Computer Science and Information Systems 21(4):1629–1649 https://doi.org/10.2298/CSIS240314051Z

A Novel Industrial Big Data Fusion Method Based on Q-learning and Cascade Classifier

Xi Zhang[1], Jiyue Wang[2,*], Ying Huang[3], and Feiyue Zhu[2]

[1] School of Mechanical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
zhangxi_9534@163.com

[2] School of Vehicle and Traffic Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
wangjiyue004@126.com

[3] School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
huangying_hy31@163.com

Abstract. The traditional industrial big data fusion algorithm has low efficiency and difficulty in processing high-dimensional data, this paper proposes a Q-learning-based cascade classifier model for industrial big data fusion. By combining cascade classifier and softmax classifier, feature extraction and data attribute classification of source industrial big data are completed in this cluster. In order to improve the classification rate, an improved Q-learning algorithm is proposed, which makes the improved algorithm randomly select actions in the early stage, and dynamically change in the late stage in the random selection of actions and actions with the highest reward value. It effectively improves the defects of traditional Q-learning algorithm that it is easy to fall into the local optimal and has slow convergence speed. The experimental results show that compared with other advanced fusion algorithms, the proposed method can greatly reduce the network energy consumption and effectively improve the efficiency and accuracy of data fusion under the same data volume.

Keywords: industrial big data fusion, Q-learning, cascade classifier, feature extraction.

1. Introduction

With the rapid development of industrial Internet of Things (IoT) technology, wireless sensor networks, as the core component of IoT sensing layer, have been widely used in various environmental monitoring fields [1-3]. In practice, most sensor nodes are powered by batteries, which leads to limited network resources. Due to the uneven geographical distribution of a large number of nodes, there is too much redundant information in the data [4], which increases energy consumption and transmission delay. In addition, due to the widespread interference in the application environment of the Internet of Things, it will directly weaken the data communication transmission capability and reduce the

* Corresponding Author

accuracy of data acquisition, affecting the overall performance of the Internet of Things system.

In order to solve the above problems, some fusion technologies for industrial big data came into being [5,6]. The main idea is to fuse data from multiple different source nodes to eliminate redundancy and reduce data transmission volume, so as to improve network performance, extend network life and reduce energy consumption. The traditional fusion algorithms are mostly based on BP neural network, SOFM and other shallow network models. These models are prone to problems such as over-fitting, model training trapping local minimum, slow convergence rate, etc., resulting in reduced algorithm efficiency, weak feature extraction and classification ability, and unable to effectively process high-dimensional data [7,8]. In view of the above defects, the application of deep model has become a new direction for the integration of industrial big data. In 2015, LeCun et al. [9] proposed the concept of ”deep learning”. Then, convolutional neural network (CNN) [10], stack auto-encoder (SAE) and other models have been widely used. Currently, deep learning has been studied in the field of data fusion. References [11-13] proposed to use the SAE model in deep learning to carry out feature fusion on industrial big data, significantly improving the accuracy of fusion. In reference [14], combining the deep model and sparse filtering algorithm, the BSSFM model was designed to extract data features, which improved the efficiency of fusion and reduced energy consumption. However, the use of SAE and other models would generate a large number of weight parameters in the training process, increasing the difficulty of model training.

Reference [15] studied a multi-sensor data fusion method in a complex environment. According to the results of trust, the idea of evidence iterative fusion was introduced to correct evidence conflicts from the level of evidence sources. Then it fused the modified evidence data, that is, it realized the fusion of multi-sensor data. Reference [16] proposed a multi-source sensor data fusion method based on manifold learning. T-SNE algorithm was adopted to reduce the dimension of high-dimensional multi-source sensor data, and the low-dimensional probability matrix was updated to generate a reasonable exclusion gradient between the points with small distance from the multi-source sensor data and completed the data fusion. Reference [17] proposed a multi-sensor data fusion method based on data correlation. MDF technology was used to identify hidden association features in group data, and the fusion of multi-sensor data was completed through the fusion of association features. Reference [18] proposed a multi-sensor data fusion method for UAV detection and tracking to measure the reliability of the data set. According to the reliability results, the Gaussian process model was constructed to complete the multi-sensor data fusion. However, the above methods have low efficiency in the process of fusion.

The multi-modal data fusion methods can be roughly divided into two categories: model-independent fusion methods and model-based fusion methods. Among them, the model independent method is simpler but less practical, and the fusion process is easy to produce losses. Model-based fusion method is more complex, but it has high accuracy and practicability, and is also the mainstream method used at present [19].

In the process of multimodal fusion, the time of fusion is an important consideration. According to different fusion periods or fusion levels, there are three model-independent fusion methods, each of which has its own characteristics [20]. In different experiments, different fusion methods can be tried to get better results. Some of the characteristics of the modes, such as the different data acquisition rates, present new challenges on how to synchronize the entire fusion process. The following is a detailed overview of the three fusion methods. Table 1 compares the three data fusion methods.

Table 1. Performance comparison of three model-independent fusion methods

Method	Information loss	Fusion difficulty	Fault tolerance	Fusion level	Fusion stage
Early fusion	Middle	Hard	Poor	Low	After feature extraction
Post fusion	Big	Middle	Middle	Middle	Post-decision
Hybrid fusion	Small	Easy	Good	High	Simultaneously

Early fusion, also known as feature fusion, is a fusion method that is carried out immediately after the feature extraction of the modes. The advantage of feature fusion is that the correlation between multiple features from different modes can be exploited early on, and is suitable for situations where the modes are highly correlated. For example, early fusion is used when combining audio and video features of speech recognition. However, it is difficult to extract features, so it is not the most ideal fusion method.

This method is difficult to represent the time synchronization cabinet between multimodal features. Since the representation, distribution, and density of various modes may vary, simple connections between properties may ignore the properties and correlations unique to each mode, and may create redundancy and data dependencies between the data. Features that need to be fused are also required to be represented in the same format before fusion. As the number of features increases, it is difficult to obtain cross-correlations between these features. Figure 1 shows an early fusion approach.

