

Optimizing Back Propagation Neural Network Parameters to Judge Fault Types of Ball Bearings

Bo-Lin Jian,¹ Cai-Wan Chang-Jian,² Yu-Syong Guo,¹
Kuan-Ting Yu,¹ and Her-Terng Yau^{1*}

¹Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

²Department of Mechanical and Automation Engineering, I-Shou University, Kaohsiung 84001, Taiwan

(Received February 1, 2019; accepted December 13, 2019)

Keywords: ball bearing, back propagation neural network, support vector machine, fault detection, approximate entropy

When current technology keeps advancing, global machine tool manufacturers are gradually moving toward smart production lines. The ball bearing is an important fixed part of a rotating shaft; its key function is to bear the load acting on the shaft and maintain the center position of the shaft. If the bearing is damaged, there will be abnormal vibration, runout, and abnormal noise. Hence, the fault detection and recognition of the ball bearing are particularly important. The fault signal data of the ball bearing used in this study are obtained from the Case Western Reserve University (CWRU), and we establish a ball bearing status recognition model according to different signal-captured positions. First, the infinite impulse response (IIR) filter and approximate entropy (ApEn) are used to extract the features of the signals. Afterwards, the data extracted from the features are used for model establishment and training through a back propagation neural network (BPNN) and a support vector machine (SVM). In general, the SVM classification is better than the BPNN, but through a series of experimental methods, we confirmed that the optimal BPNN parameters of this sample, including training function, data training ratio, and the number of neurons, make the recognition rate of the BPNN higher than that of the general SVM, and the accuracy rate reaches 95%.

1. Introduction

Most of the automation equipment available in the market today requires fixing of the axis of rotation, and the ball bearing supports the mechanical rotating body. The ball bearing is used to reduce the friction coefficient during rotation and ensure the accuracy of rotation. Hence, the ball bearing is an essential part of transmission equipment. If the motor breaks down during high-speed operation, it will affect the performance of the machine. If the breakdown is severe, the machine must stop processing, thus affecting the yield rate. To improve this situation, in addition to enhancing the rigidity of the bearing, it is important to determine its operating status and detect and analyze its malfunction. Thus, when the ball bearing is abnormal, it should be replaced immediately.

*Corresponding author: e-mail: htyau@ncut.edu.tw
<https://doi.org/10.18494/SAM.2020.2605>

Various factors may cause the deformation of the inner and outer bearings, such as rotation under high-load and high-speed operation, and cause excessive ball wear.⁽¹⁾ Excessive ball wear causes a decrease in stability during transmission and affects the overall performance of the system. Many scholars have conducted fault diagnosis and working life analysis on the running condition of ball bearings.^(2,3) The ball bearing status is determined on the basis of standards established by the American National Standards Institute (ANSI) and International Organization for Standardization (ISO), and most research studies analyze the bearing status relative to the changes in current at the load drive terminal.^(4,5) Most studies detect the changes in audio frequency generated during bearing operation,^(6,7) the abnormal changes in bearing temperature,⁽⁸⁾ and the vibration changes generated during bearing operation,^(9,10) and there are many types of method of bearing fault detection and diagnosis. Since the changes in vibration generated by the rotation of the bearing are not large, and there is always external noise interference during data collection, in most studies, time domain signals are converted into frequency domain signals for spectral amplitude analysis, to carry out feature extraction processes, such as Fourier transform⁽¹¹⁾ and wavelet transform.⁽¹²⁾ However, Fourier transform cannot be used to describe the partial features of the signal, and the amount of wavelet transformation is huge; thus, it is difficult to perform real-time processing.

Technologies have been advancing in recent years, enabling significant enhancements in computer technology capabilities, such as the introduction of artificial neural networks (ANNs) for fault diagnosis.⁽¹³⁾ The advantages of ANNs include the parallel distributed processing capability, the ability to learn, robustness, and fault tolerance. As for disadvantages, ANNs need a large amount of training sample data for training and application. Therefore, in this study, we use the fault signal of the ball bearing, which was disclosed by Case Western Reserve University (CWRU) for discussion, adopt an infinite impulse response (IIR) filter to carry out signal decomposition,⁽¹⁴⁾ use approximate entropy (ApEn)⁽¹⁵⁾ for feature extraction, and bring the status types of the ball bearing into the classic classifier⁽¹⁶⁾ back propagation neural network (BPNN)⁽¹⁷⁾ and support vector machine (SVM)⁽¹⁸⁾ to compare the training, testing, and recognition rates of the models.

2. Methods

2.1 Experimental equipment and architecture

The simulation data used in this study adopt the fault signal database of the ball bearing proposed by CWRU. The platform is shown in Fig. 1; the red frame line a is the motor component, and its output horsepower (hp) is 2. The blue frame line b is the torque sensor, and the brown frame line c is the torque loading device component. The bearing at the drive end adopts the ball bearing model 6205-2RS JEM SKF (Svenska Kullagerfabriken, SKF). The detailed structural parameters of the machine are shown in Table 1, and the sampling frequency of the vibration signals is 12 kHz. Table 2 shows the detailed specifications of the bearing failure. The posted positions of the acceleration gauges are the drive- and fan-measured bearings. The signals captured by the acceleration gauges are fault signals from the balls and

