Analysis and System Design of Mechanical Fault Diagnosis Based on Deep Neural Network

Keqin Zhao

School of Automobiles Studies, Tongji University, 201804 Shanghai, China

Correspondence should be addressed to Keqin Zhao; zhaokeqin@tongji.edu.cn

Received 14 June 2022; Revised 29 June 2022; Accepted 5 July 2022; Published 1 August 2022

Academic Editor: Lianhui Li

Copyright © 2022 Keqin Zhao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The operating environment of mechanical equipment is complex, and it is in high-intensity working conditions for a long time. The condition monitoring and fault diagnosis of equipment are very important. As a kind of precision part commonly used in mechanical equipment, the healthy operation of the rolling bearing is a necessary condition to ensure the reliable operation of the whole equipment. This research takes rolling bearing as the research object and is devoted to mechanical fault diagnosis analysis and system design. This paper studies the structure and working principle of rolling bearing, analyzes the types and locations of rolling bearing faults, and puts forward the overall framework of the fault diagnosis system and the workflow of the fault diagnosis system. This paper also studies the relevant theory of deep learning, proposes a neural network model framework for rolling bearing fault diagnosis, and preprocesses the rolling bearing vibration data. It uses a dropout algorithm and GRU to reduce the model parameter size and reduce the risk of overfitting.

1. Introduction

With the rapid progress of electronic technology, electromechanical equipment is gradually developing towards integration, large scale, and intelligence. There is close cooperation between different parts and components of electromechanical equipment, forming an organic whole [1–5]. Generally, the operation environment of electromechanical equipment is complex, and it is under high-intensity working conditions for a long time [6–11]. At least, the equipment is prone to failure, and at worst, it will cause casualties and serious social harm. In recent years, there have been many accidents caused by bearing failure all over the world, resulting in significant economic losses.

In this case, the equipment condition monitoring and fault diagnosis technology can be produced and developed [12, 13]. It generally requires the real-time monitoring of the status of the equipment during operation to determine the overall and local operation of the equipment [14–17]. One of the key technologies to ensure the safety and reliability of complex systems is to determine the reasonable maintenance time and formulate the corresponding maintenance system through real-time condition monitoring and fault diagnosis of electromechanical equipment. As a basic component of rotating mechanical equipment, rolling bearings are widely used in various mechanical equipment. The normal state of rolling bearings directly affects the normal operation of mechanical equipment. Rolling bearings have the advantages of low running friction resistance, high working efficiency, easy assembly, and use, so they are widely used in electromechanical equipment and play a vital role. One of the main reasons for the failure of rotating machinery is the failure of rolling bearings. The working quality of rolling bearings has a great impact on the working state of mechanical equipment.

According to statistics, 70% of mechanical faults are caused by vibration faults, while 30% of vibration faults are caused by rolling bearings [18–20]. The main reason is that the rolling bearings in mechanical equipment often play a key role in bearing and transmitting loads at the same time, and the working conditions of rolling bearings are generally poor, which is more prone to failure. The direct consequence caused by the failure of rolling bearings is to reduce and lose some functions of the electromechanical system or to cause
catastrophic accidents. On the other hand, compared with other mechanical parts, the service life of rolling bearings is very discrete. Even if the production personnel, equipment, materials, and processes are the same, the service life of a batch of bearings processed is very different. Because of this, in the actual use process, some bearings have greatly exceeded the design life but are still in good condition and normal operation, while some bearings are far from reaching the design life but have various faults and can only be replaced in advance.

Therefore, at present, enterprises generally adopt the method of regular maintenance according to the design life of bearings, which often leads to two situations: on the one hand, the rolling bearings that exceed the design life and are intact and in normal operation are removed as scrap parts during maintenance, resulting in material waste. On the other hand, the bearing that fails due to before reaching the design life is still working on the mechanical equipment with failure, which reduces the working accuracy of the electromechanical equipment between the bearing failure and the removal and scrapping. If the bearing damage intensifies during this period, it will lead to a serious failure of the whole equipment and threaten the production safety. Therefore, online monitoring of rolling bearings can effectively prevent the decline of equipment working accuracy and reduce accidents, and on the other hand, it can maximize the working capacity of bearings and save material costs. Therefore, the fault diagnosis method of rolling bearings has always been one of the key technologies in mechanical fault diagnosis.

