEG Signal denoising using hybrid approach of variational mode decomposition and wavelets for depression. Biomed Signal Proces, 2021, 65: 102337

36 Dragomiretskiy K, Zosso D. Variational mode decomposition. IEEE Trans Signal Process, 2014, 62: 531–544

37 Wu J, Wu C, Cao S, et al. Degradation data-driven time-to-failure prognostics approach for rolling element bearings in electrical machines. IEEE Trans Ind Electron, 2019, 66: 529–539

38 Kuang J, Xu G, Zhang S, et al. Learning a superficial correlated representation using a local mapping strategy for bearing performance degradation assessment. Meas Sci Technol, 2021, 32: 065003

39 Huang Z, Zhu J, Lei J, et al. Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. J Intell Manuf, 2020, 31: 953–966

40 Bandyopadhyay I, Purkait P, Koley C. Performance of a classifier based on time-domain features for incipient fault detection in inverter drives. IEEE Trans Ind Inf, 2019, 15: 3–14

41 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, 2017. 6000–6010

42 Douzas G, Bacao F, Last F. Improving imbalanced learning through a heuristic oversampling method based on *k*-means and SMOTE. Inf Sci, 2018, 465: 1–20

43 Jin Y R, Qin C J, Zhang Z N, et al. A multi-scale convolutional neural network for bearing compound fault diagnosis under various noise conditions. Sci China Tech Sci, 2022, 65: 2551–2563

44 Jin Y R, Li Z Y, Liu Y Q, et al. Multi-class 12-lead ECG automatic diagnosis based on a novel subdomain adaptive deep network. Sci China Tech Sci, 2022, 65: 2617–2630

45 Liu G K, Shen W M, Gao L, et al. Active label-denoising algorithm based on broad learning for annotation of machine health status. Sci China Tech Sci, 2022, 65: 2089–2104

46 Han Z Z, Huang Y Z, Li J, et al. A hybrid deep neural network based prediction of 300 MW coal-fired boiler combustion operation condition. Sci China Tech Sci, 2021, 64: 2300–2311

47 Jin Y, Li Z, Qin C, et al. A novel attentional deep neural network-based assessment method for ECG quality. Biomed Signal Proces, 2023, 79: 104064

48 Ke L, Zhang Y, Yang B, et al. Fault diagnosis with synchrosqueezing transform and optimized deep convolutional neural network: An application in modular multilevel converters. Neurocomputing, 2021, 430: 24–33

49 Yuan Y, Ma G, Cheng C, et al. A general end-to-end diagnosis framework for manufacturing systems. Natl Sci Rev, 2020, 7: 418–429

50 Zou F, Zhang H, Sang S, et al. An anti-noise one-dimension convolutional neural network learning model applying on bearing fault diagnosis. Measurement, 2021, 186: 110236

51 Qin C, Jin Y, Zhang Z, et al. Anti-noise diesel engine misfire diagnosis using a multi-scale CNN-LSTM neural network with denoising module. CAAI Trans Intell Technol, 2023, doi: 10.1049/cit2.12170

52 Liu S, Lu Y, Li J, et al. A blockchain-based interactive approach between digital twin-based manufacturing systems. Comput Indu Eng, 2023, 175:108827

53 Yu H, Sun H, Tao J, et al。 A multi-stage data augmentation and ABi-ResNet-based method for EPB utilization factor prediction. Automat Constr, 2023, 147: 104734

54 Liu S, Lu Y, Zheng P, et al. Adaptive reconstruction of digital twins for machining systems: A transfer learning approach. Robot Com-Int Manuf, 2022, 78: 102390