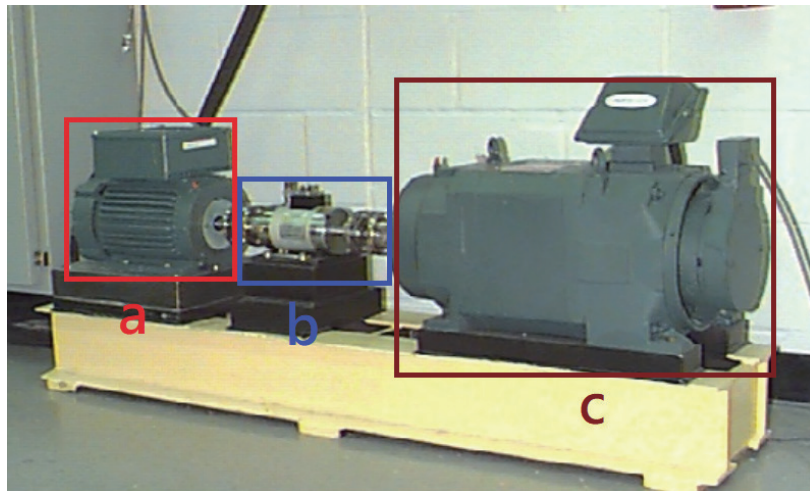


Fig. 1. (Color online) Fault simulation platform.

Table 1
6205-2RS JEM SKF bearing structural parameters.

Types	Parameters
Diameter of inner ring	25 mm
Diameter of outer ring	52 mm
Thickness	15 mm
Diameter of rolling elements	8 mm
Diameter of bearings	39 mm
Contact angle	0°

Table 2
Fault types of ball bearing.

Types	Parameters
Load of electric motor	0/1/2/3 (N·m)
Fault diameter of inner bearing	0.007/0.014/0.021 (in.)
Fault diameter of ball wear	0.007/0.014/0.021 (in.)

inner bearing. The PC hardware for analyses and recognition in this study is an Intel® Core™ i7-7700 CPU 3.60 GHz. The RAM is 16 GB, the display card is an NVIDIA GeForce GTX-1050Ti, the verification software is Matlab 2019a, and the toolboxes are machine and deep learning toolboxes.

The database used in this study was established from the vibration signals described above. We cut each piece of the original signal (indefinite length point, at least 60000 points or above) into four sections (each section contains 12000 points). The original classification has four different load statuses of the electronic motor, three different fault diameters, and two fault statuses, and each fault status is classified into inner bearing fault and ball wear fault. Hence, in this study, we classified the above fault statuses into four types: the fan-measured acceleration gauge signal for internal bearing failure, the drive-measured acceleration gauge signal for internal bearing failure, the fan-measured acceleration gauge signal for ball wear failure, and the drive-measured acceleration gauge signal for ball wear failure. These four types of fault have 192 pieces of data in total, and each type has 48 pieces. The experimental flow chart designed in this study is shown in Fig. 2. This process goes through data preprocessing first, which includes signal classification, segmentation processing, and ball bearing database establishment. Feature extraction is performed by IIR and ApEn. IIR is filtered from 170 to 150, 140 to 120, and 110 to 90 Hz. Therefore, the analysis signal increases to three pieces, that