2. Overall Design of Mechanical Fault Diagnosis System

Bearings are easily damaged during use. On the one hand, due to the problems of low machining accuracy and imperfect technology in the production process of the bearing, the bearing will inevitably have many defects when it leaves the factory. On the other hand, the working environment of rolling bearings is usually harsh. When rolling bearings work in exposed situations, dust, small stones, and the impact load and overload of the machine itself will also damage the bearings.

Rolling bearing failures can be divided into the following types:

2.1. Fatigue Spalling. Fatigue spalling is caused by cyclically acting loads between the raceways and rolling elements. Cracks first appeared in the interior, and over time, the cracks slowly extended towards the surface. Finally, it extends to the contact surface to form peeling pits visible to the naked eye. Each spalling hole gradually extended and eventually joined together to form a large spalling. Fatigue spalling further contributes to bearing damage by causing vibration and noise during operation.

2.2. Wear. The entry of foreign objects into the bearing body is the main factor causing bearing wear. Raceways, rolling elements, cages, and journals are all easily accessed by foreign objects. Wear will increase the rolling bearing clearance and raise the noise and vibration during operation.

2.3. Plastic Deformation. When the rolling bearing is overloaded or invaded by foreign matter with high hardness, the raceway will be plastically deformed to form a pit. Dimples can cause flaking around and cause vibration and noise.

2.4. Corrosion. Corrosion is usually surface rust caused by chemical action such as water, lubricating oil, and air.

2.5. Broken. The excessive load on the bearing often causes brokers during the working process.

2.6. Gluing. Bearing gluing refers to the phenomenon that the surface of one part of the bearing and the surface of another part stick to each other. It is usually caused by poor lubrication conditions of rolling bearings under high-speed and heavy-load conditions.

Rolling bearing faults can be divided into three types: inner and outer ring faults and rolling element faults according to the location of the fault. Due to the need to collect data on various faults when classifying according to the form of faults and conducting experiments, these failures are often accompanied, and it is impossible to extract a single type of data accurately. And these faults often need to work for a long time under certain conditions to form inconvenient experimental acquisition. Therefore, this paper adopts the method of fault location to classify the faults.

According to the acceleration, rotational speed, vibration, and other signals collected during the working process of the rolling bearing, the method of deep learning [21–26] is used for fault diagnosis, and the online fault diagnosis system for the rolling bearing is designed. On this basis, mechanical equipment’s health management and failure prediction are realized.

First, the rolling bearing acceleration data collected by the data recorder installed at the work site are transmitted to Ethernet’s remote monitoring center. Then, the data on the remote monitoring center server are normalized. Finally, we input the processed data into the trained rolling bearing fault diagnosis system and output the current operating status information of the bearing after identification and display it on the interface. The specific fault diagnosis process is shown in Figure 1.

The data recorder collects the vibration signal of the rolling bearing, and the collected data are transmitted to the remote monitoring center through the Ethernet and preprocessed, such as denoising and normalization. Then, we input it into the trained neural network to obtain the current state information of the rolling bearing to realize the remote monitoring of the running state of the rolling bearing.

As shown in Figure 1, the fault diagnosis model based on GRU belongs to the field of deep learning, and the training process requires many data samples with labels. If the data are only collected through experiments, it requires a lot of
labor and time costs or cannot collect enough samples. In addition, considering the actual data will inevitably have noise problems due to external interference and the sensor itself. Therefore, the method of adding different forms of noise to the collected raw data is adopted, and artificially adding noise; on the one hand, the number of datasets is expanded, reducing labor and time costs; on the other hand, the model can also obtain an “immune” effect to noise and improve robustness.

The collected data are marked with one-hot encoding, convenient for later training. The evaluation method of the model is analyzed, and the k-fold validation method is selected to divide the dataset into three parts: training, validation, and testing. Given the characteristic that the vibration signal is a time series, the two-way loop GRU stacking, fully connected layer, dropout, recurrent dropout, regularization, and other technologies are used to construct the fault diagnosis model of rolling bearing.

After the model is constructed, to further ameliorate its diagnostic performance, improve the differential evolution algorithm, use the improved differential evolution algorithm to optimize the model hyperparameters on the validation set, train on the training set, and verify the model performance on the test set.

3. Design of Fault Diagnosis Model for Rolling Bearing

3.1. Two-Way Loop GRU. The three types of recurrent neural networks can process time series, but the information flow can only be passed from the front to back in time [27, 28]. Recurrent neural networks are so dependent on the order of data that the features extracted by forwarding processing of data and reverse processing of data by using recurrent neural networks may be completely different. Although using reverse-order features alone for feature execution tasks may not be ideal, it is important that reverse-order features can compensate for details not captured by positive-order features. Therefore, to further improve fault diagnosis performance, a two-way loop GRU layer is introduced into the model, which contains two gated recurrent units. Each gated recurrent unit processes the input sequence along a time direction. The representations of the two are combined after processing. Two-way loop GRU can often capture patterns that cannot be captured by unidirectional GRUs through forward- and reverse-order feature extraction.

