https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

Research on fault diagnosis of wind turbine bearings based on CEEMDAN and DQN

Rui Wang[1], Zhi Qiao[1], Changfu Liu[1], Yubai Liu[1]

[1]School of Mechanical Engineering, Liaoning Petrochemical University, Fushun 113001, China

Correspondence to: Rui Wang, 18341328936@163.com

Abstract. To address the issue of overfitting in deep learning methods for diagnosing bearing faults in wind turbine generators, this paper proposes a fault diagnosis method that combines Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) and Deep Q-learning (DQN) algorithms to achieve fault diagnosis of wind turbine bearings. First, the original vibration signals of the bearings are decomposed using CEEMDAN to obtain a series of IMF components ranging from high to low frequencies. Second, the sample entropy of the selected IMF components is calculated to construct feature vectors. Finally, a deep neural network combined with the Q-learning algorithm is used to train a DQN multi-class fault classifier, enabling precise identification of bearing faults in wind turbines. Through comparative experiments with other combined methods, the superiority and effectiveness of the proposed method in wind turbine bearing fault diagnosis are validated.

1 Introduction

The modern energy industry is accelerating its transition towards green and intelligent development, facing significant opportunities and technical challenges. In wind power generation systems, wind turbines serve as the core equipment for energy conversion, and their operational stability directly determines energy production efficiency. Bearings in the transmission system, as the core load-bearing components of wind turbines, are critical for ensuring efficient power transmission. Bearings are subjected to extreme wind loads, complex torque fluctuations, and harsh environmental corrosion over extended periods, with failure rates accounting for approximately 23% to 35% of all mechanical failures in wind turbines (de Azevedo et al., 2016). Bearing failures result in prolonged repair cycles, significant economic losses, and may even trigger cascading shutdowns, making their fault diagnosis research critically important (Kusiak and Verma, 2012). Therefore, an accurate and reliable intelligent bearing fault diagnosis system is of decisive significance for achieving predictive maintenance in wind farms, reducing the risk of major accidents, and lowering operational and maintenance costs (Qiao and Lu, 2015).

Common methods for analysing wind turbine bearing vibration signals can be categorised into time-domain analysis, frequency-domain analysis, time-frequency domain analysis, and empirical mode decomposition (Gao and Liu, 2021; Toma et al., 2021). In bearing vibration signal analysis, time-domain analysis typically involves statistical characteristics to

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

determine bearing condition, with statistical characteristics including mean, variance, standard deviation, and other statistical measures to assess signal characteristics (Gupta and Pradhan, 2017). Saruhan et al. (2014) studied the vibration analysis of rolling element bearings (REBs), using vibration response root mean square (RMS) and peak factor as bearing fault diagnosis indicators to determine the severity of the fault. Frequency domain analysis involves transforming the signal from the time domain to the frequency spectrum to study the frequency components and distribution present in the signal, often represented as a spectrum plot (Toma et al., 2020). By analysing characteristics such as the peak frequency position distribution and frequency standard deviation in the plot, the frequency domain features of the vibration signal are obtained. Kumar et al. (2022) addressed the problem of identifying defect pulses in strong interference by proposing a criterion called the edge band selection index, which determines the presence of defects by calculating the edge-enhanced square envelope spectrum. Wavelet transform (WT) decomposes signals using wavelet basis functions (Saxena et al., 2016). Wavelet basis functions can be localised simultaneously in both time and frequency domains, enabling variable time-frequency windows during signal processing. This allows for adjustable window scale factors while applying moving windows to signals, thereby achieving multi-resolution analysis and obtaining effective time-frequency domain feature information. Lou and Loparo (2004) used the wavelet transform to generate feature vectors, trained an adaptive neural fuzzy inference system (ANFIS) as a diagnostic classifier, and reliably separated bearing fault conditions under load changes. Goyal et al. (2020) processed vibration signals using the discrete wavelet transform, then calculated features such as energy entropy and kurtosis, selected the optimal features through distance metrics, and finally input them into a support vector machine (SVM) to complete the diagnosis.

Empirical Mode Decomposition (EMD) is an adaptive time-frequency analysis method used for analysing nonlinear and non-stationary signals (Goyal et al., 2020). Since the EMD decomposition process does not rely on any external basis functions, it reduces algorithmic decomposition errors caused by human factors, making it more advantageous when processing complex signals. Hu et al. (2019) first proposed EMD, which automatically decomposes signals into a set of IMF components. Yuan et al. (2023) utilised the Composite Multi-dimensional Multi-scale Iterative Fuzzy Entropy (CMvMPFE) method to extract fault features from rolling bearings and combined it with Support Vector Machines (SVM) for fault diagnosis analysis. Shi et al. (2024) combined CEEMDAN with an improved CNN network to construct a two-dimensional grey-scale image of the signal for feature extraction, followed by SVM recognition and classification. Li et al. (2019) addressed the issue of selecting sensitive IMFs in EEMD, proposing a method combining EEMD and improved band-based entropy for rolling bearing fault feature extraction, which was successfully applied to the analysis of both simulated and actual rolling bearings. Chen et al. (2021) aimed to improve the recognition accuracy of nonlinear non-stationary vibration signals in rolling bearings, proposed a new fault diagnosis method based on wavelet threshold denoising, complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) energy entropy, and particle swarm optimisation least squares support vector machine (PSO-LSSVM).

Signal processing methods are used to remove interference from signals, amplify useful fault features, and thereby achieve fault diagnosis. However, traditional fault extraction methods generally require manual intervention. With the rapid

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

mechanical sciences
DISCUSSIONS

development of artificial intelligence technology, intelligent diagnosis of mechanical equipment has garnered increasing attention. Deep reinforcement learning is a powerful algorithm that combines the feature extraction capabilities of deep learning with the decision-making and classification abilities of reinforcement learning. Mnih et al. (2015) first applied the 'Deep Q-Network' (DQN) algorithm to address pixel-based problems, achieving validation on Atari video games. Subsequently, due to the overestimation issue in DQN algorithm diagnostic results, van Hasselt et al. (2016) proposed a special modification to the DQN algorithm, demonstrating that the resulting algorithm not only reduces the observed overestimation but also improves model performance. Cobbe et al. (2018) addressed the overfitting problem caused by using the same environment for training and testing in the most common benchmarks in RL, by constructing different training and testing sets using procedurally generated environments, enabling the agent to adapt to an astonishingly large training set. Li et al. (2023) proposed a short-term wind power forecasting scheme combining federated learning and deep reinforcement learning (DRL). Mansour et al. (2021) proposed an anomaly detection and classification method using Faster RCNN with a deep reinforcement learning model, capable of automatically determining the presence of anomalies in a short time.

Although the above methods perform well in terms of identification accuracy, they still have the following problems: in complex production environments, noise signals are inevitable, which significantly reduces the diagnostic accuracy and robustness of the model. When used for fault diagnosis, the training environment and testing environment are the same, and the training time is long. Traditional deep neural networks often face problems such as local minima and overfitting, resulting in low accuracy and poor stability after model training.

