Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.010986
Article

Tech Science Press

A Reinforcement Learning System for Fault Detection and Diagnosis in Mechatronic Systems

Wanxin Zhang[1,*] and Jihong Zhu[2]

[1]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, 510006, China.
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China.
[*]Corresponding Author: Wanxin Zhang. Email: zhangwx59@mail.sysu.edu.cn

Received: 12 April 2020; Accepted: 09 June 2020

Abstract: With the increasing demand for the automation of operations and processes in mechatronic systems, fault detection and diagnosis has become a major topic to guarantee the process performance. There exist numerous studies on the topic of applying artificial intelligence methods for fault detection and diagnosis. However, much of the focus has been given on the detection of faults. In terms of the diagnosis of faults, on one hand, assumptions are required, which restricts the diagnosis range. On the other hand, different faults with similar symptoms cannot be distinguished, especially when the model is not trained by plenty of data. In this work, we proposed a reinforcement learning system for fault detection and diagnosis. No assumption is required. Feature exaction is first made. Then with the features as the states of the environment, the agent directly interacts with the environment. Optimal policy, which determines the exact category, size and location of the fault, is obtained by updating Q values. The method takes advantage of expert knowledge. When the features are unclear, action will be made to get more information from the new state for further determination. We create recurrent neural network with the long short-term memory architecture to approximate Q values. The application on a motor is discussed. The experimental results validate that the proposed method demonstrates a significant improvement compared with existing state-of-the-art methods of fault detection and diagnosis.

Keywords: Classification; reinforcement learning; neural network; feature exaction and selection; fault detection and diagnosis

1 Introduction

The demand on the automation of operations and technical processes increases progressively in recent years. Fault detection and diagnosis is a key part of process automation to ensure the process performance, product quality standards and meanwhile ensure the safety of the working environment [1–3]. The purpose of fault detection and diagnosis is to find the category, location and scale of the fault, so that effective counteractions can be taken in time to reduce the effect of the fault. Fault detection recognize the appearance of a fault in the system, and fault diagnosis categorizes the fault, which provides supports to the design of redundant systems and selection of safety policies.

Many methods have been proposed to deal with fault detection and diagnosis. We can divide these methods into two categories. The first category is based on the mathematical signal and process models of the plant, with the tool of statistical theory and soft computing. Typical methods include parity equations [4], observer-based methods [5], Wavelet analysis [6,7], principal component analysis [8], etc. The second category has no dependence on the mathematical models of the plant and applies artificial intelligence approaches for pattern analysis and classification of faults, including fuzzy systems [9,10], neural networks [11–13], fault trees [14–16], Bayes classification [17,18], artificial immune systems [19], decision trees [20], deep learning [21–24], etc.

Although these methods can have a good performance on fault detection and diagnosis, assumptions must be made for these methods, such as linear structures in a nonlinear system, accurate measurements, constant parameters, limited disturbances and open loop operations, etc. However, many assumptions cannot be satisfied in practice. As a result, the rang of the fault categories which can be distinguished by the existing method is restricted. An intelligent method capable of dealing with faults of different locations, different categories and different sizes with no limitations is required.

Furthermore, in most applications, considerable experience and expert knowledge can be obtained on the symptoms of faults. Even under circumstances that exact values and exact models are not available, there still exist some experiences on the phenomena when faults occurs, and some analyses based on the physical mechanism of the process. Expert knowledge plays an important role in fault detection, because the experience and sensory abilities of human beings can help recognize the pattens of the fault and find out the cause and location according to the phenomenon and the characteristic information. However, the expert sensory knowledge is not fully exploited by the existing methods.

To solve these two problems, this paper develops a reinforcement learning (RL) system for fault detection and diagnosis with the cognitive ability by making use of the highly specialized expert knowledge.

RL solves the learning problem through interacting with the environment, and has been widely used to determine the decision according to the evaluative feedback from the environment [25–28]. Given the past sensations and current sensory observation, the agent selects an action to obtain the desired state. An optimal policy is learned by discovering which action yields the biggest reward. Dynamic programming is a traditional way to solve optimization problem. However, only the problems with limited sizes and complexity and exact models can be solved by dynamic programming methods. Supervised learning need less limitations, however, requires large amount of data to train the model, such as a neural network-based model. Instead of exploiting the information in the input-output data, RL interacts with the environment directly, yielding a powerful learning system. Studies of applying RL to classification have attracted interests from researchers. Peng et al. [29] handled the problem of segmentation in object recognition by using the RL techniques. Greiner et al. [30] regarded the problem of classification in object recognition as a RL framework with the classifiers as policies which map states and actions. Zhao et al. [31] used reinforcement learning with convolutional neural network for automatic vehicle classification. Fault detection and diagnosis can be regarded as classification of categories, sizes and locations of faults.

