Electrical Machine Bearing Fault Diagnosis Based on Deep Gaussian Process Optimized by Particle Swarm

HAI GUO[1,*], HAORAN TANG[1], XIN LIU[1], JINGYING ZHAO[1, 2], LIKUN WANG[3]
[1] College of Computer Science and Engineering, Dalian Minzu University,
18 LiaoheWest Road, Dalian Development Zone, Dalian, 116600, CHINA
[2] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology,
Dalian, 116023, CHINA
[3] College of Electronic and Electrical Engineering, Harbin University of Science and Technology,
Harbin 150080, CHINA

Abstract—Aiming at the problems of low accuracy and slow diagnosis speed in the existing fault diagnosis model of electrical machine bearing, this paper presents an electrical machine bearing fault diagnosis method based on Deep Gaussian Process of particle swarm optimization(DGP). A total of 10 characteristics of 9 damage states and no fault states of the bearing are determined, constructing a deep Gaussian process model for electrical machine bearing fault diagnosis based on expectation propagation and Monte Carlo method, and use the particle swarm optimization algorithm to perform parameter searching optimization for its induction point value. The experimental results show that the fault recognition rate of DGP on the CWRU data set reaches 95%, significantly better than other deep learning methods, integration methods and machine learning methods. DGP method can better diagnose electrical machine bearing faults, provide technical support for the safe operation of the electrical machine which are important for real industrial applications.

Keywords—Deep Gaussian Process; electrical machine fault diagnosis; particle swarm optimization.
Received: June 24, 2021. Revised: March 18, 2022. Accepted: April 21, 2022. Published: May 18, 2022.

1. Introduction

With the deepening of machine learning research in the field of artificial intelligence, machine learning technology is increasingly used in the field of pattern recognition[1]. Traditional recognition tasks mainly apply machine learning models such as support vector machine (SVM), neural network, and random forest [2]. In recent years, deep learning has developed rapidly in academia and industry, significantly improve the accuracy of recognition on many traditional recognition tasks, demonstrates its superb ability to handle complex recognition tasks, attracted a large number of experts and scholars to conduct research on its theory and application [3-5].

Electricity fault diagnosis technology can find faults in electrical equipment at the early stage of fault diagnosis, therefore, timely targeted maintenance can be carried out, saving a lot of time and funds for repairing faults, while avoiding production stalls, it also improves economic efficiency. Also in the memory fault detection technology is also necessary, Eitan Yaakobi [6] et al. proposed a structure of single error correction WOM code with better WOM rate, this structure can effectively update and store the data in the memory. Liang Xi [7] et al. proposed a Multisource Neighborhood Immune Detector Adaptive Model for Anomaly Detection, solve various problems existing in the real-valued shape-space under dynamic environment mentioned and

improve the overall detection performances, and got better stability.

In today's production activities and daily life, the electrical machine is the most important motive power and drive unit, and it has been widely used in various fields of people's production and life.

The fault detection of the electrical machine often needs to detect the fault in a very short time, so as to carry out the targeted maintenance in time, so it needs fast detection speed and flexible detection method. The Gaussian Process has the characteristics of low computational complexity and fast convergence speed in a small sample space. The Gaussian process is named after the German mathematician Carl Friedrich Gauss to commemorate his proposal of the concept of normal distribution, developed based on statistical learning theory and Bayesian theory. In the following decades, rich research results have been obtained. Ori Shental [8] et al. proposed a Gaussian Belief Propagation Solver for Systems, compared with the traditional method, this method has a faster convergence speed. D Bickson [9] and others proposed a Gaussian Belief Propagation Based Multiuser Detection algorithm, compared with the previous formula, the new

algorithm reduces memory requirements, calculation steps and the number of messages passed. The deep Gaussian process has certain theoretical advantages and is suitable for the research of electrical machine fault detection technology. The deep Gaussian process model is a deep model that superimposes multiple Gaussian processes, any number of Gaussian processes can be superimposed. The Gaussian process controls the mapping between layers and also has the advantages of the Gaussian process. Zhao [10] et al. proposed Computer Modeling of the Eddy Current Losses of Metal Fasteners in Rotor Slots of a Large Nuclear Steam Turbine Generator Based on Finite-Element Method and Deep Gaussian Process Regression, the analysis results show that compared with the independent finite-element analysis, this method reduces the design cycle time and improves the design efficiency for a large-capacity turbine generator. Guo [11] et al. proposed Predicting Temperature of Permanent Magnet Synchronous Motor(PMSM) Based on Deep Neural Network. This model can effectively predict the temperature change of stator winding, provide technical support to temperature early warning systems and ensure safe operation of PMSMs. Wang [12] et al. proposed Cuckoo Search Algorithm for Multi-Objective Optimization of Transient Starting Characteristics of a Self-Starting HVPMSM. Experiments show that the optimization speed of this method is significantly faster than other methods, and this method has a faster convergence speed while ensuring accuracy.