3.2. Dense Connected Layer. The dense connection layer is also called the fully connected layer, and its main function is to realize the mutual mapping between linear spaces. It can map a $1 \times 6$ vector to a $1 \times 5$ vector. The fully connected layer is the most basic and widely used neural network. The fully connected layer is a network layer in which each node of this layer receives the output of all nodes in the previous layer. Due to this feature, the fully connected layer usually contains the most trainable parameters compared to other structures. According to statistics, the parameters of the fully connected layer account for about 80% of the parameters of the entire model. By inputting all data vectors into the fully connected layer, the fully connected layer can learn the overall distribution characteristics of the vector.

3.3. Final Model Structure. By combining two-way loop GRU and dense connection layer and then processing the input data simultaneously, the model can be made to learn the data characteristics from both the time dimension and the whole, by building a model in keras in TensorFlow using python and visualizing it on tensorboard. The length of the processed vibration signal samples is 300; the corresponding relationship between the labels of the samples after one-hot encoding is shown in Table 1.

The processing process after the vibration data is input into the model is as follows:
Step 1: the vibration signal vector with a length of 300 is input into the model and divided into left and right parts for processing, respectively, where the left is the two-way loop GRU part and the right is the dense connection layer part.

Step left 1: after receiving the vector of length 300 from Step 1, deform it to (10, 30) through the reshape layer.

Step left 2: after receiving a vector of shape (10, 30) and inputting it into the two-way loop GRU network stacked in two layers, in order for the second layer to can process the data of the first layer, the output of the first layer GRU is a combination of the output vectors of each time interval.

Step right 1: after receiving the vector, it is processed by two densely connected layers separated by a dropout layer and then output.

Step 2: after the left and right parts are processed separately, the output vectors have the same shape. We use the concatenate layer to combine the processing results of the left and right parts and input them into a $4 \times 1$ dense connection layer.

Step 3: the final dense connection layer uses the softmax activation function, the four elements of the output vector represent the probability that the sample belongs to the corresponding type, and the type represented by the element with the most significant value is taken as the predicted label of the sample.

### 4. Neural Network Model Optimization

With the continuous expansion of the neural network scale [29–32], theoretically, the model’s performance is getting better and better. Still, the model also faces unprecedented risks in the training process, and overfitting is one of them. Therefore, the model needs to be optimized for the overfitting problem during the training process.

Overfitting refers to the fact that the model’s performance on the training and non-training sets is too different, and the model that performs well on the training set performs poorly on the test set. This phenomenon is generally because the model learns the individual characteristics of the samples as the common characteristics of the data on the training set. It is hazardous for the model to overfit, which will lead to the loss of the model’s basic features, and the generalization performance becomes abysmal.

To improve the generalization ability of the model, this paper will use the following methods to reduce or eliminate the influence of overfitting.

### 4.1. Regularization

Regularization is the process of introducing additional information to solve a problem or prevent overfitting, which is a common method in machine learning. Occam’s razor principle tells us that if an event can be explained in multiple ways, usually the simplest explanation is the most likely to be correct. So, in neural networks, the principle means that simple models have less risk of overfitting than complex models; the simple model is the model with fewer parameters and smaller parameter values. After the model is determined, the number of parameters is fixed. At this time, reducing overfitting can only be achieved by limiting the value of the parameters. This method of making the value of model weights more regular is called weight regularization (referred to as regularization). The main way to achieve regularization is to add a more significant penalty term to encourage the model to tend to smaller weights when the weights are too large. The relationship between punishment and weight is mainly L1 and L2.

The loss function for L1 regularization is defined as

$$J = J_0 + \alpha \sum \omega$$

where $J$ is the model’s loss function, $J_0$ is the original loss function, $\alpha$ is the regularization coefficient, and $\omega$ is the weight coefficient. A regularizer defined in this way biases the model’s weights towards 0, resulting in the sparsity of the weights.

The loss function for L2 regularization is defined as

$$J = J_0 + \alpha \sum \omega^2.$$  

The meaning of the parameters in the formula is the same as in L1, but unlike L1, this regularization method does not lead to sparsity.