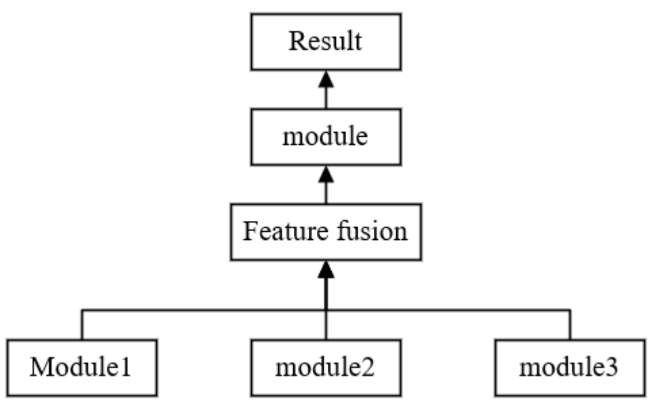

Fig. 1. Early fusion process

Late fusion, also known as decision-level fusion, refers to the fusion that occurs after a decision (classification or regression) has been made for each mode. For post-fusion, it is necessary to use corresponding models to train different modes, and then fuse the output results of these models. Compared with the earlier fusion cooperation, this fusion approach can handle simple data asynchronism. Another advantage is that it allows the use of methods that are best suited to analyzing each single mode, such as Hidden Markov Model (HMM) for audio and Support Vector Machines (SVM) for images.

However, the late fusion ignores the low-level interaction between multiple modes, and the fusion is difficult. Since different classifiers require different decisions, the learning process becomes both time-consuming and laborious. Figure 2 shows the structure of the late fusion method.

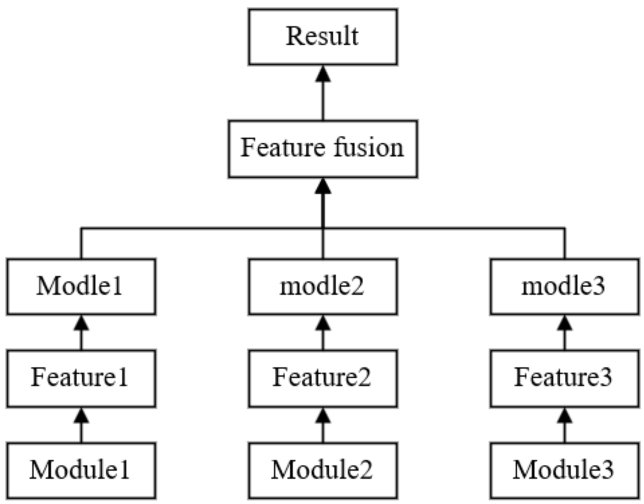

Fig. 2. Post fusion process

References [21,22] adopted fuzzy set, neural network, evidence theory and FIM algorithms for data fusion respectively, which did not consider the characteristics of sensor nodes and assumed that the collected data was correct. They did not judge the credibility of the source of the data, so the accuracy of the data was not guaranteed. Reference [23] proposed a fusion algorithm based on variable weights (VWFFA), which considered some influential factors in the communication process to weight the data. An improved fuzzy fusion algorithm was proposed in reference [24], which was weighted by considering the influence of external conditions on the calculation reliability. Both reduce redundancy and improve the accuracy of data, but they lack pertinence for different fusion stages, and the accuracy and real-time performance of data need to be further improved.

In order to improve the quality of industrial big data classification and fusion, this study proposes a new data fusion method from the perspective of deep learning.

2. Related Works

Model-based data fusion methods have a wider application range and better effect than model-independent methods, and current research is more inclined to such methods. Common methods include multi-core learning method, image model method and neural network method.

2.1. Multi-Kernel Learning (MKL)

The Multi-Kernel Learning (MKL) method is an extension of the kernel support vector machine (SVM) method and is the most commonly used method before deep learning, which allows different views of the data to be checked using different methods [25]. Because kernel can be regarded as similar function between data points, this method can better fuse heterogeneous data and use flexibly. Mirbeygi et al. [26] used MKL to sort the similarity of music artists from three aspects: acoustics, semantics and artists' social view, and proposed a new multi-core learning (MKL) algorithm, which could learn similar spatial items to generate similar spaces and combine all feature spaces into a unified embedded space in an optimal way. Figure 3 is the process of MKL.

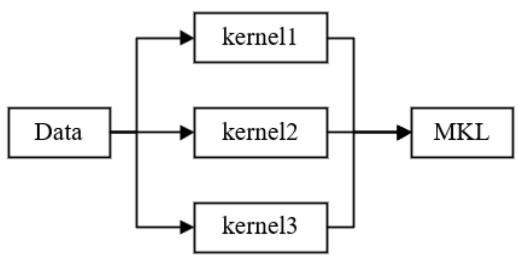

Fig. 3. Multi-kernel learning process

In some applications, there may be different information sources from different modes or results corresponding to different experimental methods, and each information source has one or more kernel questions of its own. The advantages of this method are flexible kernel selection, convex loss function (minimum value is the minimum value), and global optimal solution can be used to train the model and improve the model performance. Better MKL algorithms can be designed to improve accuracy, reduce complexity and training time.

Because in many applications, people propose many possible kernel functions, not choosing one of them but using them together, there is a lot of work in multi-kernel learning methods. The high time and space complexity is the main reason that the multi-core learning method can not be widely used. Another disadvantage is the large memory footprint and a slight dependence on training data.

2.2. Image Model Method

Image model method is also a common fusion method, which mainly fuses shallow or deep graphics through image segmentation, splicing and prediction operations, so as to obtain the final fusion result1.

Common image models are divided into generative (joint probability) models and discriminative (conditional probability) models. The use of image models in many studies, especially in statistical natural language processing, focuses on generative models that attempt to model the joint probability distribution of inputs and outputs.” In the early days, generative models were mainly used, such as dynamic Bayesian Networks (DynamicBayesian Networks) and hidden Markov models. In later studies, discriminative models were more popular, simpler and easier to learn than generative models. Common discrimination models, such as Conditional Random Field (CRF), classify and label the components of the image2. Table 2 compares the generative model with the discriminant model [27,28].

The main advantage of image models is that they are easy to explore the spatial and temporal structure of the data, and the interpretability of the model is enhanced by embedding expert knowledge into the model. The disadvantage is that there are complex dependencies between features, and the generalization of the model is not strong.

Table 2. Comparison of generation model and discrimination model

Comparison item	Generative model	Discrimination model
Characteristic	Finding the optimal classification surface between different categories	A posteriori probability modeling
Distinction	Model the joint distribution	Model the conditional distribution
Sample size	more	Less than the generative model
Accuracy rate	Low	Higher than the generation model

2.3. Neural Network Method

Neural network method is one of the most widely used methods at present. Long Short-Term Memory (LSTM) and Recurrent Neural networks (RNNS) are often used to fuse multi-modal information. For example, bidirectional long short-term memory network is used for multi-modal emotion recognition; multimodal Recurrent Neural Networks (m-RNN) are used to directly take the image representation, word vector and implicit vector as the input of multimodal judgment, which shows a good effect in image subtitle processing and other tasks [29].