To address this issue, this paper proposes a fault diagnosis method that combines adaptive white noise complete empirical mode decomposition with deep Q-learning, effectively solving the challenge of identifying non-linear, non-stationary vibration signals from wind turbine bearings. First, adaptive white noise complete empirical mode decomposition technology is used to extract feature information from the time-domain signal. The entropy features of the extracted samples are treated as the current state of the agent and input into the deep Q-network to further extract state features. Simultaneously, the Q-learning algorithm is employed for policy learning, and the environment is evaluated based on the actions taken by the agent. Through continuous interaction between the agent and the environment, the method enhances decision-making capabilities by trial and error and strategy updates, ultimately achieving the optimal strategy for fault diagnosis. The method framework is shown in Fig. 1. Experimental results indicate that the proposed method not only significantly improves the model's generalisation capability but also increases diagnostic efficiency by over 10%, providing reliable technical support for real-time fault diagnosis in industrial scenarios.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

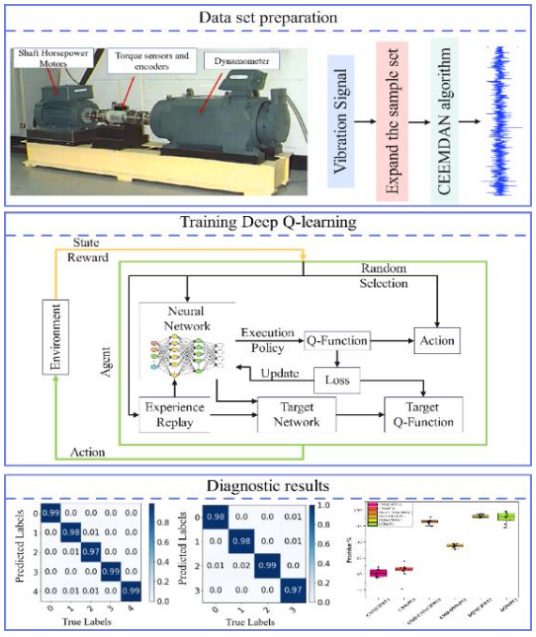


Figure 1: Framework of the proposed method.

2 Algorithm principles

2.1 CEEMDAN method and sample entropy principle

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) addresses the residual noise and computational efficiency issues of EEMD while retaining its advantages, through staged adaptive noise and residual recursive decomposition. This significantly improves modal purity and computational efficiency, making it an important tool for nonlinear signal analysis. This algorithm is based on the EMD and EEMD algorithms. During each IMF decomposition stage, it dynamically adjusts the amplitude of the added positive and negative noise pairs to avoid residual noise. To ensure signal energy conservation, it uses recursive decomposition of residuals to generate IMF at each stage. Additionally, it reduces the total average number of times the EEMD algorithm is applied, enabling the reconstruction error to approach zero under a limited number of integrations, thereby significantly improving decomposition accuracy (Liu and Ko, 2021).

CEEMDAN repeatedly adds noise and performs decomposition multiple times. Each instance of white noise added is paired, and its amplitude gradually decreases with increasing iteration counts. Gaussian white noise $\beta_0 w^{(k)}(t)$ is added to the signal $x(t)$ to be decomposed, yielding the new signal $x^{(k)}(t)$. The new signal is then subjected to EMD decomposition to obtain the first IMF. then add paired positive and negative Gaussian white noise, perform EMD decomposition again, and continue until the residual signal $r_j(t)$ becomes a monotonic function and no further decomposition is required. At this point, the noise residue obtained from the decomposition is almost negligible. The following are the formulas for adding noise and the final decomposition of the original signal:

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences DISCUSSIONS

$$x^{(k)}(t)=x(t)+\beta_0 w^{(k)}(t) \tag{1}$$

$$x(t)=\sum_{j=1}^{J} IMF_j(t)+r_J(t) \tag{2}$$

Where in, β_k is the noise amplitude coefficient, β_0 is the initial noise coefficient, which is 0.2 times the standard deviation of the signal, k is the kth set (k = 1,2,3…N), N is the number of sets, j is the jth-order IMF component, and J is the number of sets.

Entropy can be used as an indicator to assess the uncertainty of time series in bearing fault diagnosis (Heng et al., 2024). There are various methods for calculating entropy to assess signal uncertainty and complexity, but considering computational efficiency and the accuracy of complexity estimation, this paper chooses sample entropy for analysis. The following introduces sample entropy:

Construct an embedding vector for the signal sequence $X = \{X_1, X_2, \cdots, X_N\}$, construct an embedding vector with embedding dimension m, convert the signal into m and m+1 dimensional vectors, and calculate the sample entropy E_{sam} based on the average similarity probabilities $B^m(x_r)$ and $B^{m+1}(x_r)$ of vectors X_i^m and X_i^{m+1}.

$$\begin{cases} X_i^m=(x_i,x_{i+1},\ldots,x_{i+m-1}), & i=1,2,\ldots,N-m+1 \\ X_i^{m+1}=(x_i,x_{i+1},\ldots,x_{i+m}), & i=1,2,\ldots,N-m \end{cases} \tag{3}$$

$$E_{sam}(m,x_r,N)=-\ln\left(\frac{B^{m+1}(x_r)}{B^m(x_r)}\right) \tag{4}$$

Where in, if $B^{m+1}(x_r)$ is equal to $B^m(x_r)$, then the entropy value tends to 0, and the signal is highly periodic. Conversely, when the entropy value is large, the signal is complex.

In summary, the calculation of sample entropy requires the values of the vector embedding dimension m and the standard deviation parameter x_r. For given values of m and x_r, the smaller the value of E_{sam}, the higher the similarity of the given time series. In this paper, the parameter value x_r is selected as 0.1 times and 0.2 times the standard deviation of the data, respectively, and the embedding dimension m is set to 1 and 2, respectively.

2.2 Deep reinforcement learning (DQN)

Mnih et al. (2015) combined deep learning for sensory processing with reinforcement learning to develop the 'Deep Q-Network' algorithm, which achieves human-level performance in many Atari video games that use raw pixel inputs. When combined with deep learning techniques, reinforcement learning methods have outperformed human players in Atari games and achieved significant accomplishments in fields such as autonomous vehicles and drones. Reinforcement learning addresses Markovian problems through the Bellman equation, and DQN essentially solves the optimal Bellman equation. Generally, reinforcement learning algorithms consist of five components: agent, environment, state, action, and reward. The goal of learning is to identify action patterns that maximise rewards.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences DISCUSSIONS

Reinforcement learning achieves learning through interaction between the agent and the environment. The environment states (s) constitute the state space $S = \{s_1, s_2, \ldots, s_{si}\}$, and the agent's operations, referred to as actions (a), constitute the action space $A = \{a_1, a_2, \ldots, a_{ai}\}$. After an action is executed, the environment returns a new state s' and a reward signal (r). The reward function quantifies the task objective and guides the agent's behaviour through positive/negative rewards. The policy (π) controls decision-making, defining the mapping from states to actions, with the goal of maximising the discounted cumulative reward (G_t), as shown in Eqs. (5) and (6). This framework optimises the policy through reward and punishment feedback, ultimately completing the specified task.

$$\pi(a|s) = p(A_t = a|S_t = s) \tag{5}$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{6}$$

Where in, the policy π specifies the probability p of selecting action a in the current state s, i.e., $\pi(a|s)$. The cumulative reward G_t is defined as the sum of future rewards starting from time step t, typically calculated using a discount factor $\gamma \in [0,1]$, such that the influence of future rewards on the present gradually diminishes. At time t, the agent is in state S_t and takes action A_t according to policy π, resulting in the next state S_{t+1} and an immediate reward R_{t+1}. The iterative formula for the Q-learning algorithm in reinforcement learning is as follows:

$$q(s,a) \leftarrow q(s,a) + \eta \left[r' + \gamma \max_{a'} q(s',a') - q(s,a) \right] \tag{7}$$

Where in, $q(s,a)$ is the Q function; s' and a' are the next state and action; η is the learning rate, which is used to balance the speed of learning.