### 4.2. Dropout

The number of training parameters in a deep learning model is usually huge. If the number of training samples does not meet the standard at this time, it is difficult to avoid overfitting. After the overfitting model, it is almost unusable in practice although it performs well on the training set. At this time, if a combination of multiple different models is used, overfitting can be suppressed to a certain extent. However, the way of combining multiple models will significantly increase the training and testing time of the model.

To solve this problem, the concept of dropout is proposed. The so-called dropout is when the output of neurons in the model has a certain probability of $p$ becoming 0 during the training process. In this way, the neuron is not involved in this process. This enables the model to learn more generalized features rather than relying on the local features of a neuron.

The specific working process of dropout is as follows:

1. Set the trigger probability $p$ of dropout, $p \in (0, 1)$. The neuron will be masked out with probability $p$.
2. Make the masked neuron not participate in the training process and keep its input and output fixed.
(3) For other neurons that are not masked, update their weights and biases in the usual way.

(4) To restore the masked neurons to normal, repeat steps 2 and 3.

The neurons participating in the dropout will be output to the next layer of neurons through the probability component internally during the training process.

Before adding dropout, the calculation formula of the network is

\[ z = \sum_{i=1}^{n} w_i y_i + \text{bias}, \quad (3) \]
\[ y = f(z). \]

After adding dropout, the calculation formula of the network becomes

\[ y'_i = y_i r_i, \quad (4) \]

where

\[ r_i \sim \text{Bernoulli}(p), \]

\[ z = \sum_{i=1}^{n} w_i y'_i + \text{bias}, \quad (5) \]
\[ y = f(z), \]

where Bernoulli() represents a [0, 1] random variable obeying a Bernoulli distribution.

In the testing process, the network with dropout no longer randomly shields neurons as it does during training but multiplies the output value of the neuron by the probability \((1 - p)\) before outputting.

Principle analysis of dropout to reduce overfitting is as follows.

4.2.1. Play the Effect of the Multimodel Ensemble. Model ensemble refers to a means of building multiple models and combining them to work together to achieve better performance than a single model. The model ensemble can convert several low-performance models into a high-performance model through voting output.

For example, in a binary classification scenario, there are three classifier models \(m_1, m_2, \) and \(m_3\). Then, the effects of their ensemble may have the following situations (Tables 2–4).

As shown in Table 4, the correct rate of every single model is 1/3, and the correct rate after ensemble not only does not decrease but increases. This is because when the ensemble model uses the “minority-by-majority voting” method to make predictions, the majority model gives the wrong answer, so it does not work. This shows that when performing a model ensemble, the performance of a single model cannot be abysmal. Fortunately, the performance of neural network models generally does not have this problem. In Tables 2 and 3, the single-model accuracy rate is 2/3. Only the accuracy rate of Tables 1–3 is increased to 100% after the ensemble, and the results in Table 2 are the same as before the ensemble, that is, no rise and no fall. The results in Table 3 are that the 3 models are all predicting the same results for the same example, and the way to vote in 3 identical models obviously does not make sense. Like the results in Table 1, for the same example, the performance of the ensemble model will be improved only when the majority of the models can give the correct answer, thus correcting the errors of the few models.

Dropout shielding neurons are random, so a neural network with different structures can be formed after each dropout. And the performance of these networks will not be terrible; therefore, in this way, it forms a model ensemble situation similar to Table 1, thereby improving the model’s performance.

4.2.2. Improve Model Generalization Performance. Due to randomness, it is difficult for neurons to ensure that they are not masked every time; therefore, if the weight update relies too much on some fixed neurons, the network performance will be significantly degraded when they are masked. This reason drives the network to learn more general features that do not depend on certain neurons and thus make the network more generalizable. This is similar to the effect of regularization, which reduces the weight of neurons, thereby improving the network’s performance.

5. Simulation

To verify the fault diagnosis performance of the designed model, the simulation experiments were carried out on the model using the Bearing Dataset of Western Reserve University. The hardware and software environment of the experiment is shown in Table 5.

The simulation experiment is initially selected as the output size of the two-way loop GRU layer is 32. The output
Table 5: Simulation experiment hardware and software environment.

| Hardware       | Parameter                          |
|----------------|------------------------------------|
| Memory         | 32 g                               |
| GPU            | NVIDIA GeForce GTX 1060            |
| Hard disk      | 2T                                 |
| CPU            | Intel Core (TM) i5 9400F            |

| Software       | Parameter                          |
|----------------|------------------------------------|
| Python         | 3.8                                |
| Matlab         | 2018b                              |
| Tensorflow     | 2.3.1                              |
| Cuda           | 10.1                               |

Figure 2: Simulation experiment training and verification accuracy.