Existing deep learning models have been able to diagnose electrical machine faults well, but there are still many problems such as insufficient accuracy and slow training speed. For example, the semi-supervised training method of deep belief network has the problem of slow training speed, the autoencoder network has problems such as limited expression features and difficulty in reconstruction, convolutional neural network training requires a lot of data, and the effect is not ideal when processing industrial signals, RNN has problems such as gradient disappearance [13]. Therefore, this paper uses the strong recovery ability of the deep Gaussian process to outliers and the strong non-linear problem processing ability to construct a deep Gaussian process classification model optimized by particle swarm optimization, and apply it to the fault diagnosis of electrical machine rolling bearings, and early warning of motor bearing faults based on abnormal changes in the signal before the fault occurs, so as to avoid electrical machine damage and reduce losses caused by this model.

2. Deep Gaussian Process Classification Model

2.1 Deep Gaussian process

For a given N observed values, $y=(y_1,...,y_N)^T$ and D-dimensional coordinates $X=(X_1,...,X_N)^T$, The output of each hidden layer of a DGP model with L layers can be expressed as $\{H_l\}_{l=1}^{L-1}$. The number of columns in H_i is the number of nodes in layer L. It is also called the dimension of the layer and can be written as D_i.,which can be expressed by equation (1).

$$H^l=\begin{bmatrix} h_{1,1}^l & \cdots & h_{1,D_l}^l \\ \vdots & \ddots & \vdots \\ h_{N,1}^l & \cdots & h_{N,D_l}^l \end{bmatrix} \tag{1}$$

in:

$$h_{n,i}^l=f^{l,i}(h_n^{l-1}) \tag{2}$$

f_i is given by Gaussian priors, usually for ease of understanding, latent variable dimensions are ignored and H^l is written as h^l ,and $f^{l,i}(\bullet)$ is written as $f^l(\bullet)$. First, set a zero-mean Gaussian prior for $p(f^l\mid\theta^l)$ of each layer, for layers with multiple nodes, the prior function is an independent Gaussian function inside each layer. Assuming that the noise of i.i.d can be parameterized in the output of each layer, the prior and dependent variables of the deep Gaussian process can be regarded as equation (3) and equation (4):

$$p(f^l\mid\theta^l)=GP(f^l\mid 0,K^l),\quad l=1,\cdots,L \tag{3}$$

$$p(h^l\mid f^l,h^{l-1},\sigma_l^2)=\prod_{n=1}^{N}N(h_n^l\mid f_l(h_n^{l-1}),\sigma_l^2),\quad h_n^0=x_n \tag{4}$$

K^l represents the kernel matrix between a given input and layer L, where $K^l=k(h^{l-1},h^{l-1})$, For the layer with $L>1$, the input will no longer be a certain value, and the corresponding output will not obey the normal distribution. When $L=1$, the model will become a shallow Gaussian process model . Finally, the conditional probability of a given target value in the output layer is shown in equation (5).

$$p(y\mid f^L,h^{L-1},\sigma_L^2)=\prod_{n=1}^{N}N(y_n\mid f_L(h_n^{L-1}),\sigma_L^2) \tag{5}$$

Figure 1 is an example of a two-layer model, where a hidden layer and an output layer are used for a two-dimensional problem. The number of nodes (D_L) of the output layer will be equal to the dimension of the observation value y_n of the regression problem, or equal to the number of classes of the classification problem. Like the shallow Gaussian process model, adding appropriate sparse technology to the deep model can effectively reduce the computational complexity of the deep Gaussian process.

By omitting the dimension in the symbol and adding a Gaussian prior to the induction point of each layer, the final Sparse Depth Gaussian Process Model can be written as equation (6)-equation (8).

$$p(\mathrm{u}^L \mid \theta^l) = \mathrm{N}(\mathrm{u} \mid 0, K_{\mathrm{u}^l,\mathrm{u}^l}{}^{-1}), \mathrm{l} = 1,\ldots,\mathrm{L} \quad (6)$$

$$p(\mathrm{h}^l \mid \mathrm{u}^l, \mathrm{h}^{l-1}, \sigma_l^2) = \prod_{n=1}^{N} N(\mathrm{h}_n^l \mid \mathrm{A}_n^l \mathrm{u}^l, \mathrm{K}_{\mathrm{h}_n^l,\mathrm{h}_n^l} - Q_n^l) \quad (7)$$

$$p(y \mid \mathrm{u}^L, \mathrm{h}^{L-1}, \sigma_L^2) = \prod_{n=1}^{N} N(\mathrm{y}_n \mid \mathrm{A}_n^L \mathrm{u}^L, \mathrm{K}_{\mathrm{h}_n^L,\mathrm{h}_n^L} - Q_n^l) \quad (8)$$