In the DQN algorithm, a deep neural network is used to fit the Q function, denoted as $q(s,a,\theta)$, where θ represents the parameters of the neural network. The DQN network is updated using interaction data stored in the memory replay unit (D). The agent uses the prioritised experience replay method to sample small batches of data from D and optimises the weights and biases of the DQN network using stochastic gradient descent to minimise the mean squared error loss function, as shown in Eq. (8).

$$L(\theta) = E\left[\left(R_{t+1} + \gamma \max_{a \in A(S_{t+1})} q(S_{t+1}, a, \theta') - q(s,a,\theta) \right)^2 \right] \tag{8}$$

Where $L(\theta)$ is the loss function. Use $L(\theta)$ to update the main network parameters θ to minimise the loss.

$$\theta \leftarrow \theta - \eta \nabla_\theta L(\theta) \tag{9}$$

In addition, reinforcement learning requires balancing exploration and exploitation, so the ε-greedy strategy formula is used when selecting actions, as follows:

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences DISCUSSIONS

$$a=\begin{cases} a_r, & \varepsilon_r<\varepsilon \\ \underset{a\in A(s)}{argmax}\, q(s,\ a), & \varepsilon_r>\varepsilon \end{cases} \tag{10}$$

In the equation, a_r represents a randomly selected action from the action space A(s), ε_r represents a randomly selected number between 0 and 1, and ε is the greed rate, which controls the extent to which the agent explores and exploits the environment ($0 < \varepsilon < 1$). This paper adopts a variable ε, with the agent exploring more in the early stages and exploiting more in the later stages. The formula for ε is given in Eq. (11), where ε_{max} is the maximum value of ε; ε_{min} is the minimum value of ε; and c is the decay rate of ε.

$$\varepsilon = \begin{cases} \varepsilon - (\varepsilon_{\max} - \varepsilon_{\min}) * c, & \varepsilon > \varepsilon_{\min} \\ \varepsilon_{\min}, & \varepsilon < \varepsilon_{\min} \end{cases} \tag{11}$$

Figure 2 shows a schematic diagram of the DQN algorithm. The agent consists of a deep Q-network, an objective network, and a replay memory unit. The agent can perceive abstract representations of data states through deep neural networks and learn strategies based on reward values from environmental feedback until it learns the optimal strategy. The specific process and logical structure of the DQN algorithm are shown in Table 1.

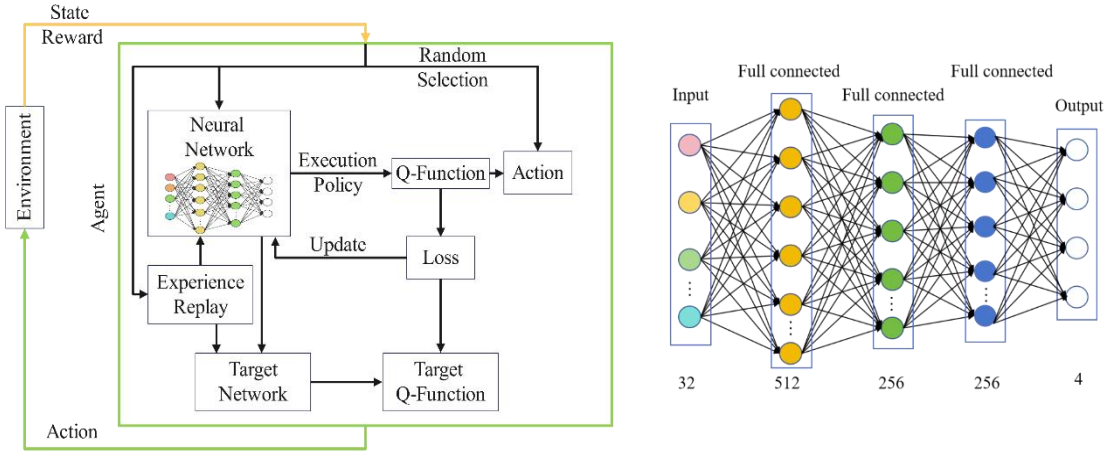


Figure 2: Diagram of the DQN algorithm: (a) structure of the DQN algorithm; (b) structure of the Q network.

To clarify the details of the algorithm implementation, Table 1 shows the complete pseudocode for the DQN algorithm. This pseudocode systematically presents the overall execution flow and core logic structure of the DQN algorithm, focusing on the algorithm initialisation stage, including the parameter settings for the Q network and target network; the application of the ε-greedy strategy in action selection; the storage and sampling mechanism of the experience replay buffer; and finally, a detailed explanation of the periodic update strategy for the target network parameters.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

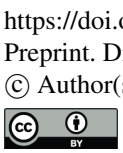
Table 1: DQN algorithm pseudocode.

State space S, action space A, discount rate γ, learning rate η, greed rate ε
Initialise experience pool D with capacity N
Initialise Q-network parameters θ
Initialise target Q-network parameters θ'
Initialise state s
Set training episodes episodes
For episode = 1, episodes
In state s, use the ε-greedy policy to obtain action a
The agent executes action a, and the data environment returns the reward value r and the next data state s' corresponding to the fault type.
In state s', use the ε-greedy strategy to obtain the next action a'
Store (s, a, r, s') in D
Randomly select B experience samples $(s_{dj}, a_{dj}, r_{dj}, s'_{dj})$, $dj = 1,2,\ldots,B,\ B < N$
Update the Q function: $q(s,a) \leftarrow q(s,a) + \eta\left[r' + \gamma \max_{a} q(s', a') - q(s,a)\right]$
Perform gradient *descent* on the loss function $L(\theta)$ to update the neural network parameters θ
Every C updates, update the target network parameters $\theta' = \theta$
Until $q(s, a, \theta)$ converges, terminate the state
End

3 Proposed algorithm

This study proposes a vibration signal fault diagnosis method based on deep reinforcement learning (DQN), which achieves intelligent diagnosis by constructing a gamified training environment using simulated vibration signals. In this framework, the intelligent agent selects a sample (e.g., $s_{si,sj}$) from the fault type sample set $S = \{s_{0,0}, s_{0,1}, \cdots, s_{si,sj}, \cdots, s_{SI-1,SJ-1}\}$. Each sample is represented by four groups of sample entropy features (totalling 4 × IMF component count) to form the state representation. The initial state for each round of the game is random. The agent's execution behaviour is associated with the category labels of the fault samples. The dimension of the action space is equivalent to the number of fault types, and each action directly corresponds to a fault type label $A = \{a_1 = 0, a_2 = 1, \cdots a_{ai} = SI - 1\}$, where ai is the number of actions available in a given state. The intelligent agent continuously executes actions and drives state transitions until the predefined termination condition $s_{si,SJ-1}$ is met, thereby completing the current training round. The state transition network shown in Fig. 3 (different coloured star marks represent fault types, and the connecting lines represent transition paths) intuitively illustrates this dynamic process. Through this policy-optimised DRL mechanism, the intelligent agent gradually learns the

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences DISCUSSIONS

optimal diagnostic strategy through continuous interaction, ultimately achieving autonomous extraction of vibration signal features and precise identification of fault types.

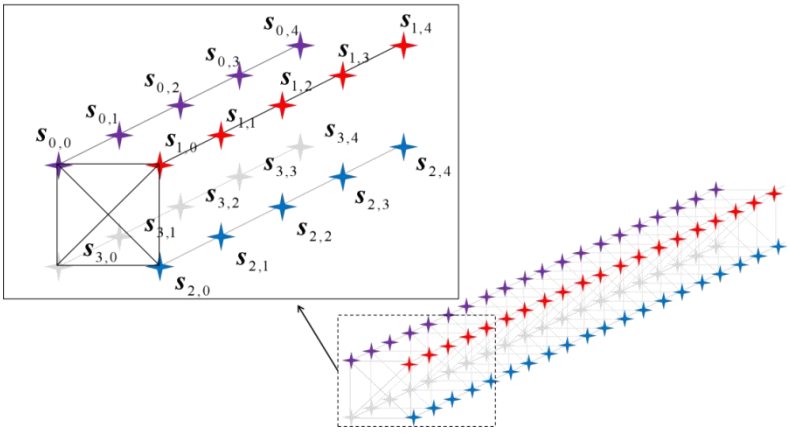


Figure 3: Reinforcement learning environment.