The remainder of this paper is organized as follows. Background of proposed scheme, with the description of feature exaction, is given in Section 2. In Section 3, the proposed method is discussed in detail. RL system with recurrent neural network (RNN) for fault detection and diagnosis is designed. The long short-term memory (LSTM) architecture is adopted. In Section 4, studies and application on a permanent-magnet brushless motor is shown. Comparison results with four state-of-art methods is given. Finally Section 5 concludes the paper.

Frequently used symbols are summarized in Tab. 1.

Table 1: List of symbols

Letter symbol	Description
k	Discrete time $k = t/T_0 = 0, 1, 2, \ldots (T_0$ is sampling period)
$\boldsymbol{y}$	Output vector of the signal model
$\boldsymbol{u}$	Input vector of the signal model
$\boldsymbol{x}$	State vector of the signal model
$\boldsymbol{\theta}$	Parameter vector
S	Spectral density
R	Covariance or correlation function
$\boldsymbol{s} = \{s_1, s_2, \ldots\}$	State vector of the reinforcement learning system
$\boldsymbol{a} = \{a_1, a_2, \ldots\}$	Action vector of the reinforcement learning system
$\boldsymbol{A}$	Action space of the reinforcement learning system
F	Fourier transform

2 Background of Proposed Scheme

We propose a RL-based fault detection scheme as shown in Fig. 1. Different methods can be selected for signal model, such as state space model, correlation function, spectrum analysis, etc. The general mathematical form for the model can be given by $\boldsymbol{y}(t) = h[\boldsymbol{u}(t), \boldsymbol{x}(t), \boldsymbol{\theta}]$. In practice, the measurements of $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ are available, and $\widehat{\boldsymbol{\theta}}$ is obtained through parameter estimation which is constant in most cases. The auto-covariance function and power density for a variable (such as y_i) are given as

$$R_{y_i y_i}(\tau) = \mathrm{cov}[y_i, \tau] = E\{y_i(k) y_i(k+\tau) - \bar{y}_i^2\}, \tag{1}$$

$$S_{y_i y_i}(i\omega) = \mathrm{F}\{R_{y_i y_i}(\tau)\} = \sum_{\tau=-\infty}^{\infty} R_{y_i y_i}(\tau) e^{-i\omega T_0 \tau}. \tag{2}$$

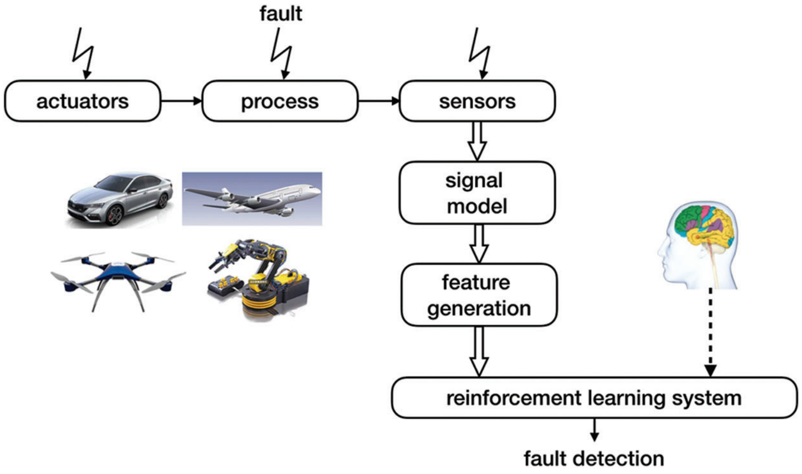


Figure 1: The framework for the proposed method

These two functions are of great importance in some applications, because they express the internal similarity inside the signal and will be affected when a fault occurs.

Given the measured variables and estimated parameters, features exaction is then considered. Related features, such as amplitudes, windowed sums, derivatives, variations from the steady-state values, exceeded thresholds, frequencies, are usually considered. Two principles for selection of features are given:

- The features chosen should have inherent dependences on the faults to be detected;
- The features chosen should have the ability to distinguish different faults.

In the exacted features, abnormal change may come out as a result of existing faults in the process. Different types of faults may present changes with completely different forms. For example, an abrupt fault presents a stepwise change, and an incipient fault gives a drifting change, and an intermittent fault presents an intermittent change. To distinguish these three kinds of different faults, length of data should be considered in the features selection to prevent missing the intermittent change or ignoring the drifting change.

The RL system generates the experiences of human beings and learns the fault detection policy. The policy is determined based on the value function which is denoted as the expectation of the discounted long-term rewards for the current state $\boldsymbol{s}_k$. $\boldsymbol{s}_k$ consists of the generated features. Selection of the features depends on the specific application. The principle for the selection is that the features chosen to build the state vector of the RL system are able to reflect the symptoms of the faults.

3 Proposed Method

3.1 RL System

In most applications of fault detection and diagnosis, we have a Markov decision process. Thus RL techniques can be applied. We propose a RL system for fault detection and diagnosis, by using Q-learning to achieve the optimal strategy without being aware of the process models. Expert knowledge and prior experiences are exploited to update the value function.

