Modulated Gabor filter based deep convolutional network for electrical motor bearing fault classification and diagnosis

Shahabodin Afrasiabi | Mohammad Mohammadi | Mousa Afrasiabi | Benyamin Parang

Department of Power and Control Engineering, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
(Email: shpower77@yahoo.com; musa.afra@shirazu.ac.ir)

Correspondence
Mohammad Mohammadi, Department of Power and Control Engineering, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran.
Email: m.n.mohammadi@shirazu.ac.ir

Abstract
The high applicability of the electrical motor has led to gain attention in condition monitoring to diagnosis the most common type of fault in this machine, bearing element. The emergence of deep neural networks (DNN) provides the opportunity to design a network for early bearing fault diagnosis with high speed and without any additional feature extraction technique. However, robustness against the noise and some deficiencies in fully capturing features are still challenging issues. To resolve this problem, this paper proposes a one module Gabor filter based convolutional neural networks (CNN), namely Gabor convolutional neural network (GCNN), for bearing fault detection and classification. GCNN is a modulated Gabor filter to enhance the ability in capturing temporal features as well as enhance understanding spatial features with fewer parameters and higher robustness against noises and can be considered as a computational efficient deep structure. The simulation results of the bearing fault detection/classification are studied on two different experimental prototypes, including case Western Reserve University (CWRU) and Paderborn University (Paderborn) datasets. The superiority of this method is shown by comparison with accelerated CNN (ACNN), adaptive CNN, standard CNN, support vector machine (SVM), learning vector quantisation (LVQ), and feedforward neural network (FFNN).

1 | INTRODUCTION

The electrical motor is the most widely used rotatory electrical machines in the industries and transportation systems. Thus, it is a crucial task to detect the failure of industrial motors and thereby prevent wasting the huge amount of money and time, which should be devoted to repairing and maintenance of this type of electrical machine [1, 2]. Based on the official report [3], about 30–40% of failures in the industrial motors concerned the bearing. Early bearing fault diagnosis is crucial to prevent serious failures, although they might not lead to an immediate breakdown. Thus, designing a suitable structure for the bearing fault detection, which performs fast and accurate is essential.

Three major groups of methods have been presented in the previous literature, that is, model-based, signal-based, and data-driven methods. The main principle of model-based methods is the difference between the mathematical model of electrical motors and corresponding components with measurement signals. Although model-based methods provide fast performance, accurate modelling of the complex components of industrial motors is too difficult [4]. Moreover, one-to-one mapping between the mathematical model and measurements is not practical. In contrast with model-based methods, signal-based methods are independent of accurate mathematical modelling. Signal based methods are carried out based on the analysis of the measurement signals, regardless of whether they are electrical or not, in time or frequency domain [3]. Signal based methods are highly dependent on the predefined threshold and generally suffer from high sensitivity to noise [5]. To resolve the drawbacks of the signal and model-based methods, data-driven methods are widely used in the previous investigations. Data-driven methods are a potential solution to handle the non-linear and complex nature of industrial motors without any requirement to accurate model and predefined threshold.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2021 The Authors. IET Science, Measurement & Technology published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

wileyonlinelibrary.com/iet-smt
IET Sci. Meas. Technol. 2021;15:154–162.
terms of structure, data-driven methods are divided into shallow and deep based structures. Shallow-based structures usually use raw or analysed signals in the time/frequency domain using, for instance, artificial neural networks (ANN) [6], support vector machine (SVM) [7], learning vector quantisation (LVQ) [8], fuzzy logic (FL) [9] etc. The shallow based structures are usually unusable to extract discriminative features from the raw data due to their small hypothesis space. Moreover, feature engineering on the raw data and additional feature extraction is computationally expensive and cannot be expressed as a general solution in the fault detection/classification issues [10].

Deep learning emergence as an evolutionary concept in machine learning has attended a large number of researchers due to the ability in capturing complex and nonlinear features from the raw time series. Deep learning can enhance learning ability and improve the accuracy and reliability in the STLFF.