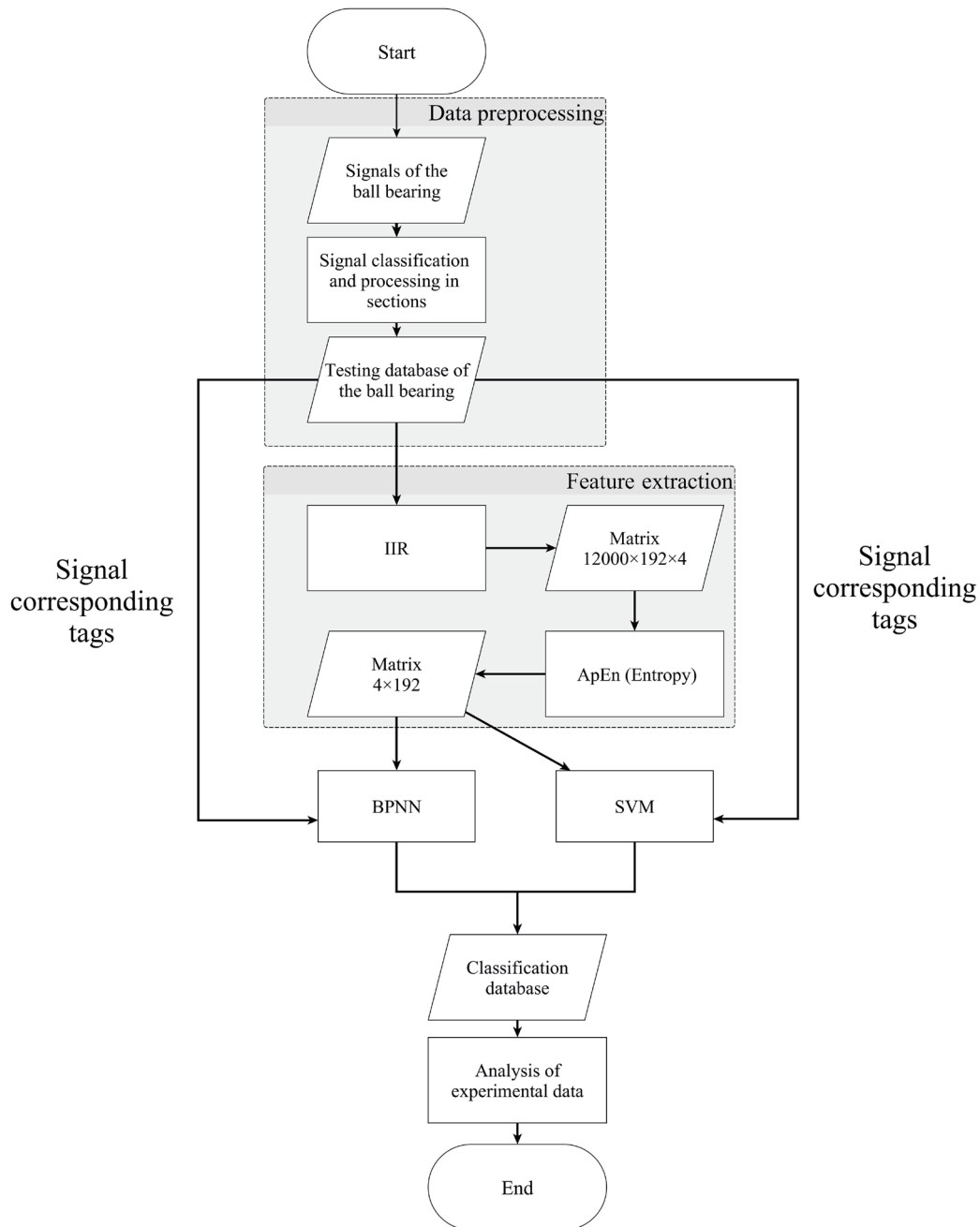


Fig. 2. Experimental flow chart.

is, four pieces in total when the original signal is included. ApEn can obtain feature values from each piece; thus, the final data is 4×192 . At the end of the experiment, we used BPNN and SVM classifier recognition rates for discussion, and determined the optimized parameters of the BPNN, to adopt a simple and easy-to-use model to establish bearing fault analysis.

2.2 Feature extraction

The experimental process is carried out according to Fig. 2. Feature extraction is implemented after obtaining the data from the database, and in this section, we illustrate the processing details of feature extraction. In this study, we use the IIR filter, also referred to as the IIR digital filter, which is a type of digital filter. Since there is a feedback loop in the IIR response to the pulse input signal, which is infinitely continuous, the IIR digital filter task is to change the spectrum of the input signal through computing. The IIR digital filter has the following advantages: it only needs low-order numbers to satisfy the filtering conditions and the computing efficiency is higher, but the computing has to use the entire segment of the signal continuously. Its input–output relationship is shown in the following difference equation:

$$\sum_{k=0}^m b_k x(p-k) - \sum_{k=1}^n a_k y(p-k). \quad (1)$$

In Eq. (1), $x(p)$ and $y(p)$ are the input and output of the filter, respectively, b_k and a_k are the coefficients of the filter, and m and n are the orders of the filter.

The above-mentioned IIR filter and ApEn are used to perform feature extraction. ApEn is a type of complexity index, which can indicate whether the data segment has similar results or conditions; if the data stands out abnormally, we can see clearly the changes in statistical results. At present, this method can be applied to some research studies related to frequency band complexity analysis. The following are demonstrations of ApEn:

The time series for N data points is shown in Eq. (2).⁽¹⁹⁾

$$\{d(1), d(2), d(3), \dots, d(N)\} \quad (2)$$

Select any m pieces of consecutive data from the series to rise to the m -dimensional vector, as shown in Eq. (3).⁽¹⁹⁾

$$X(i) = [d(i), d(i+1), \dots, d(i+m-1)], i = 1, 2, \dots, N-m+1 \quad (3)$$

The difference between the two vectors is defined as Eq. (4).⁽¹⁹⁾

$$L[X(i), X(j)] = \max_{k=0 \rightarrow m-1} [|d(i+k) - d(j+k)|] \quad (4)$$

Given the noise filter coefficient as r , record the number of data that satisfy the $d[x(i), x(j)] \leq r$ condition and compare the sum of this value with $N - m$, that is, the ratio of each similar number to the total number, as defined in Eq. (5).⁽¹⁹⁾

$$C_i^m(r) = \frac{1}{N-M+1} \sum_{i=1, i \neq j}^{N-m+1} |d[x(i) - x(j)]|$$

$$d(i, j) = \begin{cases} 1, & |d[x(i) - x(j)]| \leq r \\ 0, & |d[x(i) - x(j)]| \geq r \end{cases} \quad (5)$$

After extracting the features via ApEn, the size of the data changes from $12000 \times 192 \times 4$ to 4×192 .

2.3 BPNN

The BPNN is composed of a multilayer perceptron (MLP) and an error back propagation (EBP),⁽²⁰⁾ which is named the BP algorithm. The BPNN is a supervised learning network,⁽²¹⁾ which is the most representative and popular application among neural networks. It is divided into the forward propagation of network input and the back propagation of output error. If the actual output cannot meet the expected network output, the error message will be transmitted backward from the output layer to the input layer, and the weight will be modified continuously to reduce the error successively to achieve the purpose of learning.