The improved reward mechanism is as follows: a fault type count vector Np is added to the intelligent agent to record the number of times each fault type sample is identified by the intelligent agent during its operation. ε_n is the difference between the maximum and minimum values of the count vector. The count formula is shown in Eq. (12), and the improved reward function is shown in Eq. (13).

$$\varepsilon_n = max(N_P) - min(N_P) + 1 \tag{12}$$

$$r = \begin{cases} \frac{1}{\log_{\xi}(\varepsilon_n + \alpha)}, & right \\ \&-\varepsilon_n \times \beta, & wrong \end{cases} \tag{13}$$

In this process, each time the intelligent agent performs a diagnosis, it records the change in the number of fault types of the diagnostic samples in $N_p = \{n_{p0}, n_{p1}, n_{p2}, n_{p3}, n_{p4}\}$. The condition for the reward value change is as follows: when the difference β between the count vectors is greater (i.e., the imbalance in sample training is greater, the fewer times the intelligent agent diagnoses a certain type of sample), the smaller the reward r given, where $\varepsilon_n \in (1, +\infty)$. The selection range of the ξ parameter is $\xi \in (1, +\infty)$. Here, $\xi = 20$ and $\alpha = 0.1$ are taken to control the value of r to be greater than 0, and β is the negative reward control parameter $\beta = 0.001$.

4 Experimental verification and discussion

The flowchart of the proposed fault diagnosis method is shown in Fig. 4. The main steps are as follows:

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

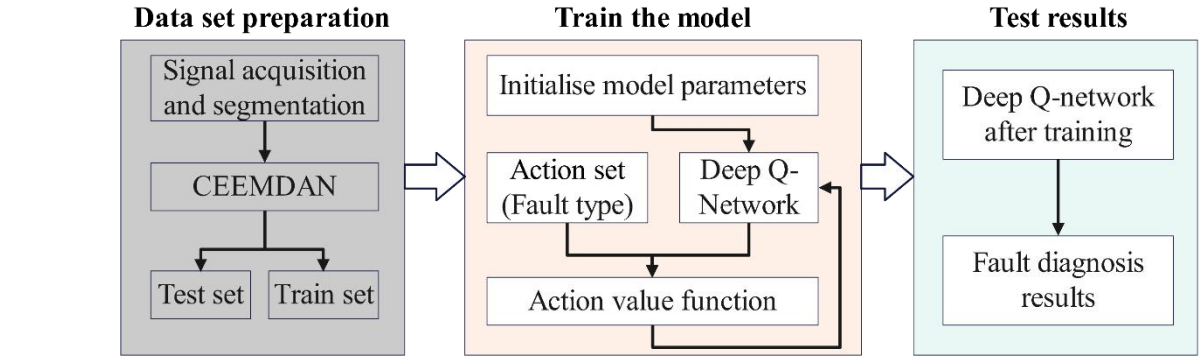


Figure 4: Flowchart of the proposed fault diagnosis method.

First, collect the data. The time-domain data collected is divided as follows: the dataset is divided into four fault types. The collected data is sampled at 1024 points per sample, with a sample overlap rate of 30%. To ensure an adequate sample size and enhance the model's robustness, 50% of the samples are randomly zeroed using the random zeroing method, resulting in an expanded sample set. Therefore, the total sample size is 800 × (1 + 50%) × 4 = 4800. Next, the data is processed using the CEEMDAN method to decompose each sample into 8 IMF components, and the sample entropy of each IMF is calculated. Additionally, to ensure complete feature information representation, different sample entropy parameters are used to calculate 4 distinct sample entropies for each IMF. Thus, after feature extraction, each sample yields a feature vector composed of 8 × 4 = 32 feature values. Finally, a deep Q-network is trained to obtain the optimal strategy, enabling the agent to make decisions based on the data state of a specific fault type, i.e., to determine the fault type. The test samples are then input into the trained deep Q-network to output the diagnostic results.

4.1 Experimental data

This study utilises experimental data from the bearing database published by Case Western Reserve University (CWRU) and the German Paderborn Bearing Dataset (PU). The experimental platform is shown in Fig. 5. The test bearings from CWRU are SKF6205 deep groove ball bearings, with the data including four fault states: normal bearing, roller fault, inner ring fault, outer ring failure. The sampling frequency of the vibration data is 12 kHz. The PU uses the 6203 deep groove ball bearing, with a vibration signal sampling frequency of 64 kHz. The data includes five types of fault states: undamaged (K002), artificial outer ring damage (KA06), artificial inner ring damage (KI07), real outer ring damage (KA16), and real inner ring damage (KI04) for training the DQN model, with the dataset partitioning shown in Table 2.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

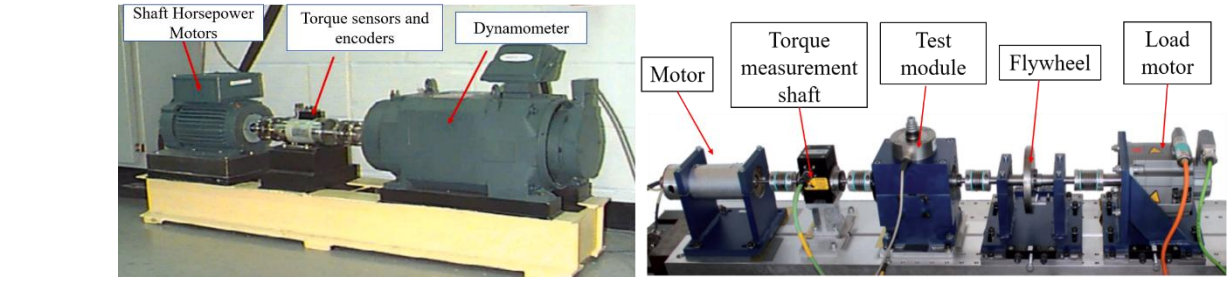


Figure 5: Experimental platforms: (a) CWRU; (b) PU.

Table 2: Division of the rolling bearing fault datasets.

CWRU dataset				PU dataset			
Fault type	Training set	Test set	Label	Fault type	Training set	Test set	Label
Normal	960	340	0	K002	1116	282	0
Inner-ring fault	960	340	1	KI04	1116	282	4
Roller fault	960	340	2	KI07	1116	282	2
Outer-ring fault	960	340	3	KA06	1116	282	1
				KA16	1116	282	3

4.2 Data processing

Based on the method proposed in Sect. 2, the signal-EMD library in the Python software platform is first used to extract features from each set of vibration data. CEEMDAN decomposition is then performed on the denoised rolling bearing fault vibration data. Taking the inner ring fault of a rolling bearing as an example, a total of seven modal components and one residual component are obtained, as shown in Fig. 6.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