Same effect may be performed by different fault in a real-world environment, which is hard to categorize for a mathematical model. The RL system gives determinations in the form of proportional distribution. When the prior knowledge is limited, similar percentage is distributed for different faults. More accurate distribution for faults is given when more experiences and expert knowledge is available. The optimal policy can be learned if enough measured data in different situations is available.

Each element in the action space corresponds to a certain type of fault. The agent selects one action from the action space to execute according to the Q values. With the widely used ε-greedy method, the agent selects the action with biggest estimated value. An overview of RL system proposed is illustrated in Fig. 2.

3.2 Q Learning with LSTM Network

Recurrent neural network (RNN) is considered to learn Q values owing to its ability of exploiting contextual information from input and output sequences. Each input is the state variable. The number of input units is equal to the number of features exacted from the environment. Each output is the Q value of the corresponding action. The number of output units is equal to the number of types of possible faults. In order to deal with the vanishing gradient problem when using classic RNN, long short-term memory (LSTM) architecture is used. A set of recurrently connected memory blocks constitute the hidden layer. Each memory block contains one memory cell, controlled by the input gate, output gate and forget gate for write operation, read operation and reset operation, respectively. Fig. 3 shows the LSTM network with three input units, one hidden layer of four single-cell memory blocks, and two output units. The illustration for each memory block is given in Fig. 4.

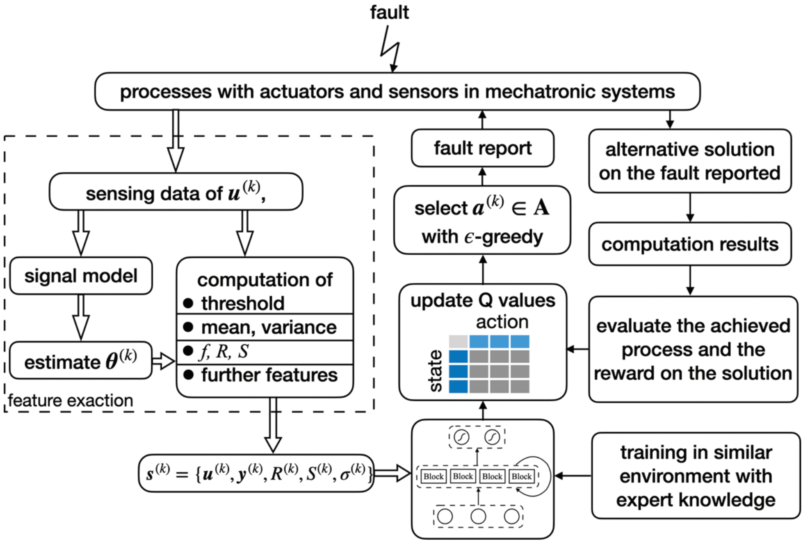

fault
processes with actuators and sensors in mechatronic systems
sensing data of $u^{(k)}$,
signal model
estimate $\theta^{(k)}$
computation of
threshold
mean, variance
f, R, S
further features
feature exaction
$s^{(k)} = \{u^{(k)}, y^{(k)}, R^{(k)}, S^{(k)}, \sigma^{(k)}\}$
fault report
select $a^{(k)} \in \mathbf{A}$ with ϵ-greedy
update Q values
action
state
Block
alternative solution on the fault reported
computation results
evaluate the achieved process and the reward on the solution
training in similar environment with expert knowledge