Deep neural networks (DNN) are also divided into several main categories, involving auto-encoder (AE), deep Boltzmann machine (DBM), recurrent neural network (RNN), generative adversarial network (GAN), and convolutional neural network (CNN) [3]. Although AE and DBM improve the capability of learning from raw data through a dimensional reduction procedure, the main disadvantage of AE and DBM is a disability in understanding long sequences in both spatial and temporal features [11]. GAN as an artificial feature generation network is only used in encapsulated DNN-based structure. GAN is unable to maintain stability during the training process due to dependency on Nash equilibrium and is difficulty in learning from discrete data like signals obtained from the digitalised recorders [3]. RNN-based structures such as long short-term memory (LSTM) and gated recurrent unit (GRU) are utilised in the bearing fault identification in [12, 13], respectively. In memory-based RNNs, the gates control information throughout the time series. If the feature is important, the DNN-based method memorises it as a high-weighted feature, otherwise, it forgets or resets it. RNN-based structures including LSTM and GRU suffer from the disability in the spatial features [14]. CNN is a widely used deep structure method in time series analysis. Despite the great performance in capturing spatial features, CNN is unable to fully realise the temporal feature, especially in long-tailed time series associated with high variations [5]. Thus, with the huge effort of the researchers in the field of bearing fault identification in industrial motors, the problem is still challenging. On the other hand, vibration signals measured by accelerometers are typically the basis of the monitoring of the industrial motors, while they are adversely influenced by environmental noises, resonance etc. Therefore, the robust performance of the bearing fault detection method in noisy conditions is another challenge in this field.

To resolve these problems, we aim to develop a deep structure in this paper that has a powerful structure to learn features from the raw data in fault detection/classification of the bearing fault. To this end, a Gabor convolutional neural network (GCNN) is proposed in this paper. GCNN is one module-based network, which benefits from Gabor filter with less parameter in comparison with the standard CNN and able to handle the temporal features by the capacity to transform with convolutional layer redesigning. In contrast with hybrid methods that add additional feature extraction to only generate features, GCNN benefits from Gabor functions to modulate convolutional layers based Gabor filter and promote understanding ability from the raw signal such as vibration by capturing temporal features and spatial frequency and localisation. Furthermore, GCNN learns features automatically with fewer parameters than the standard CNN. To verify the effectiveness of the proposed GCNN method in the bearing fault diagnosis and classification in the induction motors, it is compared with accelerated CNN and the standard CNN as deep learning-based fault detection and SVM, LVQ, and FFNN as shallow based structures in the fault detection and classification. Furthermore, to verify the transferability of the proposed GCNN detector/classifier method, two experimental datasets, including case western reserve University (CWRU) and Paderborn University (PU) are utilised in the numerical results.

We can summarise the contribution of the paper as follows:

- Designing a time series detection with the integration of Gabor filter and standard CNN to provide a network in form of the one module deep network
- GCNN is able to perform faster than standard CNN with fewer parameters
- Providing the deep structure with a high level of robustness against noises

The rest of the paper is organised as follows: Section 2 provides a background from the modulated Gabor filter based convolutional layer. The experimental dataset and input data preparation are introduced in Section 3. Section 4 describes the structure of the designed GCNN for bearing fault detection. Section 5 discusses the results and comparisons and Section 6 concludes the paper.

2 | BACKGROUND OF THE DEEP CONVOLUTIONAL GABOR NETWORK IN FAULT CLASSIFICATION/DETECTION

In a binary/multi-class time series classification problem, such as bearing fault detection and classification, the time series dataset is formed as $X^r$ and $Y^r$, which represent input and output. In bearing fault detection of the induction motor, $Y^r_{FD} = (j^{FD}) \in B^{1x2}$, where $F_D = \{H, F\}$. In the fault classification problem, $Y^r_{FD} = (j^{FD}) \in B^{1x3}$, while $F_D = \{F_h, F_l, F_o\}$. Note that $F, H, F_h, F_l, F_o$ stand for faulty, healthy, ball fault, inner race fault, and outer race fault, respectively. In addition, the input dataset is shown by $X^r = \{x_i^r\}$, where $i = \{1, ..., N\}$ and the members of the input dataset are vibration signals measured by the experimental prototype. The main aim of this paper is to construct a network to map $Y^r$ with a minimised difference with $Y^r$. Therefore a deep Gabor network as a potential solution is proposed.