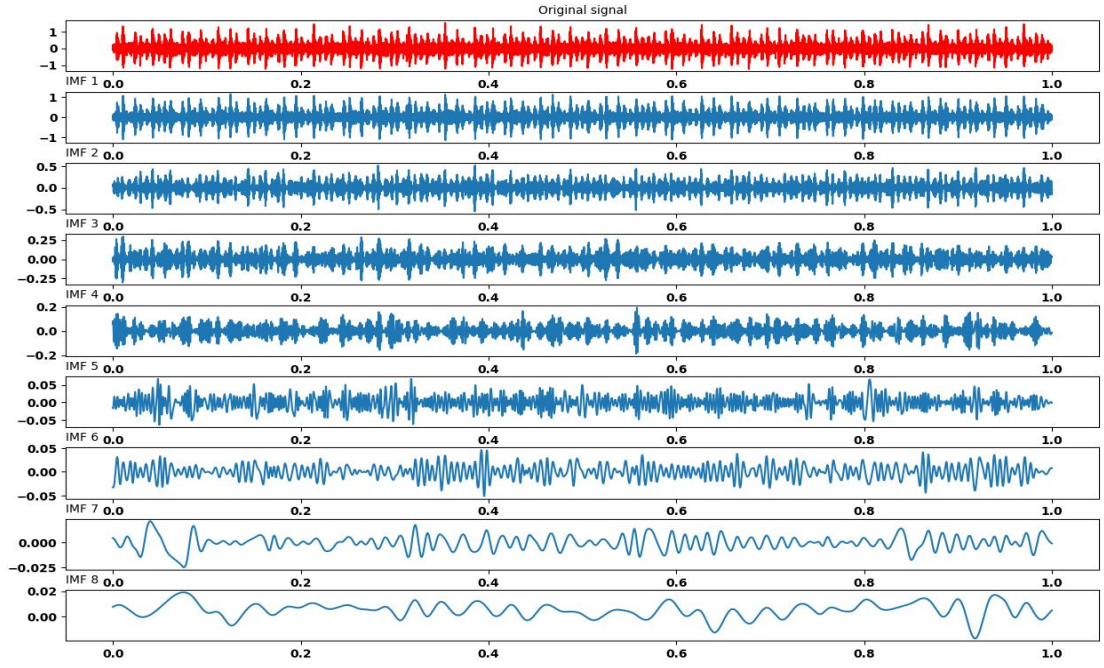
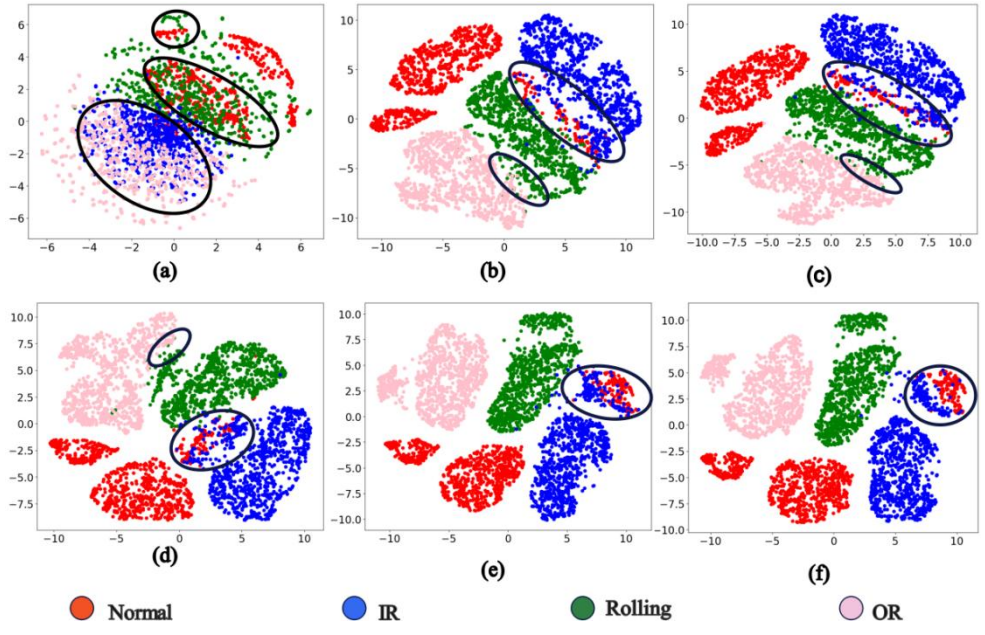


Figure 6: CEEMDAN decomposition results of the vibration signal for a rolling bearing inner-ring fault.



245

Figure 7: Comparison of t-SNE results obtained with (a) the original data and (b) 4, (c) 8, (d) 16, (e) 24, and (f) 32 sample entropies.

This study utilises the feature extraction capabilities of visualisation algorithms to demonstrate the advantages of using sample entropy as a feature vector. T-SNE is currently one of the most commonly used non-linear algorithms for

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

mechanical sciences DISCUSSIONS

dimensionality reduction in high-dimensional data. Fig. 7 shows the two principal components obtained by the T-SNE algorithm for different numbers of sample entropies. The circled areas in the figure represent sample mixing regions. The legend labels Normal, IR, Rolling, and OR correspond to four types: normal operating condition, inner ring fault, rolling element fault, and outer ring fault. The figure shows the impact of the number of sample entropies on feature representation. In Fig. 7, (a) the original data features are chaotic due to composite faults, with extensive overlapping regions that are extremely difficult to distinguish; (b) is the classification result using the sample entropies of the first four IMFs as features; (c) is eight sample entropies, i.e., one group of sample entropies; (d) shows two groups of sample entropies; (e) shows three groups of sample entropies; (f) shows four groups of sample entropies. Comparing (b) and (c), it can be seen that the classification results of the two figures are similar, indicating that the sample entropies after IMF 4 have little effect on classification, and IMF 5-8 are ineffective false components. When classifying using sample entropies with different numbers of groups as features, the features with more groups show more pronounced performance. In (f), the classification is clear and distinct, and from the figure, it can be seen that each category consists of two piles of different sizes, accurately reflecting the structure of the sample data. The larger pile represents the sample entropy extracted from the original data, while the smaller pile represents the sample entropy features obtained from the sample data expanded using the random zeroing method.

The prepared dataset is then used to construct the training environment for DQN and train the DQN model. Table 3 lists the parameters for the DQN method used for bearing fault diagnosis. The target model parameters are updated after every 10,000 iterations. In addition, the action exploration parameter is gradually adjusted from 1 to 0.017 during the first 30,000 iterations.

Table 3: Model parameters.

Hyperparameters	Symbols	Values
Learning rate	η	0.00105
Reward discount factor	γ	0.06482964
Batch size	B	128
Network hard update coefficient	T	10000
Decay rate	c	1/30000
Experience pool capacity	N	100000
Number of iterations	episodes	50000

Due to the lack of stability in the early stages of the DQN algorithm, the model was trained 10 times under different random numbers to estimate its performance. Fig. 8 shows the training results of the DQN model on the CWRU dataset. Fig. 8a shows the training reward curve of the DQN model. The solid line represents the mean value verified multiple times, and the shaded area represents the range of standard deviation. As can be seen from the figure, as the number of training rounds

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences DISCUSSIONS

increases, the cumulative reward obtained by the agent shows a decrease in reward values during the initial exploration phase, accompanied by significant fluctuations. After the 12th training round, the reward begins to rise, with fluctuations decreasing in the later stages and stabilising around 301. To verify the accuracy of the multiple training results, the model was trained 10 times with different random number settings for each training run. All 10 training runs began to converge at the 25th training round, and the subsequent convergence became stable. (b) shows the training loss curve for the DQN model. As shown in the figure, the maximum loss value is 10, and the loss curve eventually converges stably to around 0.001. The experimental results indicate that the mean of the loss function decreases gradually from an initial value of approximately 4.28 and eventually approaches 0. Although the training loss fluctuates between 0 and 3 during the training process, the training loss converges, and no overfitting occurs.

PU data is used for model training. Fig. 9a shows the reward curve for the DQN model training. As can be seen from the figure, as the number of training rounds increases, the cumulative reward obtained by the agent shows significant fluctuations in the early stages, but the overall trend is upward, with fluctuations decreasing and stabilising around 75 in the later stages. All 10 training sessions converged at the 300th training round, and the subsequent convergence tended to stabilise. (b) shows the loss curve of the DQN model training. As shown in the figure, the loss curve converges stably to around 0.001.