Some researchers have achieved better results through model patchwork than multi-core learning and image models. The application of neural network method in multi-mode fusion has strong learning ability and good expansibility. The drawback is that as the number of modes increases, deep learning becomes less interpretable and needs to rely on large amounts of training data. Table 3 compares the three model-based fusion approaches.

Table 3. Performance comparison of three model-independent fusion methods

Method	workload	Model interpretability	Feature dependence	Applications
MKL	big	weak	simple	EasyMKL
Image-based	small	strong	Complex	Multimodal data classification
NNM	big	weak	simple	Multi-modal emotion recognition

Neural network structure optimization based on Genetic algorithm (GA) is one of the earliest meta-heuristic search algorithms for neural network structure search and optimization. In the early 2000s, an algorithm called Neural Evolution of Augmented Topology (NEIT), which also uses GAs to evolve increasingly complex neural network structures, received a lot of attention. Shinozaki et al. P applied GAs and covariance matrix evolution strategies to optimize the structure of DNN and parameterized the structure of DNN to a simple binary vector based on directed acyclic graph representation. Since the GA search space can be very large and each model in the search space is expensive to evaluate, parallel searches using large GPU clusters speed up the process. These neural network structure search and optimization techniques can be easily extended to multi-modal settings if a suitable representation of the network architecture is designed and it is not very expensive to train and test multiple architectures during the search process.

3. Q-learning Algorithm

The basic idea of reinforcement learning is to maximize the cumulative reward value obtained by the Agent from the environment in order to learn the optimal strategy to achieve the goal [30]. Reinforcement learning is a self-supervised learning method, which is different from supervised and unsupervised learning [31]. It does not require any data to be given in advance, but obtains learning information and updates model parameters by receiving environment's reward for action. The most important idea of reinforcement learning is continuous interaction with the environment, whose interaction elements include state, behavior and reward [32]. The process of reinforcement learning is shown in Figure 4. At time t, the Agent accepts the reward value R given by the environment and the state s at the next moment. At time $t+1$, the agent responds to the environment with an action a.

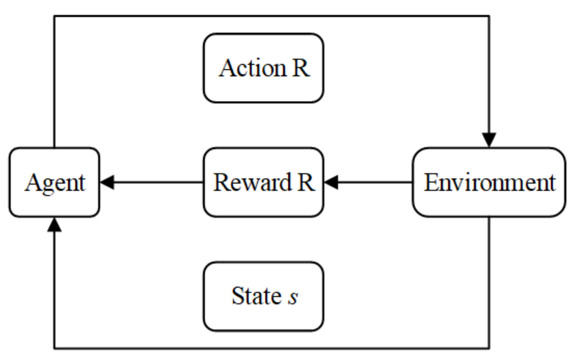

Fig. 4. Basic model of reinforcement learning

Reinforcement learning is divided into two types: model-based reinforcement learning and model-free reinforcement learning. In this paper, Q-learning algorithm in model-free reinforcement learning is selected to solve the problem and establish the algorithm model

[33]. In Q-learning algorithm, the environment will make the corresponding reward value R according to the action feedback of the Agent. The algorithm combines state s and action a together to construct Q-table and store the Q value of each update. It selects the action that can get the maximum reward according to the Q-table. After each action is selected, Q-table is updated. The Q-table update rules are as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma max Q(s',a') - Q(s,a)]. \quad (1)$$

Where s represents the current state. a represents the action selected from the current state. s' is the next state. a' represents the action selected from the next state. α represents the learning rate of control convergence ($0 < \alpha < 1$). γ represents the discount factor ($0 \leq \gamma < 1$), $\gamma = 0$ indicates immediate reward, and $\gamma = 1$ indicates future reward).

Where α stands for learning rate. If the learning rate is higher, the previous learning results will be retained less, and the Agent will more prefer the present and future results. As a discount factor, if $\gamma \to 0$, the Agent tends to choose the immediate reward; if $\gamma \to 1$, it means that the Agent is more far-sighted and tends to consider the influence of the subsequent reward. Algorithm 1 shows the update steps of Q-learning algorithm.

Algorithm 1 Power method

```
Input: Parameters;
Output: State;
Initialize Q(s, a), s ∈ S, a ∈ A;
for each episode
Initialize current-s
while not finished (all states) do
    At a certain time t, an optional action a is selected according to the current state s through
    the greedy dream strategy;
    Perform action a to obtain the state s' at time t + 1 and the reward R.
    Update Q-table;
    Compute formula (1);
    Record next state s' and convert it to the current state;
end while
```

Q-learning algorithm selects actions in two ways. One is exploration, which randomly explores an action and a state to transform; The other is utilization, for all the actions that may be selected to search for the next state, generate the maximum reward value and the action corresponding to the next state value.

4. Data Fusion Strategy

In this paper, the deep learning algorithm is used to design a hierarchical automatic dimension reduction classifier to process low-dimensional data from multi-source sensors and obtain ideal feature extraction results [34]. The specific implementation process is detailed as follows. Data dimension reduction is the process of mapping data from high-dimensional feature space to low-dimensional feature space, which can avoid the interference of chaotic and invalid data and improve the quality and efficiency of computation.

The designed automatic dimension reduction classifier (ADRS) is a single hidden layer structure. The ADRS structure is shown in Figure 5.

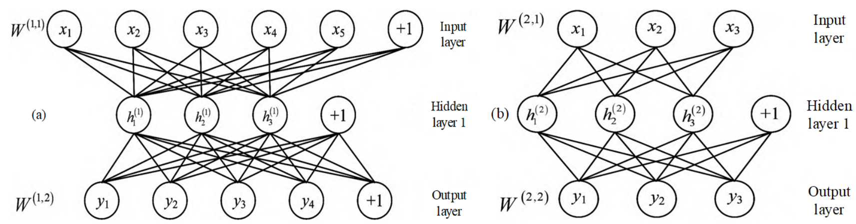

Fig. 5. ADRS structure. (a) k=1, (b) k=2

The input layer refers to the data input into the neural network. The input data must be numeric, meaning that non-numeric content needs to be converted to numeric value. Typically, data processing is the most time-consuming part of creating a machine learning model.

The hidden layer is composed of most of the neurons in the neural network and is the core part that processes the data to obtain the desired output. The data will pass through the hidden layer and undergo many weight and bias adjustments.

The output layer is the end product of processing data in a neural network and can represent different things. Typically, the output layer is made up of neurons, each representing an object.

The objective of ADRS dimension reduction classifier is to find the optimal fusion parameters in the sensor nodes (W, b). Fusion parameters determine the performance of data fusion. Selecting the optimal fusion parameters can improve the data fusion capability.