Figure 2: Illustration of the proposed RL system

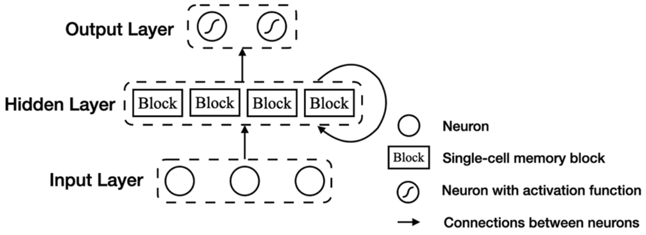


Figure 3: LSTM network

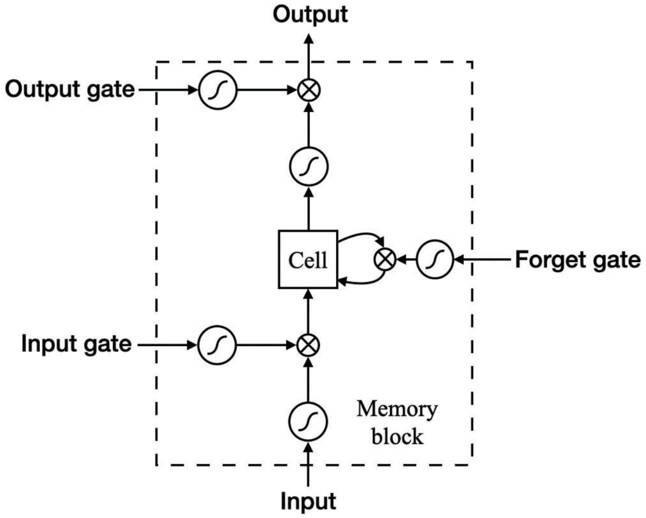


Figure 4: Illustration for the single-cell memory block

For each neuron i in the hidden layer and the output layer, the activation function is given by the standard logistic sigmoid function, denoted by $f(\cdot)$. The reason for the choice of the activation function is that the standard logistic sigmoid function has better approximation properties in the approximation theory, compared with other types, such as splines and polynomials [32]. Additionally, the standard logistic sigmoid function has a lower computation cost when applied in back propagation algorithm for neural networks [33]. We denote the input of the neuron i at moment k by $\mathrm{neui}_i(k)$, and the output by $\mathrm{neuo}_i(k)$. Then the output of the neuron with activation function is calculated by

$$\mathrm{neuo}_i(k) = f(\mathrm{neui}_i(k)) = \frac{1}{1 + e^{-\mathrm{neui}_i(k)}}. \tag{3}$$

At each step, the agent determines one action according to the Q values, and then new Q values are observed from the environment. Meanwhile, to minimize the mean squared error, weights of the LSTM for learning Q values are updated by the backward propagation algorithm, based on the gradient descent form which is of great significance and widely used [34–36]. The input layer, hidden layer and output layer are connected by weights. The input layer receives information from the input signal, and then transmits it to the hidden layer. According to the connections in Fig. 4, the input of the cell is

$$\mathrm{neui}_C(k) = \sum_j w_{jC}\mathrm{neuo}_j(k) + \sum_m v_{mC}\mathrm{neuo}_m(k-1), \tag{4}$$

where w_{jc} is the weights of the connection from input layer to the cell, and v_{mi} is the weights of the connection from other blocks to the cell in the case that the output of the block m at the last moment is cycled to the cell.

The output of the cell is

$$\mathrm{neuo}_C(k) = \mathrm{neuo}_F(k)\mathrm{neuo}_C(k-1) + \mathrm{neuo}_I(k)f(\mathrm{neui}_C(k)), \tag{5}$$

where $\mathrm{neuo}_F(k)$ is the output of the forget gate, and $\mathrm{neuo}_I(k)$ is the output of the input gate.

The neurons in the hidden layer develop internal representations for the exacted features in a way which exploits more information on $\boldsymbol{s}$ to produce the appropriate Q values for given observations from the environment. At each step of RL, the agent determines one action a according to the state s, and then the agent receives reward r. Given the new state s', the temporal different error is given by

$$e(s,a) = r + \gamma \max_{a'} Q(s',a') - Q(s,a). \tag{6}$$

where $Q(s, a)$ is the Q value of a at s, γ is the factor of discount. The selection of the action is evaluated, and Q values are updated accordingly,

$$Q(s,a) = Q(s,a) + \alpha e(s,a), \tag{7}$$

where α is the learning rate coefficient.

Through the iterative process, the agent learns from the experimental experiences and expert knowledge by exploring the real environment. The optimal policy determines the probability of the existing fault according to the observed state. The rules for the policy are understandable.

4 Application for Motors

Studies and applications on motors are shown in this section. A permanent-magnet brushless motor is considered, the structure of which is shown in Fig. 5. The overall diagram for the process is shown in Fig. 6. The measured signals consist of the voltage U, the current I and the mechanical rotation speed ω. The signals are first handled by analog filters to get rid of the sawtooth on the signal edges, and then sampled from the analog form to digital form. Finally, the sampled signals are sent to the personal computer. In the process, the servo amplifier with feedback of I and ω is designed to implement speed control.

By defining $\boldsymbol{x} = [I(t), \omega(t)]^T$, $\boldsymbol{u} = [U(t), M(t)]^T$ where $U(t)$ is the voltage input and $M(t)$ is the load input, the process can be modeled by a state-space representation,