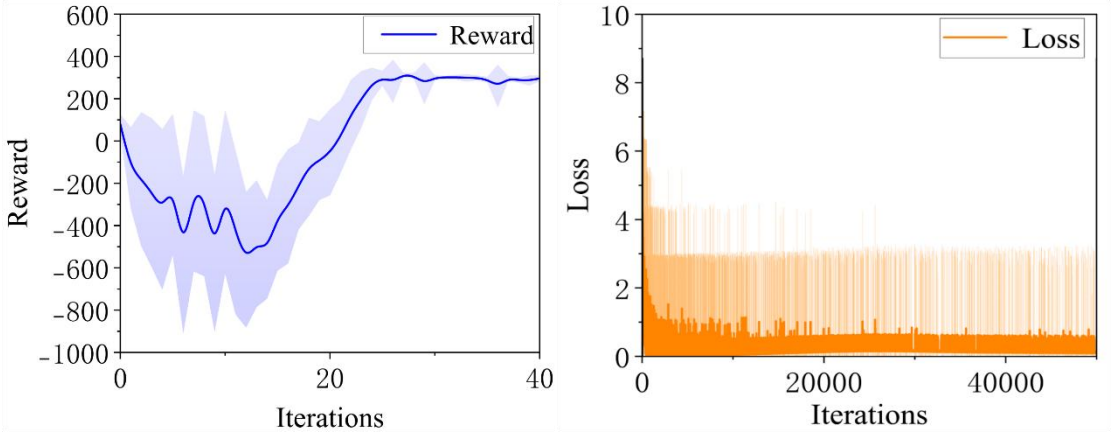


Figure 8: Training results of the DQN model on the CWRU dataset: (a) reward curve; (b) loss curve.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

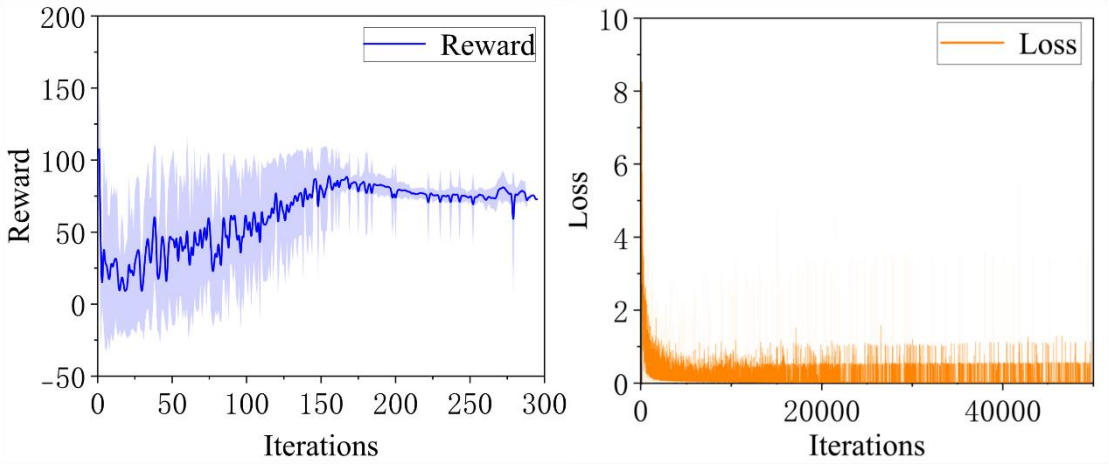


Figure 9: Training results of the DQN model on the PU dataset: (a) reward curve; (b) loss curve.

Furthermore, this paper compares the performance under different decay rates and learning rates to reveal the impact of decay rate and learning rate on the convergence characteristics of the model. The experiments were conducted with four decay rates (1/10,000, 1/20,000, 1/30,000, and 1/40,000) and four learning rates (0.1, 0.01, 0.001, and 0.00001), respectively, as shown in Fig. 10.

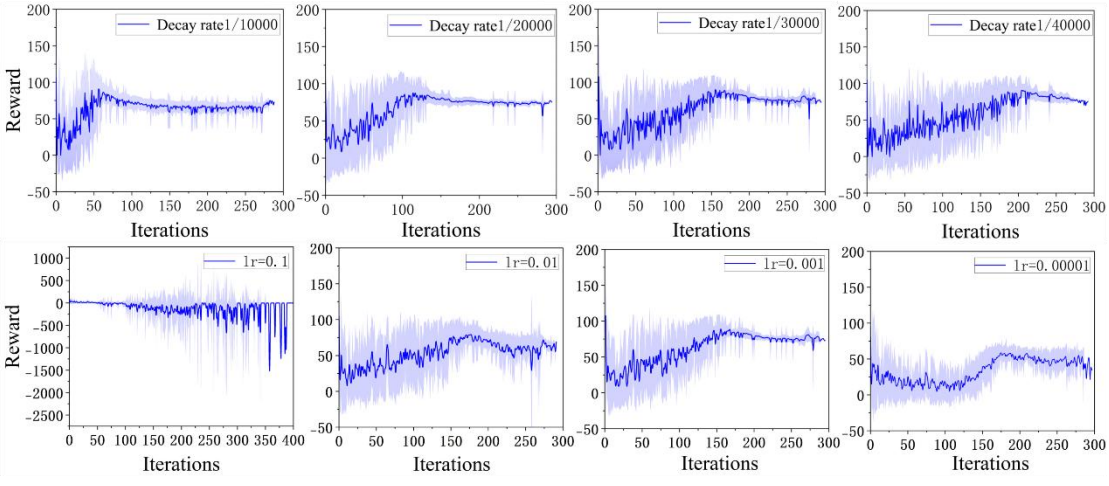


Figure 10: Comparison of different decay rates and learning rates.

Analysis of Fig. 10, Decay rate results analysis: When the decay rate is 1/10,000, the model score exhibits a typical three-stage change: a rapid rise to a peak of 90 points, followed by a slight decline, and finally stabilising at around 67 points. When the decay rate is reduced to 1/20,000, the initial rise slope decreases slightly (peak of 87 points), but the final convergence value improves to 74 points, demonstrating better stability; when the decay rate is further adjusted to 1/30,000, the model reproduces the rapid rise characteristic (peak of 90 points) while maintaining a high convergence level (75 points); when the decay rate is set to 1/40,000, although the initial rise is slightly weaker (peak of 86 points), it still achieved a

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

mechanical sciences
DISCUSSIONS

convergence value (74 points) similar to that of 1/30000. The experimental results indicate that when the decay rate is within the range of 1/20000 to 1/40000, the model can maintain good learning efficiency and achieve better final performance, with a decay rate of 1/30000 showing the best overall performance. Analysis of learning rate results: When the learning rate is 0.1, the model's reward value fluctuates sharply between -1500 and 1000, and shows a continuous downward trend as training progresses, indicating that this learning rate setting is too high, causing the model to fail to converge stably. In contrast, when the learning rate is set to 0.01 and 0.001, the model achieves effective convergence, with the reward value gradually increasing from the initial 0 points to around 90 points before stabilising. Among these, the final convergence score for a learning rate of 0.001 is 75 points, while the convergence score for a learning rate of 0.01 is 70 points. However, the latter exhibits approximately 5 points of fluctuation during convergence, indicating slightly lower stability compared to the former. When the learning rate is set to 0.00001, the overall fluctuation is smaller, with the reward curve first decreasing, then increasing, and finally converging slowly around a reward score of 51.