The global optimal solution of multi-source sensor network was found to optimize the initial weights and thresholds of the classification process. The essence of deep learning training is to update the weight. Before training a new model, it is necessary that each parameter has a corresponding initial weight. The selection of initial weight has certain influence on the placement of local minimum points and the improvement of network convergence speed. A neuron is a nonlinear unit with multiple inputs and single outputs. Only when the sum of inputs exceeds a certain value can the output respond, which is generally called the threshold value.

The input sensor node information x can achieve dimension reduction under the action of the output result y, so as to obtain the output result $a^{(k,2)}$ of the hidden layer. And $a^{(k,2)}$ is regarded as the characteristic result obtained after dimensionality reduction of input items for classification.

In the calculation process, to ensure the sparse robustness of features of $a^{(k,2)}$, three constraint functions should be used to achieve this, as shown in Equation (2):

$$minL(W, b_{argW,b}) = \frac{1}{m}\sum_{i=1}^{m}[x^i, y^i] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}[W_{ji}^{(k,l)}]^2 + \beta\sum_{j=1}^{S_2}KL(\rho/\bar{\rho}_j). \quad (2)$$

Where $L(W, b)$ represents the node energy loss function. ρ, λ and β represent the weight coefficient value of the classifier. KL represents constant constraint term in sparsity condition, $KL = 0.5||y^{(i)} - x^{(i)}||^2$. $x^{(i)}$ and $y^{(i)}$ represent constraint constant of input layer and output layer respectively. m indicates the number of samples in the ADRS. n_l indicates the number of network layers of the sensor. s_l represents the number of units contained in layer l. $W_{ji}^{(k,l)}$ represents the weight of connection between layer l and unit j in layer $l+1$ in the $k-th$ ADRS network. $W^{(k,l)}$ represents cell bias at each level in the sensor network.

In order to minimize the value of loss function, parameters in sensor network are trained by gradient descent algorithm to achieve dimension reduction. The specific implementation process is detailed as follows:

- Step 1. For all layers $l - l$, the element bias variable parameter satisfies $\Delta W^{(k,l)} = 0$, $\Delta b^{(k,l)} = 0$.
- Step 2. Set $i \in [1, m]$. The calculation formula is as follows:

$$\Delta W^{(k,l)} = \Delta W^{(k,l)} + \nabla_{W^{(k,l)}} \delta^{(k,l+1)} (\alpha^{(k,l)})^T. \tag{3}$$

$$\Delta b^{(k,l)} = \Delta b^{(k,l)} + \nabla_{b^{(k,l)}} \delta^{(k,l+1)}. \tag{4}$$

  Where $\alpha^{(k,l)}$ represents the loss parameter of unit n in layer l. $\delta^{(k,l+1)}$ represents the residual of element n in layer $l+1$. T stands for loss constraint.
- Step 3. Update the parameter values in ADRS.

$$W^{(k,l)} = W^{(k,l)} - a[\frac{1}{m}\Delta W^{(k,l)}] + \lambda W^{(k,l)}. \tag{5}$$

$$b^{(k,l)} = b^{(k,l)} - a[\frac{1}{m}\Delta b^{(k,l)}]. \tag{6}$$

- Step 4. Return to step 2, loop calculation, until meet preset conditions, then output $(W^{(k,1)}, b^{(k,1)}, W^{(k,2)}, b^{(k,2)})$.

After the above dimension reduction, the classifier is designed. Cascade automatic classifier ADRS is obtained by several dimension reduction processors through the cascaded action. ADRS has an obvious advantage that the output result of the input item can only be obtained after several dimensional reduction processing, eliminating most redundant information. The parameter values of each layer in ADRS can be calculated by the greedy floor algorithm, even if the output results of the upper layer are used as the input items of the next layer. The specific implementation process is as follows.

Set the number of hidden layers in ADRS to N_k. After the input data x is processed by the greedy algorithm, the parameters and output result $a^{(1,2)}$ are obtained, as shown in Figure 5(a). Then $a^{(1,2)}$ is taken as the input item, and the output result $a^{(2,2)}$ is obtained after the next round of greedy algorithm [35]. The process is shown in Figure 5(b).

Repeat training process. For the ADRS with N_k layers, after a greedy training, parameter set $(W^{(k,1)}, b^{(k,1)})|k = 1, \cdots, N_k$ is obtained. $(W^{(k,1)}, b^{(k,1)})$ is regarded as the connection weight value between ADRS levels. The multi-source sensor coding classifier is shown in Figure 6.

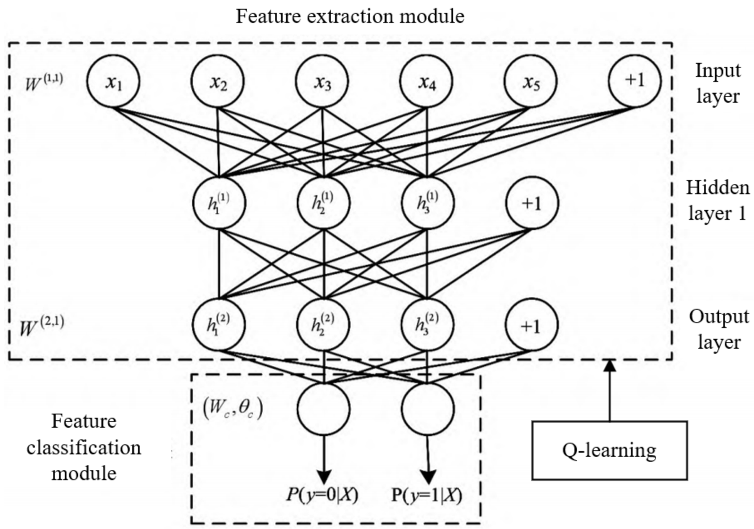

Fig. 6. Multi-source sensor feature classification model

The result of feature extraction plays a decisive role in the accuracy of subsequent data fusion and is the key to the successful completion of data fusion. Therefore, ADRS and Softmax classifier are combined to realize feature classification through cascading. The implementation process is as follows:

- Step 1. By training industrial big data samples to get parameter $(W^{(k,1)}, b^{(k,1)})|k = 1, \cdots, N_k$.
- Step 2. Through Q-learning processing of sample features, the best data features are obtained.
- Step 3. After training, data features are obtained. After supervised training and the function of Softmax classifier, output parameters (W_c, θ_c) are obtained.
- Step 4. Consider the parameters obtained in steps 2 and 3 as the original data. In the ADRS model, the supervised training algorithm is used to adjust the original data. Then, the adjusted sensor network and Softmax classifier [36] are used as the extraction module and classification module in the feature extraction model to complete the feature extraction and classification of multi-source sensor data.