Figure 3 shows the basic architecture of the BPNN. Yellow is the input layer, green is the hidden layer, and red is the output layer. The basic architecture of the neural network contains the processing unit, layer, and network. The processing unit is the most basic unit of computing, and several processing units with the same function are grouped into “layers”. These layers include the input, hidden, and output layers; these different layers form a type of neural network, and the neural network itself has both learning and recalling functions. The BPNN training process is shown in Fig. 4.

The BPNN is a typical multilayer network structure, which consists of an input layer, a hidden layer, and an output layer. The layers are interconnected by weights and deviations. The output of the previous layer is subjected to a nonlinear activation function⁽²²⁾ to process the input of the next layer; the practical equation is as shown in Eq. (6).

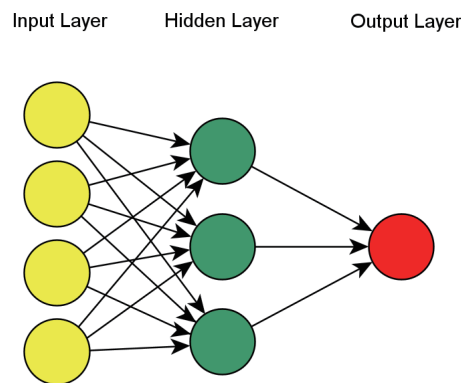


Fig. 3. (Color online) BPNN architecture.

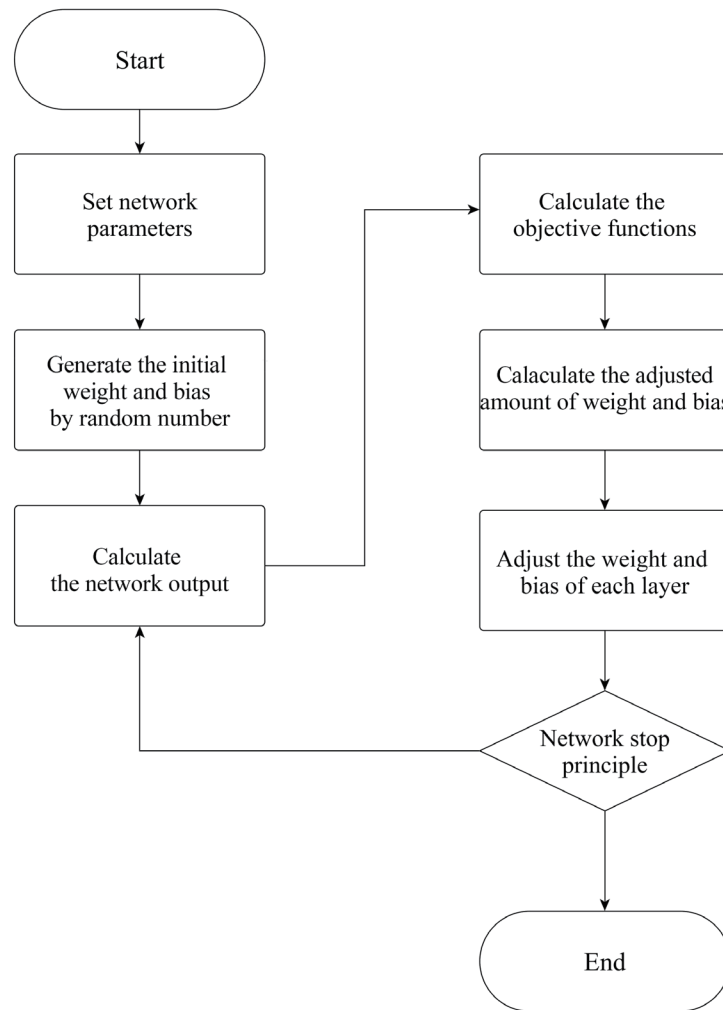


Fig. 4. BPNN flow chart.

$$S_j = f \left(\sum_{i=0}^{m-1} w_{ij} x_i + b_j \right) \quad (6)$$

$f()$ is the activation function named Sigmoid; its definition is shown in Eq. (7).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

In Eq. (6), S_j is the input of the j th hidden layer, w_{ij} is the weight that connects the weights of processing units in different layers, x_i is the input of the i th value, and b_j is a bias.

When there are errors between the predicted and actual values, the BPNN will return the errors to the weight and partial weight of each hidden layer's correction model. Then, the input of the training data according to each layer of new weight and bias computing is sent to the next

layer, and the error is corrected repeatedly until it falls in the range of acceptance. To reduce the difference between the network output and the target value, the target function shown in Eq. (8) is used.

$$E = \frac{1}{2} \sum_{i \in \text{outputs}} (y_{(i)} - \hat{y}_{(i)})^2 \quad (8)$$

In Eq. (8), $y_{(i)}$ is the actual value and $\hat{y}_{(i)}$ is the predicted value.

In this study, we use three different training number functions to train the neural network, as follows:

1. Trainlm: According to the Levenberg–Marquardt algorithm,⁽²³⁾ to update the weights and biases, Trainlm has the highest convergence speed for the medium-scale neural network model, but the disadvantage is that it needs a lot of memory to work.
2. Trainbr: According to Bayesian regularization⁽²⁴⁾ to minimize the combination of square error and weight, the network has a higher generalization ability, but it takes a longer training time.
3. Trainscg: According to the Scaled Conjugate Gradient,⁽²⁵⁾ more iterations are needed than other algorithms, but there is no need to perform a linear search in the iteration, so the amount of computing for each iteration is considerably reduced.