4.3 Algorithm comparison analysis

To verify the advantages of the proposed algorithm model, the algorithm in this paper is compared with the EMD+CNN and CNN algorithms.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

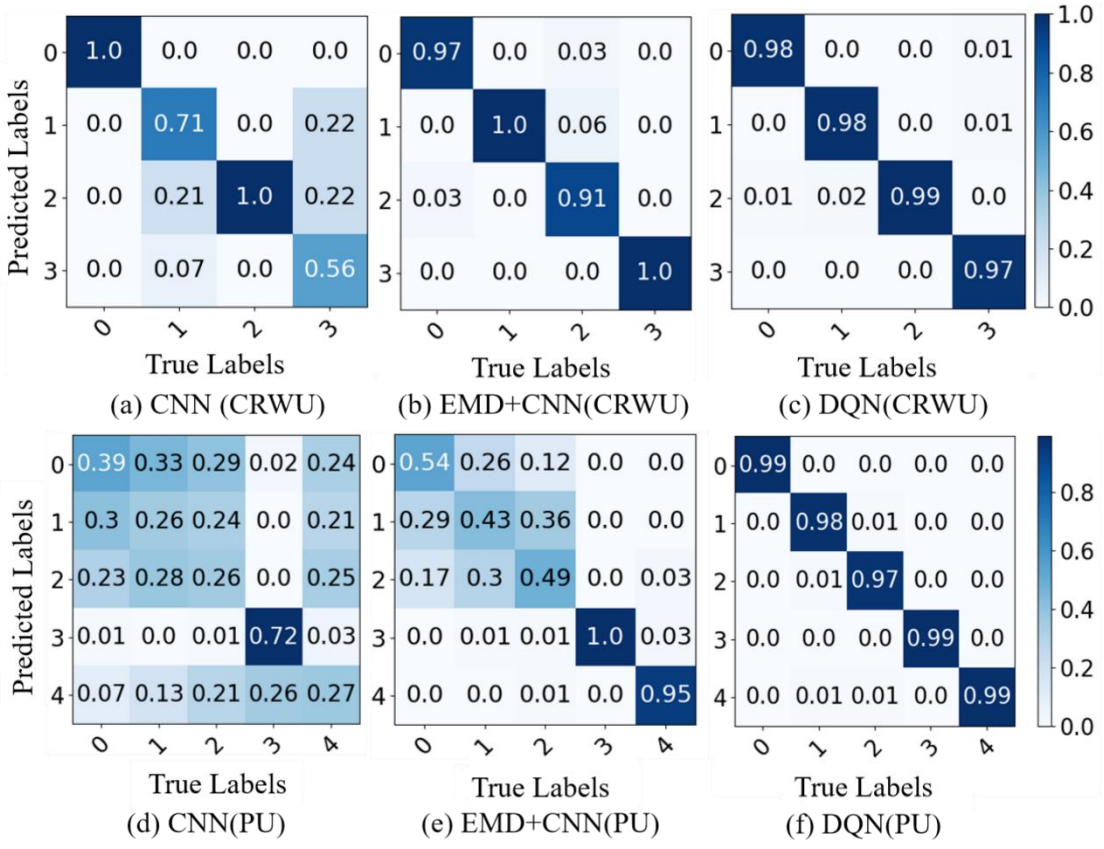


Figure 11: Confusion matrices of the different algorithms.

Figure 11 shows a comparison of the test multi-class confusion matrices for three different algorithms (DQN, EMD+CNN, and CNN) on the CWRU and PU datasets, verifying the classification accuracy and error rate of misclassification. The true labels of the test samples are considered the horizontal coordinates of the matrix, while the predicted labels are projected onto the vertical coordinates. The higher the prediction accuracy, the closer the fill colour of the box is to the upper half of the colour bar, i.e., the darker portion in the figure. The experimental results show significant differences in final accuracy among the three algorithms: (a) in the CNN algorithm achieves nearly 100% diagnostic accuracy for fault categories labelled as 0 and 2, but the accuracy for categories 1 and 3 is low, with confusion present, resulting in an overall diagnostic accuracy of 81.9%; (b) in the EMD+CNN algorithm achieves only 91% diagnostic accuracy for fault samples labelled as 2, with a final accuracy rate of 97%; (c) in the DQN algorithm achieves accuracy rates above 97% for all four fault types, but its final accuracy rate is 98.2%, approaching 1.0; (d) in the CNN algorithm achieves a diagnostic accuracy rate of nearly 99% for fault category 3, while the accuracy rates for fault categories 0 and 4 both reach 90%, Other fault types exhibit severe confusion, with an overall diagnostic accuracy rate of 81.7%; (e) in the EMD+CNN algorithm achieves a diagnostic accuracy rate of only 67% for fault samples labelled as 2, with a final overall accuracy rate of 89.1%; (f) in the DQN algorithm achieves accuracy rates exceeding 97% for all four fault types, but its final accuracy rate is 99.2%, approaching

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences DISCUSSIONS

the 1.0 level. A comprehensive comparative analysis reveals that the DQN algorithm has a significant advantage in terms of diagnostic accuracy.

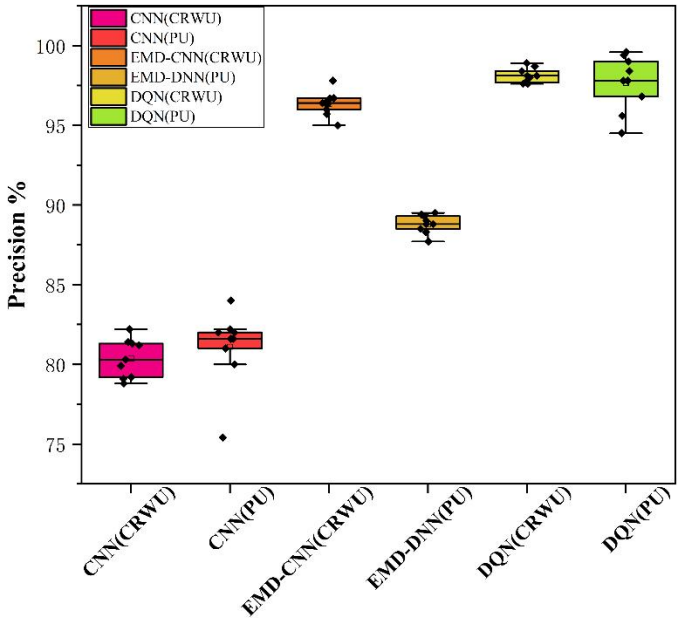


Figure 12: Comparison of the average prediction accuracy.

Figure 12 shows the average prediction accuracy of the model. When validated on two datasets, CWRU (bearing data released by Case Western Reserve University) and PU (bearing data from Paderborn, Germany), the proposed algorithm achieved higher accuracy compared to other methods. The method combining EMD and CNN was more accurate than CNN alone. This indicates that deep networks have stronger feature recognition capabilities than shallow networks. Additionally, on the CWRU dataset, DQN achieves a 1.6% higher accuracy rate than CNN under the same data features, indicating that DQN performs better than CNN in fault identification. Furthermore, on the PU dataset, DQN's fault identification accuracy is 10% higher than other algorithms, demonstrating that data processing and the proposed model contribute to improving diagnostic accuracy.

5 Conclusions

This paper proposes a fault diagnosis method that combines adaptive white noise complete empirical mode decomposition with deep Q-learning, effectively addressing the challenge of identifying non-linear, non-stationary vibration signals from wind turbine bearings. The method first uses adaptive white noise complete empirical mode decomposition technology to extract feature information from time-domain signals, and then inputs the extracted sample entropy features as the current state of the agent into the deep Q-network to further extract state features. Concurrently, the Q-learning algorithm is

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.

mechanical sciences DISCUSSIONS

employed for policy learning, and the actions taken by the agent are evaluated by the environment. Through continuous interaction and trial-and-error between the agent and the environment, the policy is updated to enhance decision-making capabilities, ultimately achieving the optimal strategy for fault diagnosis.