Deep Gabor network is mainly based on the idea of the inherit combination of the Gabor filter and convolutional neural
network, which is originally presented in [15]. Gabor filter is a
signal processing technique to extract features in both the time
and frequency domain. Adding Gabor filter as a feature extrac-
tion technique might improve the accuracy, whereas similar to
other combinational methods, it can increase computation time
and cannot be considered as a general solution in the induc-
tion motor bearing fault diagnosis. To this end, the designed
deep Gabor network is a modified form of the standard convo-
lutional layers by forming inherited Gabor convolutional layers
and thereby reducing the network parameters as well as enhanc-
ing the performance in the bearing fault classification of the
induction motors.

GCNN enhances the accuracy of the bearing fault classifi-
cation/detection problem as well as computational efficiency.
To address the designed GCNN in an industrial fault detection
problem, in the following subsections, firstly Gabor filter inte-
gration with CNN is explained. Then, the GCNN structure for
bearing fault classification is described.

2.1 Incorporating Gabor filter in CNN

Each Gabor filter has a \( D \) direction and \( \delta \) scale. To integrate
Gabor filter into the CNN, the feature maps are encoded in the
learned layers and the scale of the feature maps are embedded
into different layers. The learning ability of CNN in understand-
ing the features is improved by capturing direction and scale
through Gabor filters.

In standard CNN, the set of weights are shaped as a tensor
\( (y_{\text{out}}, y_{\text{in}}, W, W) \), where \( y_{\text{in}}, y_{\text{out}}, k_{\text{wi}}, k_{\text{le}} \) represent the input
feature map, output feature map, width, and length of convolu-
tional kernels. In GCNN, the weights are reformed as \( (y_{\text{out}}, y_{\text{in}},
D, k_{\text{wi}}, k_{\text{le}}) \). It is essential to maintain the quantity of the feature
map during the convolution process to form an inherent Gabor
filter-based set of convolutional layers. Hence, the number of
the directions \( D \) is selected to inherit Gabor filter into the con-
volution layer and compose a Gabor filter based convolution
layer using \( D \) Gabor filters for a given scale \( \delta \) as:

\[
f_i(d, \delta) = f_{i\text{out}} \odot G(d, \delta),
\]

where \( f_i(d, \delta) \), \( f_{i\text{out}} \), and \( G(d, \delta) \) represent the learned kernel, and a
set of Gabor filters with different \( d, \delta = (1, \ldots, D) \) directions
and \( s, \delta = (1, \ldots, \delta) \) scales, respectively. Note that symbol \( \odot \)
represents element-wise product operation between the real part
of Gabor filters.

The set of the Gabor filters are defined as:

\[
G(d, \delta) = \frac{\|k(d, \delta)\|}{4\pi^2} \exp\left(-\frac{\|k(d, \delta)\|^2}{8\pi^2}\right),
\]

\[
(\exp(j, k(d, \delta)z) - \exp(2\pi^2 j, k(d, \delta)))
\]

where \( z = (X, Y) \) and \( k(d, \delta) \) is defined as:

\[
k(d, \delta) = \sqrt{\frac{(\pi)}{2} \exp\left(j, \delta \frac{\pi}{D}\right)}.
\]

Consequently, the modulated Gabor convolutional network is:

\[
f_i(d) = (f_i(d, 1), \ldots, f_i(d, \delta))
\]

It is true that each Gabor filter based convolutional layer
needs feature maps with \( (y_{\text{out}}, y_{\text{in}}, D, k_{\text{wi}}, k_{\text{le}}) \) dimension; how-
ever, only feature maps with \( (D, k_{\text{wi}}, k_{\text{le}}) \) are stored due to
utilisation of the Gabor filter [15]. Therefore, the Gabor filter
is inherently incorporated with CNN by reducing the network
parameters and thereby reducing the computational burden of
the proposed method, while Gabor filter enhances the learning
ability. Figure 1 shows the procedure to modulate the Gabor
filter into the convolutional neural layers and produce the
Gabor based convolutional layers.

3 EXPERIMENTAL DATASETS AND
PREPARATION

To implement a bearing fault detection/classification the input
dataset should be prepared for the process. In the following sub-
sections, the two different experimental datasets and how they
are rearranged for the training and testing process is presented.