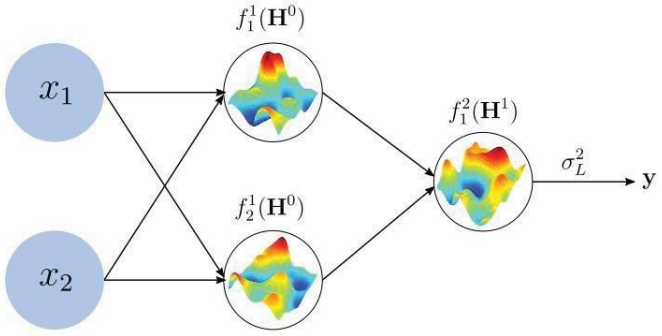

Fig. 1 An example of a DGP

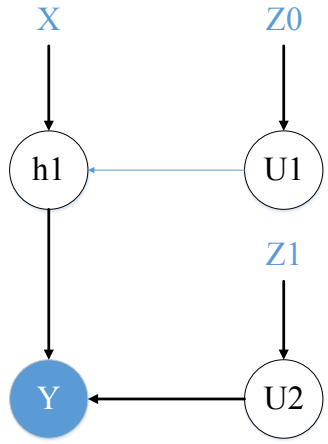

Fig. 2 Gauss process model with two-layer depth sparseness

The sub-index of the covariance matrix k is their corresponding output, for example, $K_{\mathrm{u}^l,\mathrm{u}^l}$ represents the covariance matrix of the induced point u_L. It takes position $K_{u_l,u_l} = k(\mathrm{z}^{l-1}, \mathrm{z}^{l-1})$ as a parameter, the covariance matrix K_{h_l,h_l} of nodes in a layer, and uses the output of the previous layer $K_{h_l,\mathrm{h}_i} = k(\mathrm{h}^{l-1}, \mathrm{h}^{l-1})$ to construct. We also define matrices A and Q as equation (9) and equation (10):

$$A_n^l \square K_{h_n^l,\mathrm{u}^l} K_{\mathrm{u}^l,\mathrm{u}^l}^{-1} \quad (9)$$

$$Q_n^l \square K_{h_n^l,\mathrm{u}} K_{\mathrm{u}^l,\mathrm{u}}^{-1} K_{\mathrm{u}^l,h_n^l} + \sigma_l^2 \quad (10)$$

Figure 2 shows a 2-layer deep sparse Gaussian process model. The hidden layer depends not only on the output of the layer but also on the induced point variables.

2.1 Reasoning technology of deep Gaussian process

In the deep Gaussian process model, the output of inference induction points $\{\mathrm{u}_l\}_{l=1}^L$ and hidden layer $\{\mathrm{h}_l\}_L^{l=1}$ is performed by marginalizing latent variables, which can predict the posterior probability of the test set and calculate the slight possibility of hyperparameter adjustment. However, both variables are difficult to handle. When considering a deep Gaussian process model with L=2 layers, the joint distribution of the model is shown in equation (11).

$$\mathrm{p}\left(y, \mathrm{h}^1, \{\mathrm{u}^l\}_{l=1}^2 \mid \{\sigma_l^2, \theta^l\}_{l=1}^2, X\right) = p\left(y \mid \mathrm{u}^2, \mathrm{h}^1, \sigma_2^2\right) p\left(\mathrm{h}^1 \mid \mathrm{u}^1, X, \sigma_1^2\right) \prod_{l=1}^{2} p\left(u^l \mid \theta^l\right) \ldots (11)$$

To simplify the description, all model parameters are now grouped into equation (12).

$$\alpha = \left\{\{z^l\}_{l=0}^1, \{\theta^l, \sigma_1^2\}_{l=1}^2\right\} \quad (12)$$

The marginal possibility is obtained by the marginalization $\{\mathrm{u}^l\}_{l=1}^2$ and the hidden variable h^l in equation (11) to obtain equation (13).

$$p\left(\mathrm{y} \mid \alpha, X\right) = \int p\left(y, \mathrm{h}^1, \{\mathrm{u}^l\}_{l=1}^2 \mid \alpha, X\right) du^1 du^2 dh^1 \quad (13)$$

However, some of the integrals in equation (13) are difficult to handle because they involve calculating the covariance function with respect to random variables [14]. Integration can be achieved by joint distribution in extension, as shown in equation (12). Starting from equation (10), a corresponding distribution of the output of layer $p\left(\mathrm{h}^1 \mid \mathrm{h}^{l-1}\right)$ is obtained, need to calculate a density-dependent nonlinear kernel function (Nonlinear kernel function of density) h^{l-1}.