In the process of solving problems, Q-learning algorithm generally uses the greedy strategy to make selection. This strategy selects the corresponding action with the maximum reward value in each selection, which is called greedy. However, it does not take the global consideration into account. In the initialization process, the Q value is set as the equal value or random value, that is, the learning is carried out in the environment without prior knowledge. At the same time, the learning rate is generally not dynamically adjusted after initialization, and the learning process is characterized by large decision-making space, slow learning speed and uncertain learning effect [37]. As a result, the result is easy to fall into local optimal and the optimal solution cannot be found. In this paper, Q-learning algorithm is improved, and a "Sigmoid "function is introduced as a dynamic selection strategy, so that the improved Q-learning algorithm can randomly select actions in the early stage and dynamically select the optimal actions in the late stage. This improved method can improve the phenomenon that the greedy strategy falls into the local optimal in the early stage and may choose the sub-optimal action in the late stage of learning.

This paper introduces "Sigmoid" as an action selection strategy, and its expression is as follows:

$$P(s_t, a) = \frac{exp(Q(s_t, a))}{1 + exp(Q(s_t, a))}. \tag{7}$$

According to Equation (7), when Q value of the initial state is set to 0, each action has the same selection probability and is more random in the early stage of selection. When the learning reaches a certain level, the reward value of each action starts to stand out, and the probability P changes with the update of Q value. The system will gradually tend to choose the action with high reward value.

In this paper, three kinds of selection strategies are discussed, which are "completely random" strategy, "semi-random" strategy and greedy strategy. We introduce parameter B and have the following options.

1. When $B = 0$, then $P > B$. Implement the greedy dream strategy.
2. When $B = 1$, then $P < B$. Implement the "completely random" strategy, that is, the actions are randomly selected throughout the selection process.
3. When $B \in (0, 1)$. Implement the "semi-random" strategy, that is, a "Sigmoid "function is introduced as a dynamic selection strategy. When $P \geq B$, the algorithm carries out the exploration strategy and selects the action with the maximum current reward value. When $P < B$, the algorithm adopts a "completely random" strategy and selects actions at random.
4. Combined with the improvement objectives of this paper, the strategy selection when $B \in (0, 1)$ is implemented, and $B = 0.5$ is set through a large number of experiments.

5. Experiments and Analysis

Matlab platform is used to simulate and analyze the data fusion algorithm, and the facility industry monitoring system is taken as the application scenario. In order to highlight the performance of the proposed data fusion, STDF [38], CCA [39] and FKCI [40] are used for comparative analysis.

The simulation parameters are shown in Table 4, where the number of nodes and the network scope refer to the random distribution of 100 sensor nodes in the perception area within the range of $100m \times 100m$. In order to compare the efficiency of various data fusion algorithms, the un-optimized LEACH protocol is adopted, and the energy consumption of sending, receiving and fusion data of nodes is calculated according to the first type of wireless communication energy consumption model.

The classification error rate of feature extraction with each algorithm is shown in Table 5. Where n, d and c represent the network layer number, the dimension of input data and the number of data classification respectively. As can be seen from Table 5, for low-dimensional and low-category data, the classification error rate of proposed method is

Table 4. Experiments parameters

Parameter	Value	Parameter	Value
Network range	$100m \times 100m$	Packet length	2000bit
Note number	100	Cluster message length	100bit
Sink node coordinate	(50,50)	Packet head length	100bit
Initial node energy	0.5J	Maximum simulation number	2000
Convolution kernel size	3×3	Convolution step	1

basically the same as that of STDF and CCA. With the increase of the input data dimension, the number of parameters of CCA and FKCI increases sharply, and the performance begins to decline, resulting in a significant increase in the error rate. In contrast, the error rate of proposed method based on the depth model is always at a low level.

Table 5. Inaccuracy of feature extraction and classification with different algorithms%

Method	d=200, c=4	d=600, c=6	d=900, c=12	d=2500, c=12
STDF(n=4)	8.4	16.7	19.6	28.1
STDF(n=5)	8.1	14.2	16.2	25.9
CCA	7.8	13.7	15.9	24.6
FKCI	7.1	12.5	13.4	20.2
Proposed	6.4	8.2	8.9	10.5

The average time of feature extraction and classification process of each algorithm is shown in Table 6. The analysis shows that Proposed has higher execution efficiency and can achieve faster data fusion compared with STDF, CCA and FKCI. Based on the powerful dimension reduction ability of CNN, as the dimension of input data increases, Compared with the other two algorithms, the feature extraction time of Proposed is greatly shortened, showing the advantages of processing high-dimensional data.

Table 6. Average time of feature extraction and classification with different algorithms/ms

Method	d=200, c=4	d=600, c=6	d=900, c=12	d=2500, c=12
STDF(n=4)	4.71	7.59	9.12	38.37
STDF(n=5)	5.49	8.04	9.58	40.96
CCA	5.22	7.54	8.69	36.54
FKCI	4.27	6.93	8.44	32.82
Proposed	3.18	3.26	3.71	9.50

This paper considers the effect of different number of nodes on the performance of wireless sensor networks in the $200m \times 200m$ region. Assuming that the flow rate is 20 packets/s, the average accuracy rate (AAR), average delay time (ADT) and average energy consumption (AEC) of the data within 10s are calculated and normalized, and the result is shown in Figure 7 and table 7.

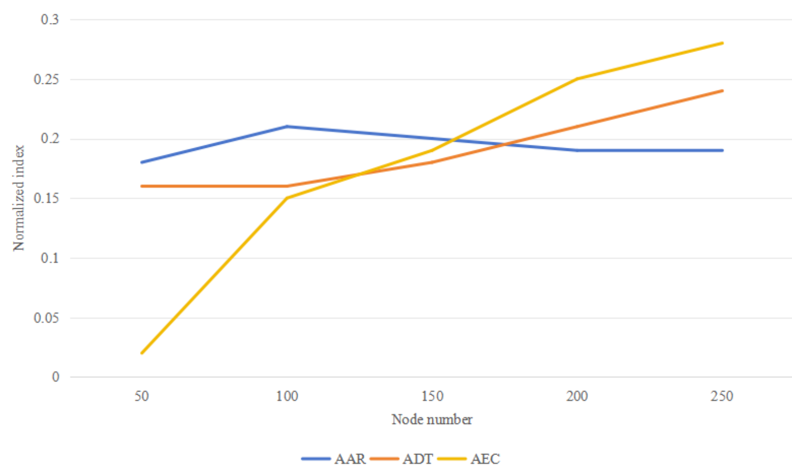

Fig. 7. Effect of node number on algorithm performance

Table 7. Effect of node number on algorithm performance

Node number	AAR	ADT	AEC
50	0.18	0.16	0.02
100	0.21	0.16	0.15
150	0.2	0.18	0.19
200	0.19	0.21	0.25
250	0.19	0.24	0.28

In a fixed area, AAR, ADT and AEC increase with the increase of nodes. When the number of nodes is increased from 100 to 150, the increase rate of data accuracy decreases, while the time delay and energy consumption increase significantly. Therefore, this paper selects 100 nodes as reference values for subsequent experiments.