3.1 Experimental datasets

As mentioned before, in this paper two different datasets includ-
ing CWRU and PU datasets are used.

The CWRU is one of the most reliable experimental datasets
which is available online for the public in [16]. The experimental
setup of the CWRU dataset is illustrated in Figure 2.

As can be seen from Figure 2, an induction motor connected
to a dynamometer and a torque transducer/encoder is placed
between the induction motor and dynamometer. The raw vibration signals are recorded by accelerometers, which are located at both ends of the induction motor. Four different major classes including healthy, inner race fault, outer race fault, and ball fault of the induction motors. The data was sampled with 12 kHz frequency and three different diameters including 0.007 inch, 0.014 inch, and 0.021 inch for generating a large number of the fault data is taken into account. Considering one healthy performance of the induction motor, nine different fault conditions are considered. Overall, 120,000 raw vibrations signal from ten different conditions are available in the CWRU dataset.

The second dataset, PU is also available for the public and includes three different datasets, that is, health conditions, artificially generated bearing fault, and actual bearing fault. In this paper, we use the vibration signals in the healthy and actual bearing fault conditions. The experimental setup of the PU is depicted in Figure 3. As can be realised from Figure 3, the experimental setup of the PU dataset consists of five main components, including electric motor, torque measurement shaft, rolling bearing test module, flywheel, and a load motor. The raw data are measured at different load levels, radial forces, and speeds, which are shown in Table 1.

In the PU dataset, 15 different files contain healthy conditions, outer race bearing fault, and inner race bearing faults. The healthy conditions are labelled with K0-series (001–005), while KA-series (04, 15, 16, 33, 30) and KI-series (04, 14, 16, 18, 21) show outer race and inner race faulty conditions, respectively. To produce this dataset, each experiment is repeated 20 times and the sampling frequency is 64 kHz. More information about the PU dataset is provided in [17].

### Table 1: Different operational condition in the PU dataset

| Condition Number | Rotational Speed (rpm) | Load Level (Nm) | Radial Force (N) |
|------------------|------------------------|-----------------|-----------------|
| #1               | 1500                   | 0.7             | 1000            |
| #2               | 900                    | 0.7             | 1000            |
| #3               | 1500                   | 0.1             | 1000            |
| #4               | 1500                   | 0.7             | 400             |

### 3.2 Input data organisation

The designed GCNN structure performs based on a 2D input dataset, while the raw vibration signals are fundamentally 1D signals. To convert them into 2D raw signals, the input dataset is reshaped as the:

\[
X = \begin{bmatrix}
v_1 & v_2 & \ldots & v_{K-k+1} \\
v_2 & v_3 & \ldots & v_{K-k+2} \\
\vdots & \vdots & \ddots & \vdots \\
v_k & v_{k+1} & \ldots & v_K 
\end{bmatrix}
\]

As the input dataset only consists of the raw vibration samples, we denote \( v \) to each discrete measured data. The input dataset \( X \) is decomposed of multiple sequences with \( K \) samples and the embedding dimension is \( k \). The 2D input dataset size is \( k \times (K - k + 1) \). To train the designed GCNN network, the raw vibration time series is decomposed into several subsequences, as depicted in Figure 4. Each subsequence is composed of \( K_{tri} \) samples with \( L \) length of the sliding windows, obtained as:

\[
K_{tri} = \frac{S_v - S_{sub}}{L},
\]

where \( S_v \) and \( S_{sub} \) show the number of samples at each vibration samples and subsequences, respectively. The overlap is only considered in the training process and for the testing process, we have not considered the overlap in the dataset.
TABLE 2  Training and testing data size in the CWRU dataset

| Condition | Training Data | Testing Data |
|-----------|---------------|--------------|
| Health    | 1980          | 75           |
| Ball fault-0.007 inch | 1980 | 75 |
| Ball fault-0.014 inch | 1980 | 75 |
| Ball fault-0.021 inch | 1980 | 75 |
| Inner race fault-0.007 inch | 1980 | 75 |
| Inner race fault-0.014 inch | 1980 | 75 |
| Inner race fault-0.021 inch | 1980 | 75 |
| Outer race fault-0.007 inch | 1980 | 75 |
| Outer race fault-0.014 inch | 1980 | 75 |
| Outer race fault-0.021 inch | 1980 | 75 |