The experimental results show that the DQN algorithm demonstrates significant advantages in terms of diagnostic accuracy and robustness. On the CWRU dataset, the DQN algorithm achieves an accuracy rate exceeding 97% for the identification of all four fault categories, with an overall accuracy of 98.2%, representing a 16.3% improvement over traditional CNN methods. On the PU dataset, its performance further improves to 99.2%, outperforming the EMD+CNN method by over 10%. The DQN algorithm leverages the end-to-end decision optimisation mechanism of reinforcement learning to adaptively uncover deep correlations between fault features, achieving balanced and stable high-precision diagnosis across datasets. This significantly outperforms the shallow feature extraction mode of traditional CNNs and the manual feature fusion strategy of EMD+CNN, providing a superior solution for intelligent health management of complex industrial equipment.

Code and data availability

The vibration data analysed in this study are publicly available from the Case Western Reserve University (CWRU) Bearing Data Center and the Paderborn University (PU) Bearing Data Center. The CEEMDAN feature extraction and the DQN implementation used in this study are available from the corresponding author upon reasonable request.

Author contributions

R. W. conceived and designed the research, wrote the original manuscript. Z. Q. and C. L. supervised the whole work and revised the manuscript. Y. L. carried out the experiments and processed the experimental data. All authors read and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Financial support

This work was supported by the Talent Scientific Research Fund of Liaoning Petrochemical University (grant no. 2021XJJL-005), the Liaoning Provincial Natural Science Foundation Programme (grant no. 2022-BS-293), the Basic Scientific Research Project of the Liaoning Provincial Department of Education (grant no. LJKMZ20220718), and the Fushun Revitalization Talents Programme (grant no. FSYC202207005).

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

References

Chen, W., Li, J., Wang, Q., and Han, K.: Fault feature extraction and diagnosis of rolling bearings based on wavelet thresholding denoising with CEEMDAN energy entropy and PSO-LSSVM, Measurement, 172, 108901, https://doi.org/10.1016/j.measurement.2020.108901, 2021.

Chui, C. K. and He, W.: Spline manipulations for empirical mode decomposition (EMD) on bounded intervals and beyond, Appl. Comput. Harmon. Anal., 69, 101621, https://doi.org/10.1016/j.acha.2023.101621, 2024.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, D.: Quantifying generalization in reinforcement learning, arXiv [preprint], 2018.

de Azevedo, H. D. M., Araújo, A. M., and Bouchonneau, N.: A review of wind turbine bearing condition monitoring: state of the art and challenges, Renew. Sustain. Energy Rev., 56, 368–379, https://doi.org/10.1016/j.rser.2015.11.032, 2016.

Gao, Z. and Liu, X.: An overview on fault diagnosis, prognosis and resilient control for wind turbine systems, Processes, 9, 300, https://doi.org/10.3390/pr9020300, 2021.

Goyal, D., Choudhary, A., Pabla, B. S., and Dhami, S. S.: Support vector machines based non-contact fault diagnosis system for bearings, J. Intell. Manuf., 31, 1275–1289, https://doi.org/10.1007/s10845-019-01511-x, 2020.

Gupta, P. and Pradhan, M. K.: Fault detection analysis in rolling element bearing: a review, 2017.

Heng, L., Hao, C., and Nan, L. C.: Load forecasting method based on CEEMDAN and TCN-LSTM, PLoS One, 19, e0300496, https://doi.org/10.1371/journal.pone.0300496, 2024.

Hu, H., Zhang, L., Yan, H., Bai, Y., and Wang, P.: Denoising and baseline drift removal method of MEMS hydrophone signal based on VMD and wavelet threshold processing, IEEE Access, 7, 59913–59922, https://doi.org/10.1109/access.2019.2915612, 2019.

Kumar, A., Tang, H., Vashishtha, G., and Xiang, J.: Noise subtraction and marginal enhanced square envelope spectrum (MESES) for the identification of bearing defects in centrifugal and axial pump, Mech. Syst. Signal Process., 165, 108366, https://doi.org/10.1016/j.ymssp.2021.108366, 2022.

Kusiak, A. and Verma, A.: Analyzing bearing faults in wind turbines: a data-mining approach, Renew. Energy, 48, 110–116, https://doi.org/10.1016/j.renene.2012.04.020, 2012.

Li, H., Liu, T., Wu, X., and Chen, Q.: Application of EEMD and improved frequency band entropy in bearing fault feature extraction, ISA Trans., 88, 170–185, https://doi.org/10.1016/j.isatra.2018.12.002, 2019.

Li, Y., Wang, R., Li, Y., Zhang, M., and Long, C.: Wind power forecasting considering data privacy protection: a federated deep reinforcement learning approach, Appl. Energy, 329, 120291, https://doi.org/10.1016/j.apenergy.2022.120291, 2023.

https://doi.org/10.5194/ms-2026-195
Preprint. Discussion started: 23 November 2026
© Author(s) 2026. CC BY 4.0 License.



mechanical sciences
DISCUSSIONS

Liu, Y. Y. and Ko, Y. C.: Image processing method based on chaotic encryption and wavelet transform for planar design, Adv. Math. Phys., 2021, 7511245, https://doi.org/10.1155/2021/7511245, 2021.

Lou, X. and Loparo, K. A.: Bearing fault diagnosis based on wavelet transform and fuzzy inference, Mech. Syst. Signal Process., 18, 1077–1095, 2004.

Mansour, R. F., Escorcia-Gutierrez, J., Gamarra, M., Villanueva, J. A., and Leal, N.: Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model, Image Vis. Comput., 112, 104229, https://doi.org/10.1016/j.imavis.2021.104229, 2021.

Mnih, V. et al.: Human-level control through deep reinforcement learning, Nature, 518, 529–533, https://doi.org/10.1038/nature14236, 2015.

Qiao, W. and Lu, D.: A survey on wind turbine condition monitoring and fault diagnosis – Part II: signals and signal processing methods, IEEE Trans. Ind. Electron., 62, 6546–6557, https://doi.org/10.1109/tie.2015.2422394, 2015.

Saruhan, H., Saridemir, S., Çiçek, A., and Uygur, I.: Vibration analysis of rolling element bearings defects, J. Appl. Res. Technol., 12, 384–395, https://doi.org/10.1016/s1665-6423(14)71620-7, 2014.

Saxena, M., Bannet, O. O., Gupta, M., and Rajoria, R. P.: Bearing fault monitoring using CWT based vibration signature, Procedia Eng., 144, 234–241, 2016.

Shi, L., Liu, W., You, D., and Yang, S.: Rolling bearing fault diagnosis based on CEEMDAN and CNN-SVM, Appl. Sci., 14, 5847, https://doi.org/10.3390/app14135847, 2024.

Toma, R. N., Prosvirin, A. E., and Kim, J. M.: Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers, Sensors, 20, 1884, https://doi.org/10.3390/s20071884, 2020.

Toma, R. N., Kim, C. H., and Kim, J. M.: Bearing fault classification using ensemble empirical mode decomposition and convolutional neural network, Electronics, 10, 1248, https://doi.org/10.3390/electronics10111248, 2021.

van Hasselt, H., Guez, A., and Silver, D.: Deep reinforcement learning with double Q-learning, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Phoenix, Arizona, USA, 12–17 February 2016, AAAI Press, 2016.

Yuan, Q., Lv, M., Zhou, R., Liu, H., Liang, C., and Cheng, L.: Use of composite multivariate multiscale permutation fuzzy entropy to diagnose the faults of rolling bearing, Entropy, 25, 1049, https://doi.org/10.3390/e25071049, 2023.