MTS420/400 is used as the sensor plate model, the sensor plate includes humidity, temperature, PH, carbon dioxide concentration, smoke concentration and light sensor as a communication component, and the random high-frequency noise signal is added to the sensor to simulate the effect of the actual sensor noise. The parameters of the MTS420/400 are shown in Table 8.

Table 8. Effect of node number on algorithm performance

Parameter	MTS420/400
Temperature range/($^{\circ}$C)	-3941.3
Humidity range	0100
Signal-to-noise ratio/dB	01

This paper compares these four algorithms by counting the accuracy rate of receiving data from base station. Figure 8 shows the relationship between the accuracy rate of receiving data and the communication time of the base station. As can be seen from figure 8 and table 9, when FKCI, CCA, STDF and proposed method (node=25) are adopted for data fusion, the AAR values are 83.5%, 79.2%, 68.8% and 88.7%, respectively. Both the STDF and proposed method exclude unreliable data, while the CCA algorithm does not distinguish whether the data is correct or not, and all the data are weighted and fused, among which some incorrect data have high weights, affecting the overall accuracy rate. The algorithm in this paper considers not only the amount of data and time delay, but also the reliability of the data. High reliability will increase the weight value. Therefore, compared with the other three algorithms, the data accuracy of the proposed algorithm is increased by 5.2%, 9.5% and 19.9%, respectively.

Table 9. AAR/%

Node	Proposed	FKCI	CCA	STDF
5	85.1	80.2	79.5	76.4
10	86.7	79.3	77.5	72.5
15	88.6	81.4	76.4	71.9
20	89.4	81.7	76.9	72.3
25	88.7	83.5	79.2	68.8

In this section, the average end-to-end delay of data is used to measure the real-time performance of data. The shorter the delay, the better the real-time performance. In the simulation, data is set to eventually arrive at the base station. In this algorithm, there are three kinds of data with priorities ranging from 0 to 2, and each hop delay is required to be lower than 200ms, 400ms and 600ms respectively. Other algorithms do not consider

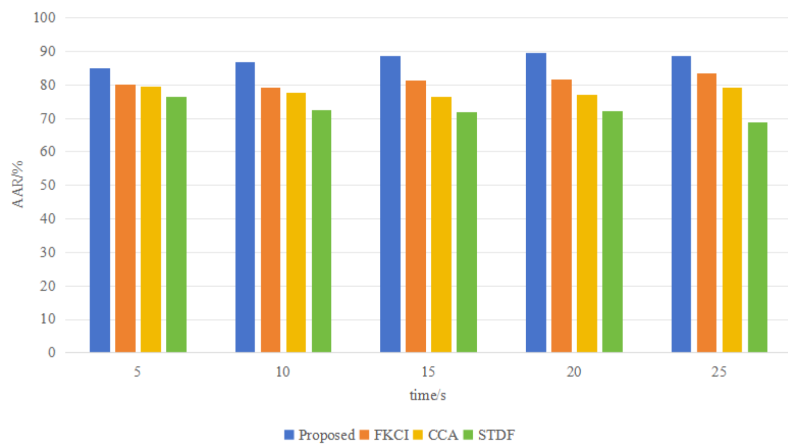

Fig. 8. The relationship between the accuracy rate of BS receiving data and the communication time

priorities. Figure 9 shows the relationship between data average end-to-end latency and traffic.

The wireless communication model [41,42] adopted in this paper is shown in equations (10-12).

$$E_{Tx}(l,d) = lE_{elec} + l\theta_{fs}d^2, d < d_0. \tag{8}$$

$$E_{Tx}(l,d) = lE_{elec} + l\theta_{amp}d^4, d \geq d_0. \tag{9}$$

$$E_{Rx}(l,d) = lE_{elec}. \tag{10}$$

Where E_{elec} represents the energy consumption of the transmitter running the transmitter or receiver, θ_{amp} represents the energy consumption of the signal amplifier, and d_0 is the threshold of free space energy attenuation. Through network simulation calculation, the relationship between the average energy consumption of nodes in the network and the communication time is obtained, as shown in Figure 10. As can be seen from Figure 10 and table 10, the average energy consumed by nodes gradually increases with the communication time. STDF algorithm uploads data directly, which results in a large number of data packets and consumes a lot of energy. Compared with STDF algorithm, CCA algorithm adopts fuzzy weighting method for data fusion, which reduces the amount of data and energy consumption. However, it does not distinguish whether the data is correct or not, and all the fusion processing, at the cost of data accuracy in exchange for less energy consumption. The algorithm in this paper removes part of the wrong data, and then merges and uploads, saving energy. FKCI algorithm in the initial phase out some of the inaccurate data, and within the cluster, the cluster between the fusion respectively, reduces the amount of data. In terms of energy consumption, compared with FKCI, STDF and CCA, the proposed algorithm is reduced by 21%, 9% and 4% respectively.