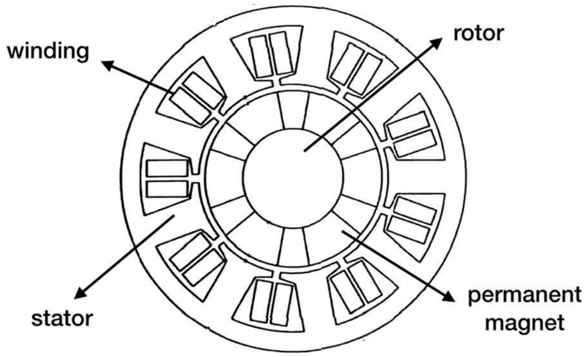


Figure 5: Structure of the permanent-magnet brushless motor

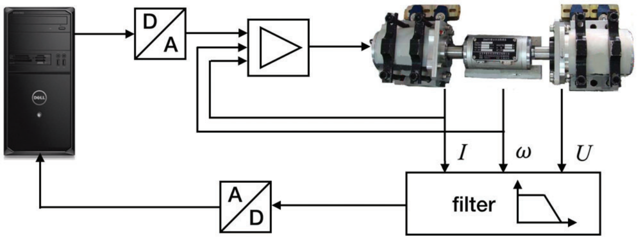


Figure 6: The overall diagram for the process

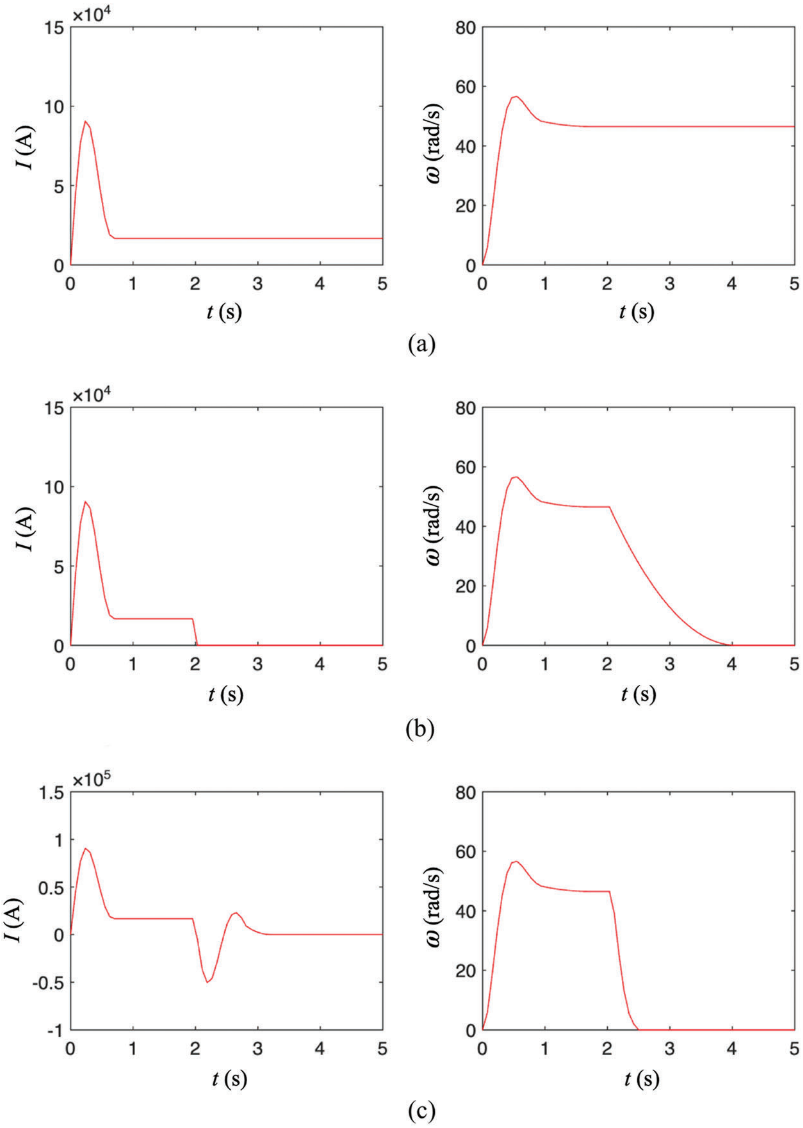
$$\dot{x} = f(x) + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} u \qquad (8)$$
$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x$$

where

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} \frac{1}{L}(-RI(t) - \psi\omega(t) - K|\omega(t)|I(t)) \\ \frac{1}{J}(\psi I(t) - F_1\omega(t) - F_0\mathrm{sign}(\omega(t))) \end{bmatrix}, \qquad (9)$$

and L is the inductance, J is the inertia constant, R is the resistance, ψ is the magnetic flux, K is the voltage drop factor, F_1 is the viscous friction, F_0 is the dry friction.

The action space $\{a_1, a_2, \ldots, a_n\}$ consists of the determination on the existence of the faults. Different action variable stands for different fault. Setting value of 1 indicates the detection of a fault. When no fault is detected, all values in the action space is set to be 0. The faults in the process shown in Fig. 6 consist of multiplicative change of the resistance which demonstrates an additive change in the logarithm of $\widehat{R}$ (estimates of R), change of the moment of inertia which demonstrates change in $\widehat{J}$, change of rotor inductance which demonstrates change in $\widehat{L}$, increased friction in the bearings which demonstrates change in $\widehat{F_1}$ and $\widehat{F_0}$, the gain fault in the voltage sensor which leads to abnormal changes in $\widehat{R}, \widehat{L}$ and $\widehat{\psi}$, the offset fault in the speed sensor which leads to abnormal change in $\widehat{\psi}$ and the additive fault in the current sensor which leads to abnormal change in $\widehat{R}, \widehat{L}$ and $\widehat{\psi}$, etc. The categories of faults are not limited to these. Converter disconnection and short circuit which can occurred at any location in the system may also exist (see Fig. 7). The abnormal changes in the estimates include fast reactions, large variance, but



(a)

(b)

(c)

Figure 7: Simulation results of two categories of faults. (a) normal states with no fault. (b) states with disconnection in the armature converter. (c) states with short circuit in the armature converter

also slow reactions. Mathematical method for detection of all fault at all locations is hard. However, based on the phenomena of the changes in the estimates, experts can recognize the fault and find the locations. Then the RL system learns from these expert knowledge, and finally find the optimal policy which determines whether any fault exists given the current state of the process.