Another thing that needs to be predicted is the posterior distribution on the induction point, which also requires the calculation of the integral (13) in the model evidence as shown in equation (14).

$$p\left(\{\mathrm{u}^l\}_{l=1}^2 \mid \alpha, X, y\right) = \frac{1}{p\left(y \mid \alpha, X\right)} \int p\left(y, \mathrm{h}^1, \{\mathrm{u}^l\}_{l=1}^2 \mid \alpha, X\right) dh^1 \ldots (14)$$

This result can be generalized to the case of $\mathrm{L} \geq 2$. For the sake of simplicity, the layer dependency will be removed from the symbol, and $\mathrm{u} = \{\mathrm{u}^l\}_{l=1}^L$ and $h = \{\mathrm{h}^l\}_{l=1}^{L-1}$ will be abbreviated, for any number of layers, the following (generalized) induction point becomes $p\left(\mathrm{u} \mid \alpha, X, y\right)$. In order to calculate the marginal likelihood and posterior, approximate reasoning techniques are needed.

Some work in the literature related to the deep Gaussian process suggests the use of a general sampling algorithm [15] to evaluate the logarithmic probability, the main difference from the method explained in this section is that the sampling algorithm does not set any distribution on the output (assuming the induced output is fixed), so they are included as model parameters. In this way, some of the benefits of regularization are lost. They also proposed to train the model by maximum a posteriori estimation (MAP), but the author did not compare the method with any other state-of-the-art technology, and no improvement was observed when the number of layers was increased.

Another method explained in [16] involves using a stochastic characterization vector $\phi(\bullet)$ to approximate the kernel function $k\left(x, \mathrm{x}'\right)$ of the Gaussian process. The core can be approximated as an inner product as shown in equation (15).

$$k(x,x') \approx \phi(x)^T \phi(x') \tag{15}$$

The results show that the deep Gaussian process model can be regarded as a Bayesian neural network, and the output of the layer is given by $g(wx+b), g(\bullet)$ is the activation function, w is the probability distribution $p(w)$, and b is Bayesian noise.

2.3 Deep Gaussian process based on expectation propagation and Monte Carlo method

The deep Gaussian process based on the expectation propagation and Monte Carlo method follows the method explained in [17], and uses the binding factor constrained expectation propagation algorithm to approach the posterior inducing point. As shown in Figure 3, the process of calculating $\ln Z_n$ at a single point in a 2-layer deep Gaussian process model. In the first level, $q(h^l)$ is given by a normal distribution (blue in the figure above) sampled from it. In the second layer, the true distribution of $\ln Z_n$ (blue) is no longer a Gaussian distribution, but is given by equation (16). The proposed method calculates and propagates the samples through the network (green) to make the model more flexible and able to approximate the heterogeneous distribution.

The final form of Z_n approximated by s samples is given by the Gaussian mixture as shown in equation (16).

$$Z_n = \frac{1}{S}\sum_{s=1}^{S} q\left(y \mid \hat{h}_s^{L-1}\right) \tag{16}$$

Where $\hat{h}_s^{L-1}$ represents $s-th$ samples from the corresponding distribution $q(\hat{h}_s^{L-1} \mid \hat{h}_s^{L-2})$, which can be calculated by the above sampling technique.

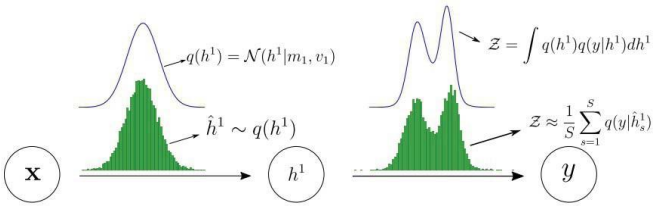

Fig. 3 The deep gauss process model is an example of network propagation

Contrary to the method proposed in [14], this method can capture the complex dependencies between DGP layers. In the literature [18], this method is also suitable for stochastic gradient descent training, such as small batch training. The final form of marginal likelihood approximation is shown in equation (17).

$$F(\alpha) = \sum_{l=1}^{L}[(1-N)\Phi(\theta_q^l) - \Phi(\theta_{prior}^l) + N\Phi(\theta^l)] + \frac{N}{|B|}\sum_{n=1}^{|B|}\ln Z_b \ldots (17)$$

Where α includes the model to be adjusted and the AEP parameters, |B| is the selected mini-batch size and Z_b can be calculated for each mini-batch using equation (16).

3. Particle swarm optimization optimizes the deep Gaussian process classification electrical machine rolling bearing fault diagnosis model

The deep Gaussian process electrical machine bearing fault diagnosis classification model constructed in this section uses expectation propagation and Monte Carlo methods to approximate the Gaussian posterior, and the particle swarm algorithm is used to search the number of induced points in the deep Gaussian process model in the range. The construction model adopts a 5-layer network structure, namely the input layer, the 3-layer hidden layer, and the output layer. The hidden layers all use the square exponential kernel as the kernel function of the Gaussian map, as shown in equation (18).

$$K_{SE}(x,x') = \exp\left(-\frac{\|d\|^2}{2l^2}\right) \tag{18}$$

The overall model network structure is shown in Figure 4:

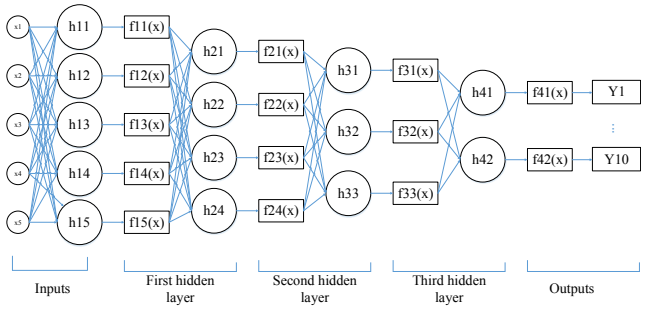

Fig. 4 Deep gauss process model network architecture

The particle swarm optimization algorithm (PSO) is a kind of swarm intelligence algorithm, and its design is based on the simulation of bird predation behavior. Assuming that there is only one food in the target area (that is, the optimal solution in the optimization problem), the goal of the flock of birds is to find this food source. Throughout the entire search process, the birds communicate with each other to let other birds find their position, and through collaboration, they can judge whether they find the optimal solution, and at the same time, they can also obtain the information of the optimal solution. Passed to the entire flock of birds, and finally the entire flock of birds can gather around the food source, that is, we have found the optimal solution, that is, the problem converges [19].

The particle swarm optimization algorithm simulates the birds in a flock of birds by designing a massless particle. The particle has only two attributes: speed *V* and position *X*. Speed represents the speed of movement, and position represents the direction of movement. Each particle searches for the optimal solution individually in the search space, which is recorded as the current individual extreme P_{best}, and the individual extreme value is shared with other particles in the entire particle swarm, find the optimal individual extreme value as the current global optimal solution G_{best} of the entire particle swarm. All particles

in the particle swarm adjust their speed and position according to the current individual extremum P_{best} found by themselves and the current global optimal solution G_{best} shared by the entire particle swarm. The idea of particle swarm optimization algorithm is relatively simple, mainly divided into: 1. Initialize the particle swarm; 2. Evaluate the particles, that is, calculate the fitness value; 3. Find the individual extreme value; 4. Find the global optimal solution; 5. Modify the speed and position of the particles. The particle swarm optimization algorithm is used to search the induction point value of the deep Gaussian process model, and the optimal parameter setting of the model is determined to improve the accuracy of model classification. The specific process is shown in Figure 5.

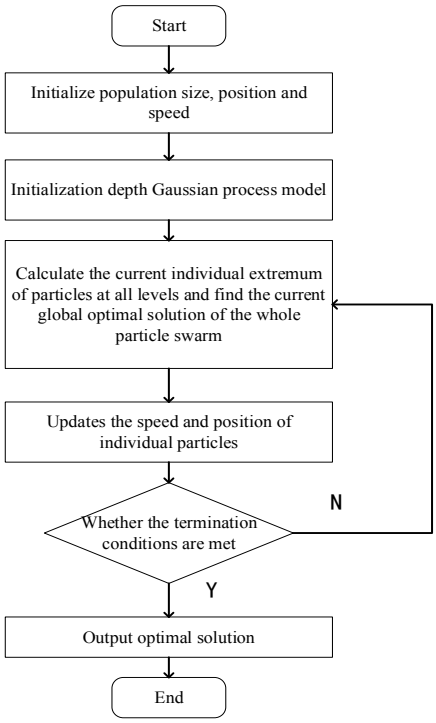

Fig. 5 Particle swarm optimization

4. Results and discussion

4.1 Sample Database

The test data comes from Case Western Reserve University (CWRU) Rolling Bearing Data Center. The CWRU data set is a world-recognized standard data set for bearing fault diagnosis. At present, the bearing fault diagnosis algorithm is updated quickly. In order to evaluate the superiority of the algorithm proposed in this chapter, all experiments use CWRU bearing data.

The CWRU bearing center data acquisition system is shown in Figure 6. The test object of this test is the drive end bearing in the picture. The model of the bearing to be diagnosed is the deep groove ball bearing SKF6205, which is manufactured by EDM under the load of 0HP, 1HP, 2HP, and 3HP. The sampling frequency of the system is 12kHz. There are a total of 3 types of defects in the diagnosed bearing, which are rolling element damage, outer ring damage and inner ring damage. The diameter of the damage is 0.007inch, 0.014inch and 0.021inch, and the specific information of 9 kinds of damage states is shown in the table. 4.1. In the experiment, 2048 data points are used for diagnosis each time. In order to facilitate the training of the deep Gaussian network, each segment of the signal is normalized, and the equation of the normalized processing is shown in (19).

$$\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{19}$$

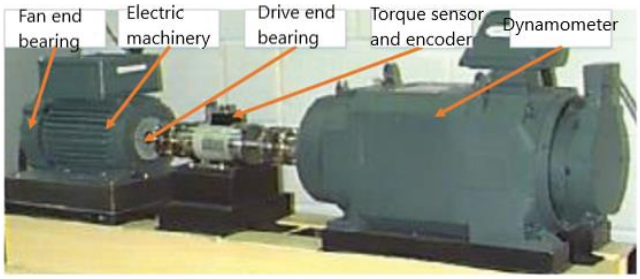

Fig.6 Data acquisition system of CWRU rolling bearing

Select 10000 pieces of data under 0HP load as shown in Table 4.1. There are 1000 pieces of data for each type of fault, including 900 training samples and 100 test samples. The training sample adopts the data enhancement method as shown in Figure 7. The length of the training sample collected each time is 2048, the offset is 1, and there is no overlap between the test samples.