Figure 3: Simulation experiments to train and verify losses.
size of the first dense connection layer is 64, and the output size of the first dense connection layer is 32.

The training parameter epoch takes 100, the loss takes categorical cross-entropy, and the optimizer takes Adam.

Simulation experiment training and validation accuracies and training and validation losses are shown in Figures 2 and 3.

The model’s accuracy after training is 0.9685499804619234. It can be seen from the above data results that the accuracy of the fault diagnosis model designed in this paper reaches 96.85% on the test set, which is comparable to the performance of the model on the training set and validation set, which indicates that the model does not have overfitting. And as can be seen from the accuracy graph (Figure 2) and loss graph (Figure 3), after 100 iterations, although the curve gradually flattens, it is not horizontal, which shows that by increasing the size of the epoch, the model’s performance can continue to improve.

6. Conclusions

As a kind of precision part commonly used in mechanical equipment, the healthy operation of the rolling bearing is a necessary condition to ensure the reliable operation of the whole equipment. According to the practical needs of rolling bearing fault diagnosis, a fault diagnosis model is designed. The model improves the performance of the model by processing and combining the vibration signals in two different ways: bidirectional GRU and dense connection layer. Dropout and regularization techniques are used to solve the overfitting problem. The simulation results show that the accuracy of the test set is as high as 96.85%, which fully meets the practical engineering needs of rolling bearing fault diagnosis.

In this paper, a fault diagnosis system is established only for the single-point fault at different positions of the rolling bearing. As for other faults, due to the difficulty of data acquisition, no further exploration is made. The fault diagnosis system established in this paper can judge the fault position of the rolling bearing, and the function is rough. In the next step of research, quantitative analysis and prediction of the severity of the fault and the remaining life of the bearing in this state can be continued to further enrich the system functions.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Z. Guifan, “Fault diagnosis method of rotating machinery based on collaborative hybrid metaheuristic algorithm to optimize VMD,” Journal of Sensors, vol. 2022, pp. 1–11, Article ID 8054801, 2022.
[2] J. Liu, H. Yang, J. He, Z. Sheng, and S. Chen, “Unbalanced fault diagnosis based on an invariant temporal-spatial attention fusion network,” Computational Intelligence and Neuroscience, vol. 2022, Article ID 1875011, 15 pages, 2022.
[3] S. Wang, B. Han, H. Bao, J. Wang, and Z. Zhang, “Parallel cross-sparse filtering networks and its application on fault diagnosis of rotating machinery,” Journal of Sensors, vol. 2022, pp. 1–12, Article ID 9259639, 2022.
[4] L. Li, C. Mao, H. Sun, Y. Yuan, and B. Lei, “Digital twin driven green performance evaluation methodology of intelligent manufacturing: hybrid model based on fuzzy rough-sets AHP, multistage weight synthesis, and PROMETHEE II,” Complexity, vol. 2020, no. 6, pp. 1–24, Article ID 3853925, 2020.
[5] D. Yang, K. Zhou, F. Qi, and K. Dong, “Multidomain feature fusion network for fault diagnosis of rolling machinery,” Shock and Vibration, vol. 2022, pp. 1–12, Article ID 5478274, 2022.
[6] Z. Dong, Y. Liu, J. Kang, and S. Zhang, “A novel generative method for machine fault diagnosis,” Journal of Sensors, vol. 2022, pp. 1–11, Article ID 5420478, 2022.
[7] L. Hua, J. Zhang, D. Li, X. Xi, and M. A. Shah, “Sensor fault diagnosis and fault tolerant control of quadrotor UAV based on genetic algorithm,” Journal of Sensors, vol. 2022, pp. 1–8, Article ID 8626722, 2022.
[8] Y. Liu, Y. Cheng, Z. Zhang, and J. Wu, “Acoustic fault diagnosis of rotor bearing system,” Shock and Vibration, vol. 2022, pp. 1–9, Article ID 8028599, 2022.
[9] Q. Zhao, “Fault diagnosis method for wind power equipment based on hidden markov model,” Wireless Communications and Mobile Computing, vol. 2022, pp. 1–9, Article ID 6937616, 2022.
[10] M. Y. L. Chew and Ke. Yan, “Enhancing interpretability of data-driven fault detection and diagnosis methodology with maintainability rules in smart building management,” Journal of Sensors, vol. 2022, pp. 1–48, Article ID 5975816, 2022.
[11] L. Li and C. Mao, “Big data supported PSS evaluation decision in service-oriented manufacturing,” IEEE Access, vol. 8, no. 99, pp. 154663–154670, 2020.
[12] J. Xu, H. Zhou, and Y. Fang, “Intelligent roller bearing fault diagnosis in industrial internet of things,” Wireless Communications and Mobile Computing, vol. 2022, Article ID 1860946, 19 pages, 2022.
[13] Q. Chang, M. Zheng, J. Luo et al., “Intelligent diagnosis model of traction seat of urban rail vehicle based on harris hawks optimization,” Journal of Sensors, vol. 2022, pp. 1–23, Article ID 2087809, 2022.
[14] L. Li, T. Qu, Y. Liu et al., “Sustainability assessment of intelligent manufacturing supported by digital twin,” IEEE Access, vol. 8, pp. 174988–175008, 2020.
[15] J. Vives, “Incorporating machine learning into vibration detection for wind turbines,” Modelling and Simulation in Engineering, vol. 2022, Article ID 6572298, 8 pages, 2022.
[16] L. Di, Y. Xiao, F. Hu, Y. Liang, J. Tan, and Y. You, “Real-time monitoring of high voltage switch based on CAD,” Advances in Multimedia, vol. 2022, pp. 1–9, Article ID 9136046, 2022.
[17] L. Li, B. Lei, and C. Mao, “Digital twin in smart manufacturing,” Journal of Industrial Information Integration, vol. 26, no. 9, Article ID 100289, 2022.
[18] H. Huang and W. Liu, “Research and design of high-fidelity experimental bed system for wireless sensor network,” Journal of Electrical and Computer Engineering, vol. 2022, pp. 1–10, Article ID 8629211, 2022.
[19] W. Tu, J. Yang, Ya. Luo, L. Jiang, J. Xu, and W. Yu, “Vibration transmission characteristics and measuring points analysis of...
bearing housing system," *Shock and Vibration*, vol. 2022, pp. 1–12, Article ID 4334398, 2022.