When the learning system is used for fault detection, input excitation to the process is important. The motor should be excited by both the random signals and the constant signals, so that the fault when existed can have the symptoms in the measured signals which can then be detected. The state is chosen as $\mathbf{s}^{(k)} = \{\Delta U^{(k)}, \Delta I^{(k)}, \Delta\omega^{(k)}, \tilde{R}^{(k)}, \tilde{L}^{(k)}, \tilde{\psi}^{(k)}, \tilde{J}^{(k)}, \tilde{F}_0^{(k)}, \tilde{F}_1^{(k)}, \sigma^{2(k)}_{\widehat{R}}, \sigma^{2(k)}_{\widehat{L}}, \sigma^{2(k)}_{\widehat{\psi}}\}$, where $\Delta U^{(k)}$ is defined by $\Delta U^{(k)} = U^{(K)} - U_0$ (U_0 is the measurement of U when no fault exists. Definitions of $\Delta I^{(k)}$ and $\Delta\omega^{(k)}$ have the same form as $\Delta U^{(k)}$), $\tilde{R}^{(k)}$ is defined by $\tilde{R}^{(k)} = (\widehat{R}^{(k)} - \widehat{R}_0)/\widehat{R}_0$ ($\widehat{R}_0$ is the estimate of R when no fault exists. Definitions of $\tilde{L}^{(k)}, \tilde{\psi}^{(k)}, \tilde{J}^{(k)}, \tilde{F}_0^{(k)}, \tilde{F}_1^{(k)}$ have the same form as $\tilde{R}^{(k)}$), and $\sigma_{\widehat{R}}^{2(k)}$ is defined by $\sigma_{\widehat{R}}^{2(k)} = \frac{1}{N}\sum_{m=k-N+1}^{k} \left(\widehat{R}(m) - \overline{\widehat{R}}^{(k)}\right)^2$ (N is a given value of length). Definitions of $\sigma_{\widehat{L}}^{2(k)}$ and $\sigma_{\widehat{\psi}}^{2(k)}$ have the same form as $\sigma_{\widehat{R}}^{2(k)}$.

We compare the proposed method with four typical artificial intelligence methods:

- Neural network. A multi-layer perceptron network is adopted, with each symptom as an input and each category of fault as an output. The number of input neurons is 12. The number of hidden neurons is 15. Back-propagation algorithm is employed. The activation function is given by the standard sigmoidal function.
- Bayes classification. With the assumption of Gaussian probability density function, Naive Bayes algorithm is employed. The state vector $\boldsymbol{s}^{(k)}$ is used as the input. Posterior probability is estimated based on the training data in training stage, and then the decision whether the sample contains a fault is made according to the posterior probability in the testing stage.
- Decision trees. Binary decision is made by using the continuously distributed symptoms to distinguish different categories of faults. In this work the Iterative Dichotomiser 3 (ID3) id employed to construct the tree.
- Fault trees. Elements include logic connections, binary events and symptoms, with a hierarchical structure according to the human comprehension. The back-propagation and least squares algorithm are adopted to tune the parameters of connection functions.

We use cross-validation for estimating the accuracy by dividing the data into two part: one is for training and the other for testing. The experiments are carried out for ten times by randomly choosing different sets of fault types. The precision is calculated by the ratio of number of true positives (i.e., the number of items that fault actually exists and is successfully detected) and number of total predicted positives. The recall is calculated by the ratio of number of true positives and number of total positives (i.e., the number of total items that fault exists). The specificity is calculated by the ratio of number of true negatives (i.e., the number of items that no fault exists and the detection result is also no fault) and number of total negatives (i.e., the number of total items that no fault exists). The F1 score is a comprehensive evaluation index of the precision and recall. The accuracy of the model is given by the ratio of the number of corrected fault detection and the total number of items. Tab. 2 shows the overall results of Precision, Recall, Specificity, F1 score, and Accuracy. It is seen that the proposed method has a higher accuracy than the state-of-art methods. The main reason is that some faults with same symptoms cannot be distinguished by existing methods, while the proposed method can interact with the environment and obtain necessary information in next state for final determination.