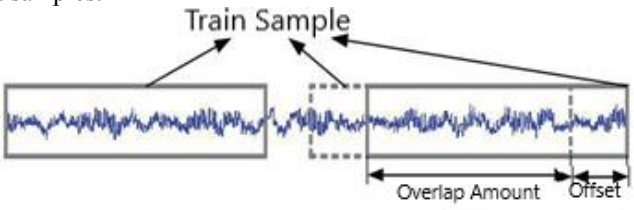

Fig.7 Data enhancement

Table 4.1 Test data set description

Fault category	No damage	rolling element*	rolling element**	rolling element***	Inner race*	Inner race**	Inner race***	Outer race*	Outer race**	Outer race***
Corresponding label	1	2	3	4	5	6	7	8	9	10

*Damage diameter is 0.007 inch (inch), **Damage diameter is 0.014 inch,***Damage diameter is 0.021 inch

4.2 Experimental conditions and evaluation indicators

The CPU used in the simulation experiment in this section is Intel i7-7130U, the memory is 4GB RAM, the programming language used is Python, and the framework used is Tensorflow, Keras and sklearn. Using 10,000 signals under zero load in the CWRU data set as samples, they are divided into 9 types of faulty bearings and 1 type of non-faulty bearings, totaling 10 categories. Each type of fault contains 1000 pieces of data, 90%

of which are taken as the training set, and the remaining 10% as the test set. The accuracy, precision, recall and F1-Score under macro-average are used as the evaluation indicators of the model. The specific formula is as follows.

In the classification problem, to analyze the effect of data and classifiers, evaluation indexes can be used for auxiliary analysis. In this paper, the following evaluation indexes are used to comprehensively analyze and discuss the experimental results.

Accuracy is the most primitive evaluation index in classification problems. The definition of accuracy is the percentage of correct results in the total sample. The equation is shown in (20):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

in:

- True Positive (TP): Positive samples predicted to be positive by the model;
- False Positive (FP): Negative samples predicted to be positive by the model;
- False Negative (FN): Positive samples predicted to be negative by the model;
- True Negative (TN): Negative samples predicted to be negative by the model.

Precision rate is the probability of the actual positive samples among all the predicted positive samples, which can be expressed by equation (21).

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

Recall rate is the probability of being predicted to be a positive sample in a sample that is actually positive, which can be expressed by equation (22).

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{22}$$

F1-score is a weighted average of the precision and recall of the model. The closer F1-score is to 1, the better the empirical effect is. The evaluation index f1-Score can be expressed by equation (23).

$$F_1 = 2 \times \frac{\mathrm{Pr}ecision \times \mathrm{Re}call}{\beta^2 \times \mathrm{Pr}ecision \times \mathrm{Re}call} \tag{23}$$

The specific parameters of the deep Gaussian process electrical machine rolling bearing fault diagnosis model are set as follows: the maximum iteration times is 500, the minimum batch_size is 100, the learning rate is 0.01, and each node contains The number of samples is 15, and the noise level is set to 1e-5. On this basis, the particle swarm optimization algorithm is used to search for the number of induced points in the range of [10,100]. The parameters of the particle swarm optimization algorithm are set as follows: population size is set to 100, the maximum number of iterations is 150, and the inertia factor is set to 2, and the weight factor is set to 0.5.

4.3 Experimental results and discussion

Under 10,000 data sets of electrical machine rolling bearings under 0HP, the deep Gaussian process model based on particle swarm optimization algorithm is used to classify faults. When the induction point is 50 and the number of iterations is 40 times, the fault diagnosis classification accuracy reaches the highest 0.95. Under the same experimental conditions, the results of comparison with deep learning models such as Deep

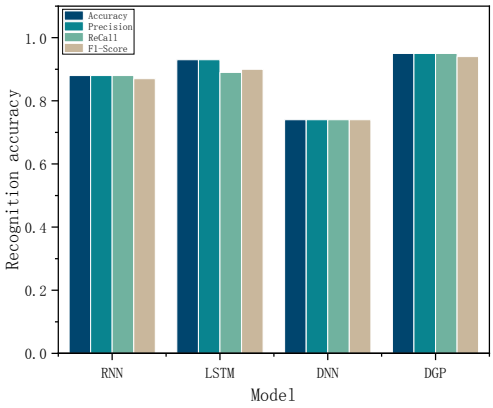

Fig.8 The accuracy of deep gauss process classification model compared with other deep learning models