TABLE 3  Training and testing data size in the PU dataset

| Condition | Files   | Training Data | Testing Data |
|-----------|---------|---------------|--------------|
| Health    | K0-series | 4800         | 1200         |
| Inner race fault | KI-series | 4800 | 1200 |
| Outer race | KA-series | 4800 | 1200 |

According to the existing samples in the CWRU experimental dataset, we devoted 1980 subsequences for the training process and each subsequence is comprised of 1024 vibration samples. To test the proposed structure, 750 different subsequences with 2048 samples have been considered. The number of data for testing and training for each class is given in Table 2.

In order to show the generality of the proposed GCNN network in fault detection and classification, the size of each member of training and testing data in the PU data centre is selected the same as in the CWR dataset. As mentioned before, 15 different files are selected from the PU data centre, as shown in Table 3. The data include healthy, inner race fault, and outer race fault conditions. We extracted 6000 different data from each file, K0, KI, and KA series, while 4800 different subsequences with 1024 samples for the training process and 1200 subsequences with 2048 samples have been considered for the testing process.

4 | THE DESIGNED GCNN BASED FAULT CLASSIFICATION AND DETECTION STRUCTURE

The proposed GCNN structure in this paper consists of three major blocks, that is, Gabor filter based convolutional layers, pooling layers, and dense layers, which are explained in the following subsections.

4.1 | Gabor filter based convolutional layer

In this set of layers, the deep features are significantly enhanced through scale and directions associated with Gabor filters. A feature map is obtained from a Gabor based convolutional layers, as follows:

\[ y_{\text{out}}^f = G\text{Conv} \left( X_{\text{in}}^f, G_i \right). \] (7)

The input and output feature maps are shown by \( X_{\text{in}}^f \) and \( y_{\text{out}}^f \), respectively, while \( G_i \) is Gabor based convolutional layer. The output of \( f^{th} \) Gabor based convolutional layers is:

\[ y_{\text{out}, i}^f = f^{\text{act}} \left( \sum_{j=1}^{\omega p_i} F_j \otimes G_{i,j} \right). \] (8)

The convolution operator is shown by \( \otimes \) symbol and \( j \) is the number of Gabor based convolutional layers. Additionally, \( f^{\text{act}}(\cdot) \) is the activation function. To prevent vanishing gradient and saturation in the training process and in order to enhance the computational efficiency, a rectified linear unit (ReLU) is selected as the activation function [18].

4.2 | Pooling layers

To pool the features extracted by the Gabor filter based convolutional layers, the pooling layer as a nonlinear subsampling layer, max-pooling is utilised in this paper. Max pooling layer is beneficial to prevent overfitting and reduce feature redundancy and size through overlapping convolution windows [5]. The output feature map of Gabor based convolutional layers is considered as the input of the pooling layer and is determined as follows [19]:

\[ y_{\text{out}, i}^p = f^{\text{act}} \left( \omega_{i,j}^{p} f^{\text{max-min}} \left( x_{i}^{p} \right) + b_{i}^{p} \right), \] (9)

where \( y_{\text{out}, i}^p, \omega_{i,j}^{p}, f^{\text{max-min}}(\cdot), x_{i}^{p}, \) and \( b_{i}^{p} \) show the output feature map, weight matrices, max-pooling function, input feature map, and max-pooling layer corresponding feature maps of the max-pooling layers. The activation function in this layer is the ReLU, similar to the Gabor based convolutional layers.

4.3 | Dense layers

Dense layers generate the final output of the bearing fault detection/classification in the induction motor and allow the designed GCNN network to control the dimension of the GCNN network. However, the input of the dense layers should be 1D-signals. Therefore, the 2D output of the Gabor based convolutional layer or max-pooling layers is flattened by the flattening layer. Two different dense layers are considered in the proposed GCNN based fault identification network. In the first one, the ReLU activation function is used and in the second one, which generates the final output, the sigmoid activation function is applied.
The dropout technique is used to reduce information redundancy and thereby prevent overfitting and reduce the computational burden.