2.4 SVM

The SVM is a widely used classification technique in the field of machine learning and pattern recognition. It is the first technique that was used to deal with the binary classification issue. The learning method of the learning system of the SVM is used to calculate an optimal hyperplane through statistical learning theory. The data between different categories has the maximal margin and minimal misclassification, and the optimal hyperplane is constructed using the support vectors. H is the classified hyperplane; H_1 and H_2 are the samples that are closest to the classified hyperplanes in each classification and are parallel to the hyperplane of the classified hyperplane; the distance between them is the margin. The optimal classification hyperplane requires that the classified hyperplane not only correctly separates the two samples, but also maximizes the classification intervals.

While $H_1 = \mathbf{w} \times \mathbf{x} + b = 1$ and $H_2 = \mathbf{w} \times \mathbf{x} + b = -1$, and \mathbf{x} is the vector of the input features, \mathbf{w} is the normal vector of the hyperplane, b is a constant, and the interval between the two is $2 / \|\mathbf{w}\|$. Hence, to obtain the maximum interval, we look for the minimum of $\|\mathbf{w}\|^2 / 2$, which satisfies the conditional Eq. (9).

$$y_i ((\mathbf{w} \times \mathbf{x}_i) + b) \geq 1, i = 1, 2, \dots, n \quad (9)$$

To sum up the above, we have the following objective function:

$$\begin{aligned} & \min \left(\frac{1}{2} \mathbf{w}^2 \right) \\ & \text{subject to } \begin{cases} y_i [(\mathbf{w} \times \mathbf{x}_i + b)] - 1 \geq 0 \quad (i = 1, 2, \dots, n) \\ \mathbf{w}_i = a_1 y_1 \mathbf{x}_1 + a_2 y_2 \mathbf{x}_2 + \dots + a_n y_n \mathbf{x}_n. \end{cases} \end{aligned} \quad (10)$$

The Lagrange method⁽²⁶⁾ ($\mathbf{w} = a_1 y_1 \mathbf{x}_1 + a_2 y_2 \mathbf{x}_2 + \dots + a_n y_n \mathbf{x}_n$) is used to optimize the classification hyperplane; then, the issue of the binary classification of the SVM can be expressed as

$$L(w, b, a) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N a_i [y_i (\mathbf{w} \times \mathbf{x}_i + b) - 1]. \quad (11)$$

The SVM has the following advantages: it has good applicability in high-dimensional space and can effectively process data with more variables than the number of samples. Moreover, some subsets can be available at the training model with no need of plenty of memory.

3. Results

3.1 Comparison of BPNN optimal training function

The internal parameter setting of the BPNN has no fixed basis and norms; therefore, in this study, we compare the different training functions in terms of the number of neurons and the amount of data training, and then select the best parameters. Under the condition of the same parameters, the three different training functions, Trainlm, Trainbr, and Trainscg, are trained by the CPU, and we also use the GPU to train Trainscg and to determine whether the computing between the GPU and the CPU will indirectly affect the experiment. Four different results are shown in Figs. 5–8.

On the basis of the results shown in Figs. 5–8 we can tell that the accuracy rate of Trainbr is higher than those of Trainlm and Trainscg. The results shown in Figs. 9–13 indicate that from the viewpoint of computing time, using a single CPU for low-dimensional computing is more suitable than using a single GPU to perform computing operations. Moreover, the experimental

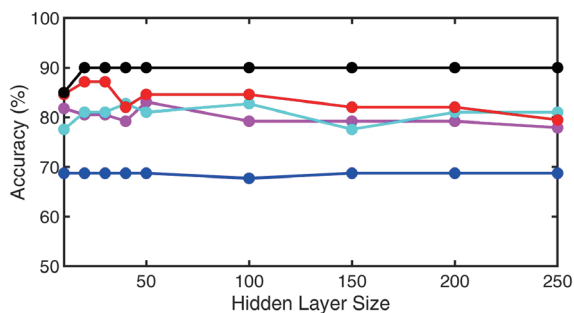


Fig. 5. (Color online) Use of GPU to train Trainscg.

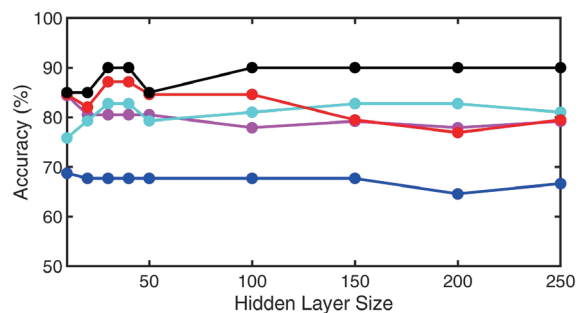


Fig. 6. (Color online) Use of CPU to train Trainscg.