Table 2: Fault detection results using machine learning methods

Method	Precision	Recall	Specificity	F1 score	Accuracy (%)
Neural network	0.957	0.954	0.956	0.955	95.44
Bayes classification	0.871	0.864	0.867	0.868	86.59
Decision trees	0.901	0.892	0.897	0.899	89.56
Fault trees	0.961	0.954	0.958	0.975	96.68
Proposed method	0.985	0.985	0.985	0.972	98.05

5 Conclusion

Fault detection and diagnosis is a key part of process automation in industry. It is difficult to diagnose the exact category of the fault, when the symptoms of different categories are similar. Many recent studies on artificial intelligence methods have been conducted for fault detection. However, assumptions are required, which restricts the diagnostic range. In this article, we have proposed RL for fault detection and diagnosis. The agent directly interacts with the environment. When the features are unclear, an action will be made to obtain new state for diagnosis from the possible faults. Our method employs LSTM to estimate Q values. The detailed theoretical analysis and experimental results of the motor problem show that our method can handle the fault diagnosis with more categories and less limitations in the applications. Furthermore, better accuracy is demonstrated.

Acknowledgement: The authors would like to thank the anonymous reviews for their helpful suggestions to improve the quality of the paper.

Funding Statement: This work was supported by the Soft Science Research Program of Guangdong Province under Grant 2020A1010020013, the National Defense Innovation Special Zone of Science and Technology Project under Grant 18-163-00-TS-006-038-01 and the National Natural Science Foundation of China under Grant 61673240.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. De Lemos, R., Timmis, J., Ayara, M., Forrest, S. (2007). Immune-inspired adaptable error detection for automated teller machines. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, 37(5),* 873–886. DOI 10.1109/TSMCC.2007.900662.
2. Chen, J., Kumar, R. (2015). Fault detection of discrete-time stochastic systems subject to temporal logic correctness requirements. *IEEE Transactions on Automation Science and Engineering, 12(4),* 1369–1379. DOI 10.1109/TASE.2015.2453193.
3. Dineva, A., Mosavi, A., Gyimesi, M., Vajda, I., Nabipour, N. et al. (2019). Fault diagnosis of rotating electrical machines using multi-label classification. *Applied Sciences, 9(23),* 5086. DOI 10.3390/app9235086.
4. Puig, V., Quevedo, J. (2002). Passive robust fault detection using fuzzy parity equations. *Mathematics and Computers in Simulation, 60(3–5),* 193–207. DOI 10.1016/S0378-4754(02)00014-9.
5. Frank, P. M. (2007). Enhancement of robustness in observer-based fault detection. *International Journal of Control, 59(4),* 955–981. DOI 10.1080/00207179408923112.
6. Muralidharan, V., Sugumaran, V. (2012). A comparative study of naïve bayes classifier and bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis. *Applied Soft Computing, 12(8),* 2023–2029. DOI 10.1016/j.asoc.2012.03.021.
7. Sun, Y., Chen, H., Tang, L., Zhang, S. (2019). Gear fault detection analysis method based on fractional wavelet transform and back propagation neural network. *Computer Modeling in Engineering & Sciences, 121(3),* 1011–1028. DOI 10.32604/cmes.2019.07950.
8. Deng, X., Tian, X., Chen, S., Harris, C. J. (2017). Fault discriminant enhanced kernel principal component analysis incorporating prior fault information for monitoring nonlinear processes. *Chemometrics and Intelligent Laboratory Systems, 162,* 21–34. DOI 10.1016/j.chemolab.2017.01.001.
9. Agrawal, V., Panigrahi, B., Subbarao, P. (2017). Intelligent decision support system for detection and root cause analysis of faults in coal mills. *IEEE Transactions on Fuzzy Systems, 25(4),* 934–944. DOI 10.1109/TFUZZ.2016.2587325.
10. Zhang, H., Han, J., Wang, Y., Liu, X. (2017). Sensor fault estimation of switched fuzzy systems with unknown input. *IEEE Transactions on Fuzzy Systems, 26(3),* 1114–1124.