4.4 Training process

In contrast with standard CNN, in GCNN networks only the learning weights $f_{out}$ are needed to be updated during the training process [15]. To this end, the gradient of the learning weights needs to sum up and:

$$\theta = \frac{\partial f_{loss}}{\partial f_{out}^i} = \sum_{d=1}^{D} \frac{\partial f_{loss}}{\partial f_j} \odot G(d, s),$$

(10)

$$f_{out} = f_{out}^i = \zeta \theta,$$

(11)

where $\theta$ and $\zeta$ show the learning weights and scale parameters, respectively. Furthermore, $f_{loss}$ is the loss function that binary cross-entropy and multi-nominal cross-entropy are selected as the loss function for bearing fault and classification, respectively. Also, the adaptive moment estimation method (Adam), which is an iterative-based stochastic gradient algorithm [18], has been considered to optimise the loss function. The training process is given in Algorithm 1.

ALGORITHM 1 Deep Gabor convolutional neural network for bearing fault detection/classification

1: Input:
2: The experimental vibration dataset $x$ from the CWRU and PU dataset, as (5).
3: Output:
4: Learning weights $\theta$
5: Initialisation:
6: Learning weights and corresponding hyper-parameter, the value of directions and scales of the Gabor filter
7: Iterative Process:
8: while epoch < $M$ ($M = 500$)
9: Generating Gabor filter based on convolutional layers based on (4) and (8).
10: Minimisation the loss function (binary/ multi-nominal cross entropy) and back propagation based on (10) and Adam algorithm.
11: Update learning weights based on (11)
12: Check the epoch number
   If epoch $\leq M$, go to the 9, otherwise end the training process
13: End

4.5 Overall structure of the designed GCNN in bearing fault identification

The designed GCNN network is visualised in Figure 5. GCNN is implemented for the bearing fault detection/classification as the following procedure:

1. The experimental data is firstly normalised, then converted from 1D vibration raw data into 2D signals and formed as a set of tensors with size ($s, 1, 1, 2, 1$).

2. As shown in Figure 5, the raw vibration signal of the industrial motor is the input for the Gabor filter based convolutional layer. This layer converts the input dataset into the feature vector with size ($s, 1, 1, 2, 512$).

3. Max pooling pools the maximum over each time interval as its output feature map. Thus, the length of max pooling outputs is smaller than the convolution layer, the feature vectors are formed as ($s, 1, 2, 512$) with a 10% dropout probability.

4. The output of the max-pooling should be converted from 2D signals to 1D signals. To this end, the output of max-pooling layers flattened by a flatten layer.

5. Two different dense layers have been considered, with the outputs that are formed as the set of signals with ($s, 128$) and ($s, 64$) sizes, respectively. In the second dense layer, the outputs are dropped out with a 25% probability.

6. In the final step, the fault detection (fault/health) conditions and fault classification (health/ball fault/inner race fault/ outer race fault in the CWRU dataset or health inner race fault/ outer race fault) takes place.

5 EXPERIMENTAL RESULTS AND DISCUSSION

To assess the proposed GCNN based fault identification, this section discusses the results obtained on the CWRU and the PU test system.

To address the performance of the proposed GCNN structure properly, we added noise to the existing data in the CWRU and PU datasets.

The original measurement vibration signal and noise signal are shown by $v_{measurement}$ and $v_{noise}$, respectively, and $SNR = 5$. Figure 6 depicts how the noisy signal produced for the test and the training process. The first part of Figure 6 shows a raw measured vibration signal, while the second and third parts show the noise and noisy signal, respectively. In other words, a sequence, as an example, consists of the original, noise, and the noisy signal from the PU dataset is illustrated in Figure 6.

The results are evaluated in terms of the confusion matrix, which consists of four main members including true positive (TP), true negative (TN), false positive (FP), and false negative
According to the confusion matrix, three typical indices, that is, the accuracy (Ac), specificity (Sp), and positive predictivity (Pp) have been considered. These indices are defined as:

\[
Ac = \frac{TP + TN}{TP + FP + TN + FP},
\]

\[
Sp = \frac{TN}{FP + TN},
\]

\[
Pp = \frac{TP}{FP + TP}.
\]

The Ac shows the level of accuracy of the designed GCNN and other fault detection methods, while specificity and positive predictivity show the rate of correct/not correct class.

The simulations are carried out in Python in a system with a GeForce GTX 1080 Ti GPU and 32 GB memory.