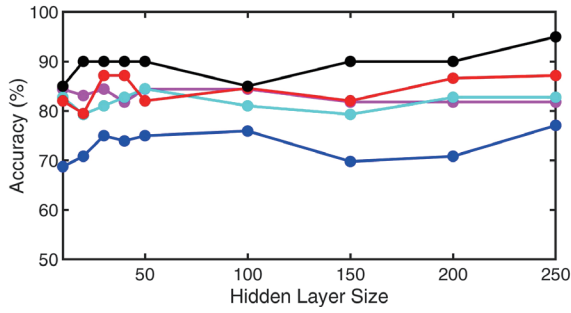


Fig. 7. (Color online) Use of CPU to train Trainbr.

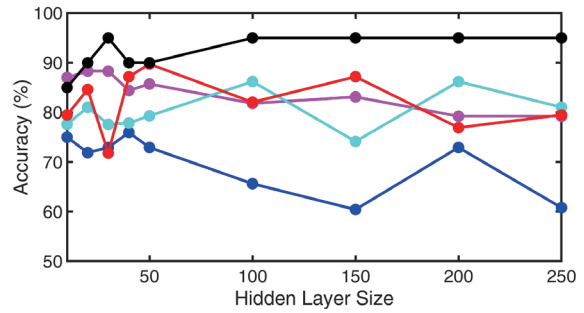


Fig. 8. (Color online) Use of CPU to train Trainlm.

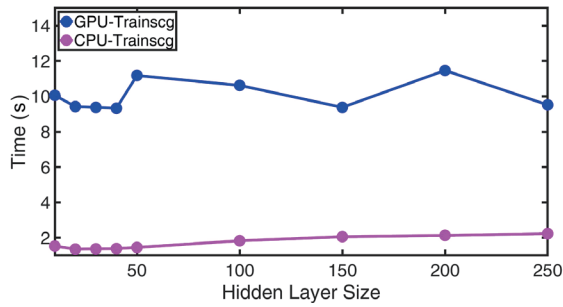


Fig. 9. (Color online) Computing time of GPU and CPU with 50% ratio training.

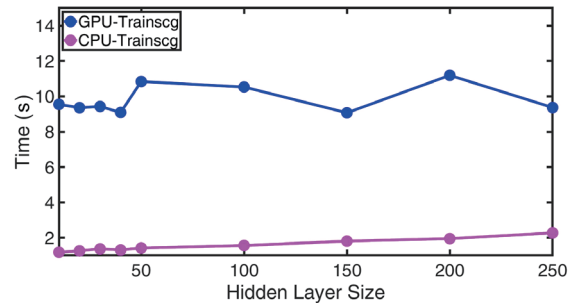


Fig. 10. (Color online) Computing time of GPU and CPU with 60% ratio training.

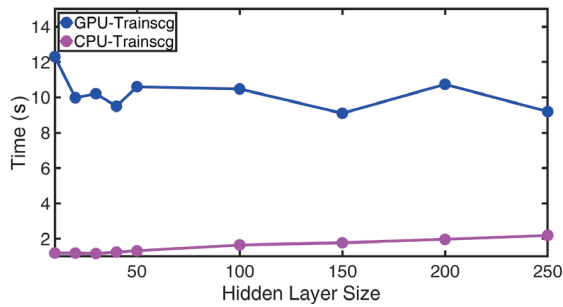


Fig. 11. (Color online) Computing time of GPU and CPU with 70% ratio training.

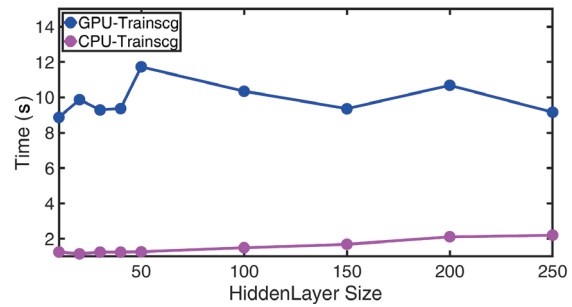


Fig. 12. (Color online) Computing time of GPU and CPU with 80% ratio training.

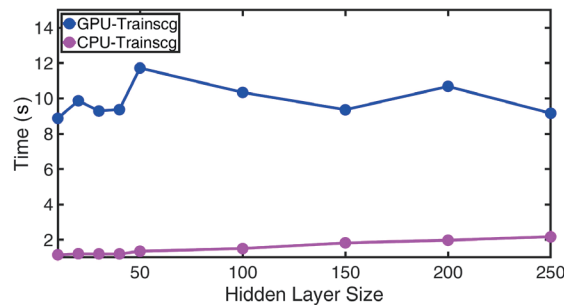


Fig. 13. (Color online) Computing time of GPU and CPU with 90% ratio training.

results show that the recognition abilities of the three different training functions for the ball bearing database from high to low are Trainbr, Trainlm, and Trainscg. Therefore, we will adopt CPU and Trainbr training methods in our subsequent analysis.

3.2 Best data training ratio of BPNN

After comparing the training functions in Sect. 3.1, we fixed the data training amount of Trainbr to BPNN at 90% and the test amount at 10%, the data training amount at 80% and the test amount at 20%, the data training amount at 70% and the test amount at 30%, the data training amount at 60% and the test amount at 40%, the data training amount at 50% and the test amount at 50%, and compared the results. The experimental results are shown in Fig. 7.

We can tell from Fig. 7 that when the training rate is 90% and the other training ratios have a very high recognition rate, the recognition rate is up to 95%, and the lowest is 85%. After the analysis, we understood that the recognition abilities from high to low of five types of data training ratios for the ball bearing fault are 90, 80, 60, 70, and 50%. Therefore, in the follow-up experiment we adopt a training ratio of 90% for exploration.