11. Mekki, H., Mellit, A., Salhi, H. (2016). Artificial neural network-based modelling and fault detection of partial shaded photovoltaic modules. *Simulation Modelling Practice and Theory, 67,* 1–13. DOI 10.1016/j.simpat.2016.05.005.
12. Jia, F., Lei, Y., Guo, L., Lin, J., Xing, S. (2018). A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines. *Neurocomputing, 272,* 619–628. DOI 10.1016/j.neucom.2017.07.032.
13. Xu, L., Cao, M., Song, B., Zhang, J., Liu, Y. et al. (2018). Open-circuit fault diagnosis of power rectifier using sparse autoencoder based deep neural network. *Neurocomputing, 311,* 1–10. DOI 10.1016/j.neucom.2018.05.040.
14. Zhou, Z., Zhang, Q. (2017). Model event/fault trees with dynamic uncertain causality graph for better probabilistic safety assessment. *IEEE Transactions on Reliability, 66(1),* 178–188. DOI 10.1109/TR.2017.2647845.
15. Duan, R., Lin, Y., Hu, L. (2017). Reliability analysis for complex systems based on dynamic evidential network considering epistemic uncertainty. *Computer Modeling in Engineering & Sciences, 113(1),* 17–34.
16. Quan, J., Chunling, Z., Siqi, W. (2019). Qualitative analysis for state/event fault trees using formal model checking. *Journal of Systems Engineering and Electronics, 30(5),* 959–973. DOI 10.21629/JSEE.2019.05.13.
17. Wong, P. K., Zhong, J., Yang, Z., Vong, C. M. (2016). Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis. *Neurocomputing, 174,* 331–343. DOI 10.1016/j.neucom.2015.02.097.
18. Chen, H., Jiang, B., Ding, S. X., Lu, N., Chen, W. (2019). Probability-relevant incipient fault detection and diagnosis methodology with applications to electric drive systems. *IEEE Transactions on Control Systems Technology, 27(6),* 2766–2773. DOI 10.1109/TCST.2018.2866976.
19. Dai, Y., Qiu, Y., Feng, Z. (2018). Research on faulty antibody library of dynamic artificial immune system for fault diagnosis of chemical process. *13th International Symposium on Process Systems Engineering. Elsevier, 44,* pp. 493–498.
20. Sun, W., Chen, J., Li, J. (2007). Decision tree and PCA-based fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing, 21(3),* 1300–1317. DOI 10.1016/j.ymssp.2006.06.010.
21. Guo, H., Zhuang, X., Rabczuk, T. (2019). A deep collocation method for the bending analysis of kirchhoff plate. *Computers, Materials & Continua, 59(2),* 433–456. DOI 10.32604/cmc.2019.06660.
22. Anitescu, C., Atroshchenko, E., Alajlan, N., Rabczuk, T. (2019). Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua, 59(1),* 345–359. DOI 10.32604/cmc.2019.06641.
23. Shamshirband, S., Rabczuk, T., Chau, K. W. (2019). A survey of deep learning techniques: Application in wind and solar energy resources. *IEEE Access, 7,* 164650–164666. DOI 10.1109/ACCESS.2019.2951750.
24. Goswami, S., Anitescu, C., Chakraborty, S., Rabczuk, T. (2020). Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics, 106,* 102447. DOI 10.1016/j.tafmec.2019.102447.
25. Wang, K., Sun, W. (2019). Meta-modeling game for deriving theory-consistent, microstructure-based traction-separation laws via deep reinforcement learning. *Computer Methods in Applied Mechanics and Engineering, 346,* 216–241. DOI 10.1016/j.cma.2018.11.026.
26. Lewis, F. L., Vrabie, D., Vamvoudakis, K. G. (2012). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine, 32(6),* 76–105.
27. Otto, A. R., Skatova, A., Madlon-Kay, S., Daw, N. D. (2014). Cognitive control predicts use of model-based reinforcement learning. *Journal of Cognitive Neuroscience, 27(2),* 319–333. DOI 10.1162/jocn_a_00709.
28. Li, H., Chen, T., Teng, H., Jiang, Y. (2019). A graph-based reinforcement learning method with converged state exploration and exploitation. *Computer Modeling in Engineering & Sciences, 118(2),* 253–274. DOI 10.31614/cmes.2019.05807.
29. Peng, J., Bhanu, B. (1998). Delayed reinforcement learning for adaptive image segmentation and feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, 28(3),* 482–488. DOI 10.1109/5326.704593.
30. Greiner, R., Grove, A. J., Roth, D. (2002). Learning cost-sensitive active classifiers. *Artificial Intelligence, 139(2),* 137–174. DOI 10.1016/S0004-3702(02)00209-6.

31. Zhao, D., Chen, Y., Lv, L. (2017). Deep reinforcement learning with visual attention for vehicle classification. *IEEE Transactions on Cognitive and Developmental Systems, 9(4),* 356–367. DOI 10.1109/TCDS.2016.2614675.

32. DasGupta, B., Schnitger, G. (1992). The power of approximating: a comparison of activation functions. *Proceedings of the 5th International Conference on Neural Information Processing Systems,* pp. 615–622.

33. Karlik, B., Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems, 1(4),* 111–122.

34. Hamdia, K. M., Ghasemi, H., Zhuang, X., Alajlan, N., Rabczuk, T. (2018). Sensitivity and uncertainty analysis for flexoelectric nanostructures. *Computer Methods in Applied Mechanics and Engineering, 337,* 95–109. DOI 10.1016/j.cma.2018.03.016.

35. Hamdia, K. M., Ghasemi, H., Bazi, Y., AlHichri, H., Alajlan, N. et al. (2019). A novel deep learning based method for the computational material design of flexoelectric nanostructures with topology optimization. *Finite Elements in Analysis and Design, 165,* 21–30. DOI 10.1016/j.finel.2019.07.001.

36. Hamdia, K. M., Ghasemi, H., Zhuang, X., Alajlan, N., Rabczuk, T. (2019). Computational machine learning representation for the flexoelectricity effect in truncated pyramid structures. *Computers, Materials & Continua, 59(1),* 79–87. DOI 10.32604/cmc.2019.05882.