[20] Y. Chen, "Neural network optimization and data fusion recognition method for intelligent mechanical fault diagnosis," *Journal of Sensors*, vol. 2021, pp. 1–9, Article ID 2695996, 2021.

[21] J. Xu, Y. Liu, J. Liu, and Z. Qu, "Effectiveness of English online learning based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1310194, 10 pages, 2022.

[22] H. He, Y. Chen, X. Huang et al., "Deep learning-based channel reciprocity learning for physical layer secret key generation," *Security and Communication Networks*, vol. 2022, Article ID 1844345, 11 pages, 2022.

[23] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, "Abstractive Arabic text summarization based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1566890, 14 pages, 2022.

[24] X. Gu, "Effect of deep learning on college students’ career planning," *Mathematical Problems in Engineering*, vol. 2022, Article ID 1573635, 12 pages, 2022.

[25] J. Zhang and Y. Lei, "Deep reinforcement learning for stock prediction," *scientific programming*, vol. 2022, Article ID 5812546, 9 pages, 2022.

[26] N. Brahimi, H. Zhang, L. Dai, and J. Zhang, "Modelling on car-sharing serial prediction based on machine learning and deep learning," *Complexity*, vol. 2022, Article ID 8843000, 20 pages, 2022.

[27] Z. Liu, D. Li, and L. Wei, "A new method for WebShell detection based on bidirectional GRU and attention mechanism," *Security and Communication Networks*, vol. 2022, Article ID 3434920, 11 pages, 2022.

[28] L. Liu, "Interrogative sentences recognition based on the GRU multiattentive layer model," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–7, Article ID 1354337, 2022.

[29] Z. Fu, "Computer network intrusion anomaly detection with recurrent neural network," *Mobile Information Systems*, vol. 2022, Article ID 6576023, 11 pages, 2022.

[30] Bo. Cao, C. Li, Y. Song, and X. Fan, "Network intrusion detection technology based on convolutional neural network and BiGRU," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1942847, 20 pages, 2022.

[31] T. Tian and F. Nan, "A multitask convolutional neural network for artwork appreciation," *Mobile Information Systems*, vol. 2022, Article ID 8804711, 8 pages, 2022.

[32] W. Gao and J. Wu, "Multigraph convolutional network enhanced neural factorization machine for service recommendation," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–19, Article ID 3747033, 2022.