3.3 Best number of neurons in hidden layers of BPNN

The results described in Sects. 3.1 and 3.2 show that the highest recognition ability was observed when the training function is Trainbr and the neural network data training ratio is 90%. Afterwards, we compared the relative recognition rates of different numbers of neurons (10, 20, 30, 40, 50, 100, 150, 200, and 250) and the corresponding accuracy rates of the numbers of neurons are shown in Fig. 7.

We can tell from Fig. 7 that when the number of neurons is 250, the corresponding recognition rate is higher than those for other numbers of neurons, and the recognition rate tends to increase when the number of neurons is 150. To summarize Sects. 3.1–3.3, we can tell that the highest recognition ability was observed when the training function is Trainbr, the neural network data training ratio is 90%, and the number of neurons is 250, and these parameters are used to recognize the ball bearing fault. From Table 3, we can verify that in this study, the BPNN has a very high recognition rate when the training ratio is 90% and the number of neurons is 250.

3.4 Comparison of classification results between SVM and BPNN

In this study, we adopt the SVM classifier to verify the classification accuracy rate; we capture 192 pieces of data by using the fault signals of the drive-measured ball and the inner bearing for analysis. When the training set is 90% and the test set is 10%, and when the training

Table 3
Corresponding accuracy rates of various numbers of neurons.

The number of neurons is 250			
Training set (%)	Test set (%)	Time (s)	Accuracy rate (%)
90	10	700.46	95
80	20	661.87	87.179
70	30	689.27	82.758
60	40	694.40	81.818
50	50	567.07	77.083

Table 4
Classification accuracy rate of classifiers.

		Test 1	Test 2	Test 3	Test 4	Test 5
Classifiers	Training set (%)	90	80	70	60	50
	Test set (%)	10	20	30	40	50
SVM	Classification accuracy rate (%)	84.375	84.375	81.25	81.25	81.25
BPNN (optimization)	Classification accuracy rate (%)	95	87.179	82.758	81.818	77.083

set is 80% and the test set is 20%, the SVM has a nearly 85% recognition rate, and the lowest recognition rate of the training sets is 81.25%. As for classifiers, the recognition performance of the SVM is better than that of the BPNN, but in this study, the recognition performance of the BPNN is better than that of the SVM through parameter adjustment and setting. Table 4 shows a comparison of the SVM and BPNN under the same training set and test set parameters. We can tell from the following table that the BPNN and SVM have the best recognition rates when the training set is 90%.

4. Conclusions

In this work, we analyzed the fault signals of the ball bearing disclosed by CWRU. After the preprocessing and feature extraction of the fault signals from the drive-measured ball and inner bearing, we performed the fault classification of the signals with the BPNN and SVM, and establish the optimal model by parameter selection. The research results show that when the BPNN uses Trainbr as the training function, the data training amount is 90%, and the number of neurons is 250, a very high classification ability is achieved. Afterwards, we used the SVM to carry out classification verification and found that the SVM has a high classification accuracy rate when the data training amount is 90%. Thus, we can tell that both the BPNN and the SVM have a very high recognition rate for the data analyzed in this study. In the final comparison of classifiers, we showed that through parameter adjustment, the classification accuracy rate of the BPNN is higher than that of the SVM.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology of the ROC, Taiwan, under Contract No. Most 107-2218-E-167-004.

References

- 1 C. Hu, B. D. Youn, P. Wang, and J. Taek Yoon: Reliab. Eng. Syst. Safety **103** (2012) 120. <https://doi.org/10.1016/j.res.2012.03.008>
- 2 J. Zhu, N. Chen, and W. Peng: IEEE Trans. Ind. Electron. **66** (2019) 3208. <https://doi.org/10.1109/TIE.2018.2844856>
- 3 Y. Qian and R. Yan: IEEE Trans. Instrum. Meas. **64** (2015) 2696. <https://doi.org/10.1109/TIM.2015.2427891>
- 4 M. J. Devaney and L. Eren: IEEE Trans. Instrum. Meas. Mag. **7** (2004) 30. <https://doi.org/10.1109/MIM.2004.1383462>