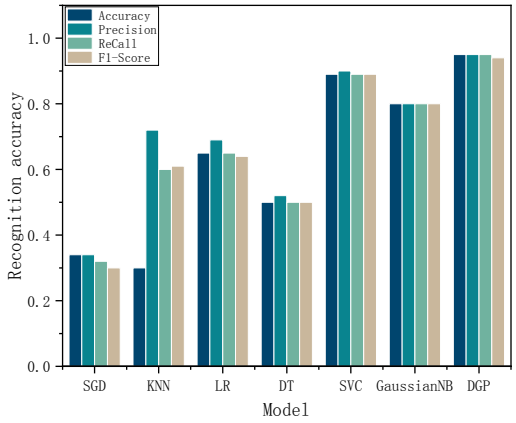

Fig.9 Comparison of accuracy between deep gauss process classification model and other machine learning algorithms

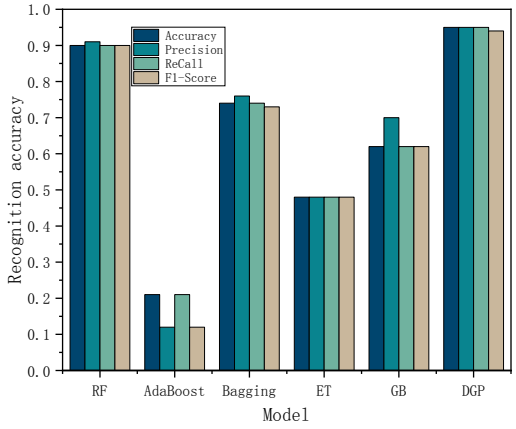

Fig.10 The accuracy of the depth gauss process classification model was compared with other ensemble learning models

Neural Network DNN, Recurrent Neural Network RNN, and long short-term memory network (LSTM) are as follows: Shown in Figure 8.

It can be seen from Figure 8 that the accuracy of the deep Gaussian process for bearing fault diagnosis under the samples used in this chapter is up to 0.95, while LSTM and RNN also maintain high accuracy rates of 0.93 and 0.88 respectively, the accuracy of deep neural network is 0.74, and the accuracy of deep Gaussian process model is higher than that of the above deep learning model.

In the same experimental environment, machine learning algorithms such as stochastic gradient descent (SGD), k-nearest neighbor (KNN), decision tree (DT), support vector machine (SVC), Gaussian NB and logistic regression (LR) are compared.

The experimental results are shown in Figure 9. The classification accuracy of the deep Gaussian process fault diagnosis model is much higher than other commonly used machine learning algorithms.

Compared with ensemble learning algorithms such as RandomForest (RF), AdaBoost, Bagging, ExtraTree (ET) and GradientBoosting(GB) in the same experimental environment, the experimental results are shown in Figure 10. The classification accuracy of the deep Gaussian process model for bearing faults is higher than that of the above ensemble learning algorithm, which is more suitable for the fault diagnosis of electrical machine bearings under large samples.

5. Conclusion

A fault diagnosis classification model of deep Gaussian process electrical machine rolling bearing based on particle swarm optimization is proposed. The basic components and structural parameters of the deep Gaussian process model are introduced. The parameter propagation formula based on expected propagation and Monte Carlo method is derived. The proposed model is trained and tested on the CWRU rolling bearing data set. The fault recognition rate of the trained model on the test set can reach 95 %, which is higher than that of other machine learning, ensemble learning and deep learning algorithms. It can better diagnose the electrical machine bearing fault and provide technical support for the safe operation of the motor.

Acknowledgment

This work was supported only by the Science Foundation of Ministry of Education of China (No.18YJCZH040).

References

[1] Zhang X Y, Bengio Y, Liu C L, "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark," Pattern Recognition, vol. 61, no. 61, pp. 348-360, 2017.

[2] Xie Z, Sun Z, Jin L, et al., "Learning Spatial-Semantic Context with Fully Convolutional Recurrent Network for Online Handwritten Chinese Text Recognition,". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no.8, pp. 1903-1917, 2018.

[3] J. Thiagarajan V B, "Designing Accurate Emulators for Scientific Processes using Calibration-Driven Deep Models," Nature Communications, vol. 11, no.1, pp. 5622–5632, 2020.

[4] Sagheer A K M, "Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems,". Sentific Reports, vol. 9, pp. 19038, 2019.

[5] Kim R G, Doppa J R, Pande P P, et al, "Machine Learning and Manycore Systems Design: A Serendipitous Symbiosis," Computer, vol. 51, no. 7, pp. 66–77, 2018.

[6] Y Aa Kobi E, Siegel P H, Vardy A, et al., "Multiple error-correcting WOM-codes," IEEE International Symposium on Information Theory. IEEE, vol. 58, no. 4, pp. 2220-2230, 2012.