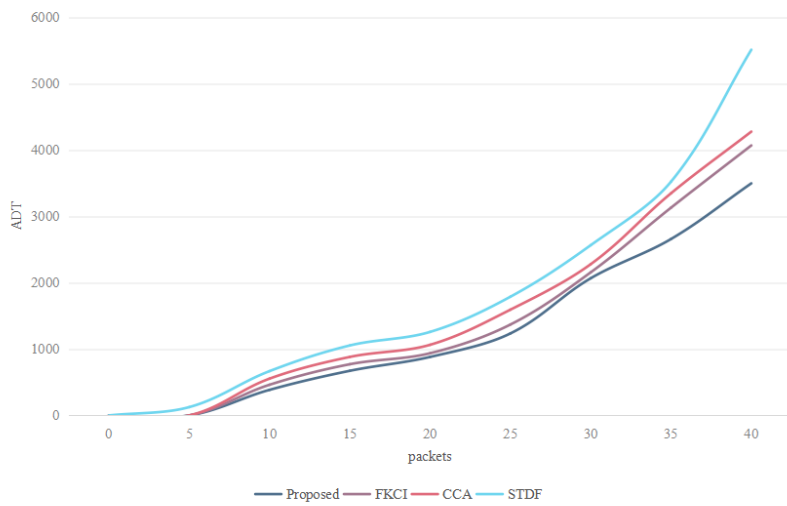

Fig. 9. Relationship between ADT and traffic

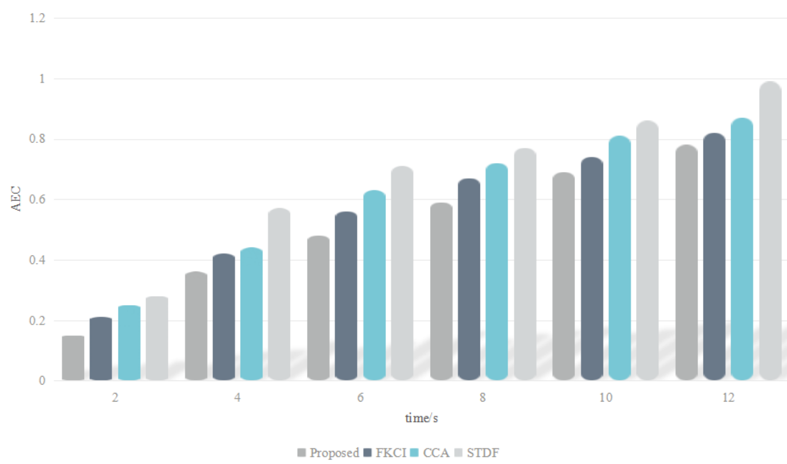

Fig. 10. Comparison of energy consumption

Table 10. Comparison of energy consumption

Time	Proposed	FKCI	CCA	STDF
2	0.15	0.21	0.25	0.28
4	0.36	0.42	0.44	0.57
6	0.48	0.56	0.63	0.71
8	0.59	0.67	0.72	0.77
10	0.69	0.74	0.81	0.86
12	0.78	0.82	0.87	0.99

6. Conclusion

Based on the characteristics of large amount and high complexity of data collected by industrial multi-source sensors, in order to meet the actual demand, the deep learning autoencoder theory and Q-learning are used to conduct in-depth research on the multi-source sensor data fusion method. The performance of the fusion method is studied from both theoretical and experimental aspects. The method can accurately extract and classify data features, reduce the energy consumption of data fusion, and greatly improve the quality of multi-source sensor data fusion. Therefore, the proposed method has obvious advantages and can provide a new reference idea for the development of multi-source sensor data fusion technology.

Acknowledgments. This paper was supported by Research Project of Zhengzhou University of Science and Technology: Vibration Analysis and Control Based on Flexible Joints and Loads of Industrial Robots (2022XJKY05); Key Teaching reform project of Zhengzhou University of Science and Technology, ”Research on Cultivating Interdisciplinary Innovative Practice Teaching Based on Robot Technology” (2022JGZD03); 2023 Teacher Development Research Project of Zhengzhou University of Science and Technology, Research on Career Development Dilemma and Implementation Path of Young Teachers in Application-oriented Universities in the New Era (JSFZ-ZXKT2023004); Key research projects of universities in Henan Province, ”Development and Research of Intelligent Gauge Detection System of Rail Transit Based on Dynamic Distance Measurement” (24B460027), ”Application Research of Rhodes Correlation Failure Model Based on Big Data Acquisition in high-speed Cutting Analysis” (24B460028); ”Research on Structure Optimization and vibration Damping Performance of high-frequency needling Equipment for Nonwoven Fabrics” (23B460011), ”Research on Key Technologies of Supercapacitor Module Equalization and Safety Monitoring and early Warning based on Evidence Theory” (23B480003).

References

1. Malik P K, Sharma R, Singh R, et al. Industrial Internet of Things and its applications in industry 4.0: State of the art[J]. Computer Communications, 2021, 166: 125-139.
2. Franco J, Aris A, Canberk B, et al. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems[J]. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2351-2383.
3. Wu Y, Dai H N, Tang H. Graph neural networks for anomaly detection in industrial internet of things[J]. IEEE Internet of Things Journal, 2021.
4. Liu C, Su Z, Xu X, et al. Service-oriented industrial internet of things gateway for cloud manufacturing[J]. Robotics and Computer-Integrated Manufacturing, 2022, 73: 102217.
5. Peng Li, Asif Ali Laghari, Mamoon Rashid, Jing Gao, Thippa Reddy Gadekallu, Abdul Rehman Javed, Shoulin Yin*, ”A Deep Multimodal Adversarial Cycle-Consistent Network for Smart Enterprise System,” in IEEE Transactions on Industrial Informatics, 2022, doi: 10.1109/TII.2022.3197201.
6. Nguyen T H T, Van Pham N, Nguyen V N N, et al. Forecasting Wind Speed Using A Hybrid Model Of Convolutional Neural Network And Long-Short Term Memory With Boruta Algorithm-Based Feature Selection[J]. Journal of Applied Science and Engineering, 2022, 26(8): 1055-1062.
7. Jiang B, Li J, Yue G, et al. Differential privacy for industrial internet of things: Opportunities, applications, and challenges[J]. IEEE Internet of Things Journal, 2021, 8(13): 10430-10451.