- 5 R. R. Schoen, T. G. Habetler, F. Kamran, and R. G. Bartfield: IEEE Trans. Ind. Appl. **31** (1995) 1274. <https://doi.org/10.1109/28.475697>
- 6 B. Van Hecke, D. He, and Y. Qu: J. Vib. Acoust. **136** (2014). <https://doi.org/10.1115/1.4028322>
- 7 H. Huang, F. Liu, L. Geng, Y. Liu, Z. Ren, Y. Zhao, X. Lei, and X. Lu: IEEE Access **7** (2019) 87410. <https://doi.org/10.1109/ACCESS.2019.2924832>
- 8 L. A. Gupta and D. Peroulis: IEEE Sens. J. **13** (2013) <https://doi.org/2292.10.1109/JSEN.2013.2251970>
- 9 Z. Huo, Y. Zhang, P. Francq, L. Shu, and J. Huang: IEEE Access **5** (2017) <https://doi.org/10.1109/ACCESS.2017.2661967>
- 10 H. Ahmed and A. K. Nandi: IEEE Access **6** (2018) <https://doi.org/10.1109/ACCESS.2018.2865116>
- 11 A. Rashid and O. Hasan: J. Symbolic Comput. **90** (2019) 65. <https://doi.org/10.1016/j.jsc.2018.04.004>
- 12 A. Anuragi and D. Singh Sisodia: Biomed. Signal Process. Control **52** (2019) 384. <https://doi.org/10.1016/j.bspc.2018.10.017>
- 13 G. Hong, L. B. Jack, and A. K. Nandi: IEEE Trans. Syst. Man Cybern. Part B Cybern. (Cybernetics) **35** (2005) 89. <https://doi.org/10.1109/TSMCB.2004.841426>
- 14 P. Upadhyay, R. Kar, D. Mandal, and S. P. Ghoshal: Int. J. Eng. Sci. Technol. **17** (2014) 8. <https://doi.org/10.1016/j.jestch.2014.02.002>
- 15 S. M. Pincus: Proc. Natl. Acad. Sci. **88** (1991) 2297. <https://doi.org/10.1073/pnas.88.6.2297>
- 16 C. Lee and S. Woo: Pattern Recogn. **88** (2019) 210. <https://doi.org/10.1016/j.patcog.2018.11.024>
- 17 Z. Wang, J. Zhang, J. Wang, X. He, L. Fu, F. Tian, X. Liu, and Y. Zhao: Optik **179** (2019) 780. <https://doi.org/10.1016/j.ijleo.2018.09.161>
- 18 M. A. S. Altobi, G. Bevan, P. Wallace, D. Harrison, and K. P. Ramachandran: Int. J. Eng. Sci. Technol. **22** (2019) 854. <https://doi.org/10.1016/j.jestch.2019.01.005>
- 19 Y. Xu, X. Ma, and X. Ning: Physica A **376** (2007) 401. <https://doi.org/10.1016/j.physa.2006.10.055>
- 20 C. R. Rekha, V. U. Nayar, and K. G. Gopchandran: Optik **172** (2018) 721. <https://doi.org/10.1016/j.ijleo.2018.07.090>
- 21 C. Chen, Y. Liu, M. Kumar, J. Qin, and Y. Ren: Comput. Ind. Eng. **135** (2019) 757. <https://doi.org/10.1016/j.cie.2019.06.052>
- 22 V. S. Bawa and V. Kumar: Expert Syst. Appl. **120** (2019) 346. <https://doi.org/10.1016/j.eswa.2018.11.042>
- 23 S. Mammadli: Procedia Comput. Sci. **120** (2017) 602. <https://doi.org/10.1016/j.procs.2017.11.285>
- 24 M. Aucejo and O. De Smet: Mech. Syst. Signal Process. **126** (2019) 98. <https://doi.org/10.1016/j.ymssp.2019.02.021>
- 25 H. Bouwmeester, A. Dougherty, and A. V. Knyazev: Procedia Comput. Sci. **51** (2015) 276. <https://doi.org/10.1016/j.procs.2015.05.241>
- 26 K. Zhang, Q. Huang, Y. Zhang, J. Song, and J. Shi: Mech. Mach. Theory **134** (2019) 512. <https://doi.org/10.1016/j.mechmachtheory.2019.01.012>

About the Authors



Bo-Lin Jian received his B.S. degree from the Department of Electrical Engineering National Formosa University in 2009, his M.S. degree in materials science and engineering from the National Taiwan University of Science and Technology in 2011, and his Ph.D. degrees from the Department of Aeronautics and Astronautics, National Cheng Kung University in 2017. He is currently an assistant professor at the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. His current research interests include signals, image processing, machine learning, and control systems.



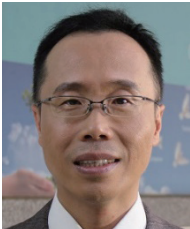
Cai-Wan Chang-Jian received his B.S. degree from Yuan Ze University, Taiwan, in 1999 and his M.S. and Ph.D. degrees from National Cheng-Kung University (Taiwan) in 2001 and 2006, respectively. From 2006 to 2010, he was an assistant professor at I-Shou University, Taiwan and he was an associate professor at I-Shou University from 2010 to 2013. Since 2013, he has been a professor at I-Shou University. His research interests are in nonlinear dynamic analysis and control, lubrication, and performance analysis of nanomaterials.



Yu-Syong Guo received his B.S. degree from the National Chin-Yi University of Technology, Taichung, Taiwan, in 2017, where he has been working toward his M.S. degree in electrical engineering since 2017. His research interests include integrated systems, Internet of Things, and chaotic systems.



Kuan-Ting Yu received his B.S. degree from the National Chin-Yi University of Technology, Taichung, Taiwan, in 2018, where he is currently pursuing his M.S. degree in electrical engineering. His research interests include nonlinear system analysis and control.



Her-Terng Yau received his B.S. degree from National Chung Hsing University, Taichung, Taiwan, in 1994, and his M.S. and Ph.D. degrees from National Cheng Kung University, Tainan, Taiwan, in 1996 and 2000, respectively, all in mechanical engineering. He is currently a professor at the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. His research interests include energy converter control, system control of mechatronics, nonlinear system analysis, and control. He has authored more than 150 research articles on a wide variety of topics in mechanical and electrical engineering.