[7] O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson and D. Dolev, "Gaussian belief propagation solver for systems of linear equations," 2008 IEEE International Symposium on Information Theory, pp. 1863-1867, 2008.

[8] D. Bickson, D. Dolev, O. Shental, P. H. Siegel and J. K. Wolf, "Gaussian belief propagation based multiuser detection," 2008 IEEE International Symposium on Information Theory, pp. 1878-1882, 2008.

[9] L XI*, Rui-Dong Wang, et al., "Multi-source neighborhood immune detector adaptive model for anomaly detection," IEEE Transactions on Evolutionary Computation, vol. 25, no. 3, pp. 582-594, 2021.

[10] Jingying Zhao, Hai GUO, Likun Wang, Min Han, "Computer Modeling of the Eddy Current Losses of Metal Fasteners in Rotor Slots of a Large Nuclear Steam Turbine Generator Based on Finite Element method and Deep Gaussian Process Regression," IEEE Transactions on Industrial Electronics, vol. 67, no. 7, pp. 5349-5359, 2020.

[11] L. Wang, H. Guo*, F. Marignetti, C. D. Shaver and N. Bianchi, "Cuckoo Search Algorithm for Multi-Objective Optimization of Transient Starting Characteristics of a Self-Starting HVPMSM," IEEE Transactions on Energy Conversion, vol. 36, no. 3, pp. 1861-1872, 2021.

[12] Guo, Hai, Qun Ding, Yifan Song, Haoran Tang, Likun Wang, and Jingying Zhao. 2020. "Predicting Temperature of Permanent Magnet Synchronous Motor Based on Deep Neural Network," Energies, vol. 13, no. 18, pp. 4782,2020.

[13] Jia F G L, Lei Y, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," Neurocomputing, vol. 272, no. 10, pp. 619–628, 2017.

[14] Damianou A C L N D, "Deep Gaussian Processes", Computer Science, pp. 207–215, 2012.

[15] Depeweg, S. et al., "Learning and Policy Search in Stochastic Dynamical Systems with Bayesian Neural Networks," Machine Learning, 2016.

[16] Cutajar K M P, Bonilla E V, "Random Feature Expansions for Deep Gaussian Processes," Machine Learning, no. 70, pp. 884–893, 2016.

[17] Bui T D, Hernandez-Lobato D, Li Y, et al., "Deep Gaussian Processes for Regression using Approximate Expectation Propagation," 33rd International Conference on Machine Learning, vol. 48, pp. 76-85,2016.

[18] Salimbeni H D M, "Doubly Stochastic Variational Inference for Deep Gaussian Processes," Neural information processing systems, no. 30, pp. 4588–4599, 2017.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, vol.4, pp. 1942-1948, 1995.

Hai Guo received the B.S. in Electronic Engineering from the Heilongjiang University, in 2000, the M.S. in Pattern Recognition and Intelligent Systems from the Kunming University of Science and Technology, in 2004, and the Ph.D. degree in Material Science from the Harbin University of Science and Technology (HUST), Harbin, China. Since 2010, he has been an Associate professor with the College of Computer Science and Engineering, Dalian Minzu University. He has authored over 30 articles in international journals and conference proceedings. His current research interests include pattern recognition and their applications.

Haoran Tang recrived the B.S. degrees from College of Mathematical, Shanghai Normal University, Shanghai, China, in 2018, the M.S. degree in Computer Science and Engineering, Dalian Minzu University in 2020.He current research interests include artificial intelligence, machine learning and electrical engineering.

Xin Liu received the B.S. degrees from College of Computer, Cangzhou Normal University, Hebei, China, in 2020. He is currently working towards the M.S. degree at Computer Science and Engineering, Dalian Minzu University.He current research interests include deep learning and artificial intelligence.

Jingying Zhao received the B.S. and M.S. degrees from School of Computer Science and Technology, Changchun University of Science and Technology, Jilin, China, in 2000 and 2003. Since 2013, she has been an Associate professor with the College of Computer Science and Engineering, Dalian Minzu University. She is currently working towards the Ph.D. degree at Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Liaoning, China. Her current research interests include pattern recognition and machine learning method and their applications.

Likun Wang(M'17) received the B.Sc., M.Sc., and Ph.D. degrees in electrical machinery and appliance from the Harbin University of Science and Technology (HUST), Harbin, China, in 2010, 2013, and 2015, respectively. Since 2017, he has been working as a Postdoctoral Fellow with the Institute of Electromagnetic and Electronic Technology, Harbin Institute of Technology, Harbin. Since 2018, he has been an Associate Professor of Electrical Machinery and Appliance with the College of Electrical and Electronic Engineering, HUST. His research interests include synthesis physical fields and dynamic operation mechanism of electrical machines and its system. Dr. Wang was the recipient of the first prize of science and technology progress of colleges and universities of Heilongjiang province in 2019.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This work was supported only by the Science Foundation of Ministry of Education of China (No.18YJCZH040).

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.