8. Shoulin Yin, Hang Li, Asif Ali Laghari, et al. A Bagging Strategy-Based Kernel Extreme Learning Machine for Complex Network Intrusion Detection[J]. EAI Endorsed Transactions on Scalable Information Systems. 21(33), e8, 2021. http://dx.doi.org/10.4108/eai.6-10-2021.171247
9. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
10. Gao S. A two-channel attention mechanism-based MobileNetV2 and bidirectional long short memory network for multi-modal dimension dance emotion recognition[J]. Journal of Applied Science and Engineering, 2022, 26(4): 455-464.
11. Xiong G, Ma W, Zhao N, et al. Multi-type diesel engines operating condition recognition method based on stacked auto-encoder and feature transfer learning[J]. IEEE Access, 2021, 9: 31043-31052.
12. Mahdavifar S, Alhadidi D, Ghorbani A. Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder[J]. Journal of Network and Systems Management, 2022, 30(1): 1-34.
13. Pang G D, Lin Y C, Qiu Y L, et al. Dislocation Density-Based Model and Stacked Auto-Encoder Model for Ti-55511 Alloy with Basket-Weave Microstructures Deformed in $\alpha + \beta$ Region[J]. Advanced Engineering Materials, 2021, 23(4): 2001307.
14. Xie Y, Gong M, Wang S, et al. Community discovery in networks with deep sparse filtering[J]. Pattern Recognition, 2018, 81: 50-59.
15. Muzammal M, Talat R, Sodhro A H, et al. A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks[J]. Information Fusion, 2020, 53: 155-164.
16. Xiao F. Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy[J]. Information Fusion, 2019, 46: 23-32.
17. Pan Y, Hong R, Chen J, et al. Performance degradation assessment of a wind turbine gearbox based on multi-sensor data fusion[J]. Mechanism and machine theory, 2019, 137: 509-526.
18. Fei S, Hassan M A, Xiao Y, et al. UAV-based multi-sensor data fusion and machine learning algorithm for yield prediction in wheat[J]. Precision agriculture, 2022: 1-26.
19. Tong W, Chu X, Jiang W, et al. A multimodal dual-fusion entity extraction model for large and complex devices[J]. Computer Communications, 2023, 210: 1-9.
20. Jiang, Y., Yin, S.: Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, vol. 20, no. 4, 2023. https://doi.org/10.2298/CSIS221228034J
21. Saranya S S, Fatima N S. IoT-Based Patient Health Data Using Improved Context-Aware Data Fusion and Enhanced Recursive Feature Elimination Model[J]. IEEE Access, 2022, 10: 128318-128335.
22. Qiu S, Zhao H, Jiang N, et al. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges[J]. Information Fusion, 2022, 80: 241-265.
23. Li M, Chen H, Shi X, et al. A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application[J]. Applied Soft Computing, 2019, 84: 105677.
24. Li M, Chen H, Shi X, et al. A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application[J]. Applied Soft Computing, 2019, 84: 105677.
25. Wang Z, Chen D, Che X. Multi-kernel learning for multi-label classification with local Rademacher complexity[J]. Information Sciences, 2023, 647: 119462.
26. Mirbeygi M, Mahabadi A, Ranjbar A. Speech and music separation approaches-a survey[J]. Multimedia Tools and Applications, 2022, 81(15): 21155-21197.
27. Qin X, Ban Y, Wu P, et al. Improved image fusion method based on sparse decomposition[J]. Electronics, 2022, 11(15): 2321.
28. Satya Sreedhar P S, Nandhagopal N. Classification Similarity Network Model for Image Fusion Using Resnet50 and GoogLeNet[J]. Intelligent Automation & Soft Computing, 2022, 31(3).

29. Y. Liu, Y. Shi, F. Mu, J. Cheng, C. Li and X. Chen, ”Multimodal MRI Volumetric Data Fusion With Convolutional Neural Networks,” in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-15, 2022, Art no. 4006015, doi: 10.1109/TIM.2022.3184360.
30. Brunke L, Greeff M, Hall A W, et al. Safe learning in robotics: From learning-based control to safe reinforcement learning[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2022, 5: 411-444.
31. Gronauer S, Diepold K. Multi-agent deep reinforcement learning: a survey[J]. Artificial Intelligence Review, 2022, 55(2): 895-943.
32. Agarwal R, Schwarzer M, Castro P S, et al. Deep reinforcement learning at the edge of the statistical precipice[J]. Advances in neural information processing systems, 2021, 34: 29304-29320.
33. Wei W, Tang J. Cooperative Output Regulation By Q-learning For Discrete Multi-agent Systems In Finite-time[J]. Journal of Applied Science and Engineering, 2022, 26(6): 853-864.
34. Liu I J, Jain U, Yeh R A, et al. Cooperative exploration for multi-agent deep reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2021: 6826-6836.
35. Han X, Han Y, Chen Q, et al. Distributed flow shop scheduling with sequence-dependent setup times using an improved iterated greedy algorithm[J]. Complex System Modeling and Simulation, 2021, 1(3): 198-217.
36. Jisi A and Shoulin Yin. A New Feature Fusion Network for Student Behavior Recognition in Education [J]. Journal of Applied Science and Engineering. vol. 24, no. 2, pp.133-140, 2021.
37. Li Q, Meng X, Gao F, et al. Approximate cost-optimal energy management of hydrogen electric multiple unit trains using double Q-learning algorithm[J]. IEEE Transactions on Industrial Electronics, 2021, 69(9): 9099-9110.
38. Fawzy D, Moussa S, Badr N. The Spatiotemporal Data Fusion (STDF) approach: IoT-based data fusion using big data analytics[J]. Sensors, 2021, 21(21): 7035.
39. Cui H, Li S. Controllable Clustering Algorithm for Associated Real-Time Streaming Big Data Based on Multi-Source Data Fusion[J]. Wireless Communications and Mobile Computing, 2022, 2022.
40. Zhen C. Using big data fuzzy K-means clustering and information fusion algorithm in English teaching ability evaluation[J]. Complexity, 2021, 2021.
41. K. Chen et al., ”Recent advances in data-driven wireless communication using Gaussian processes: A comprehensive survey,” in China Communications, vol. 19, no. 1, pp. 218-237, Jan. 2022, doi: 10.23919/JCC.2022.01.016.
42. M. Akrout, A. Feriani, F. Bellili, A. Mezghani and E. Hossain, ”Domain Generalization in Machine Learning Models for Wireless Communications: Concepts, State-of-the-Art, and Open Issues,” in IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 3014-3037, Fourthquarter 2023, doi: 10.1109/COMST.2023.3326399.

Xi Zhang graduated from Zhongyuan University of Technology with a master’s degree in mechanical engineering in 2015. After graduation, he had been working in the field of vibration and noise control and automation control. Since 2017, he has been a full-time teacher, teaching assistant and lecturer of Zhengzhou University of Science and Technology.

Jiyue Wang graduated from chang’an university with a master’s degree in mechanical engineering in 2013. He has eight years of mechanical structure and automatic control design experience. Since 2015, he has been a full-time teacher, associate professor and master tutor of zhengzhou university of science and technology. He is the director of industrial robot research institute, engaged in the application of industrial robot technology and hydraulic automatic control research.

Ying Huang graduated from Southwest Minzu University with a master's degree in electronic information in 2023. After graduation, she has been working in the field of topology, modeling, and control techniques of switching converters. Since 2023, she has been a full-time teacher of Zhengzhou University of Science and Technology.

Feiyue Zhu, 2022 graduated from Henan University of Science and Technology with a master's degree in power engineering, and has been a full-time instructor at Zhengzhou University of Science and Technology since 2022. After graduation, he has been engaged in the field of new energy automobile battery pack liquid cooled plate heat exchange analysis and structural optimisation research work.

Received: March 14, 2024; Accepted: June 29, 2024.