To verify the performance of the designed GCNN based fault classification method with the six different shallow and deep networks, as:

1. Feedforward neural network (FFNN) is composed of one hidden layer, one input layer and one output layer, where each consists of 1025, 500, and 10 neurons, respectively.
2. Learning vector quantisation (LVQ) is composed of one hidden layer, one input layer and one output layer, where each consists of 800, 60, and 10 neurons, respectively, and is based on Euclidean distance.
3. Support vector machine (SVM) with radial basis function as the main kernel function
4. The standard CNN with 1400 epoch and sigmoid activation function
5. The accelerated CNN (ACNN) with 1400 epoch and ReLU activation function, presented in [20].
6. Adaptive CNN with 1000 epoch and ReLU activation function, presented in [21].

In the following subsections, the results are evaluated in two different cases, (i) fault and healthy condition detection, (ii) fault classification as a four-class classification problem.

### Case 1: Bearing fault diagnosis

In this case, bearing fault is detected from the health conditions in the two different datasets including the CWRU and PU.

The results of the fault detection in the CWRU dataset are given in Table 4 and compared with three deep based and three shallow based fault detection methods under noisy conditions. The superiority of the proposed GCNN structure in the bearing fault of induction motor diagnosis is clearly obvious based on Table 4. Furthermore, the inferiority of the shallow based method in comparison with deep convolutional networks in the bearing fault detection is noticeable.

Table 5, similar to Table 4, compares the results obtained by the different methods in the fault diagnosis problem based on the PU dataset. All the indices are higher than 98.69%, and these values show that the proposed method is robust in noisy conditions. In contrast, the accuracy of adaptive CNN, ACNN, and standard CNN in the fault detection are less than 90.86% when signals follow the high level of the noise. Moreover, shallow-

### Table 4 Comparison between the proposed GCNN and other methods in fault diagnosis based on the CWRU dataset

| Methods      | Acc % | Spe % | Ppr % |
|--------------|-------|-------|-------|
| GCNN         | 99.73 | 99.70 | 100   |
| Adaptive CNN | 95.87 | 96.30 | 99.09 |
| ACNN         | 95.05 | 95.85 | 99.08 |
| CNN          | 93.47 | 95.11 | 97.57 |
| SVM          | 88.27 | 90.07 | 96.66 |
| LVQ          | 87.07 | 89.04 | 96.31 |
| FFNN         | 84.81 | 87.56 | 94.71 |

### Table 5 Comparison between the proposed GCNN and other methods in fault diagnosis based on the PU dataset

| Methods      | Acc % | Spe % | Ppr % |
|--------------|-------|-------|-------|
| GCNN         | 98.69 | 99.08 | 98.96 |
| Adaptive CNN | 90.86 | 93.63 | 92.47 |
| ACNN         | 90.64 | 93.46 | 91.57 |
| CNN          | 89.36 | 92.54 | 91.59 |
| SVM          | 82.64 | 85.88 | 87.81 |
| LVQ          | 81.97 | 85.25 | 85.85 |
| FFNN         | 78.92 | 81.87 | 85.85 |
TABLE 6  Comparison between the proposed GCNN and other methods in bearing fault classification in the CWRU dataset

| Methods       | Health Acc (%) | Health Sp (%) | Health Pp (%) | Ball Fault Acc (%) | Ball Fault Sp (%) | Ball Fault Pp (%) | Inner Race Acc (%) | Inner Race Sp (%) | Inner Race Pp (%) | Outer Race Acc (%) | Outer Race Sp (%) | Outer Race Pp (%) |
|---------------|----------------|---------------|---------------|--------------------|------------------|------------------|--------------------|-------------------|------------------|--------------------|------------------|------------------|
| GCNN          | 99.60          | 98.67         | 97.37         | 98.7               | 96.89            | 97.32            | 98.27              | 96.89            | 97.32            | 98.27              | 96.89            | 97.32            |
| Adaptive CNN  | 95.32          | 89.33         | 71.28         | 95.73              | 93.33            | 92.89            | 96.40              | 93.78            | 94.20            | 95.60              | 92.89            | 92.48            |
| ACNN          | 95.07          | 88.00         | 70.21         | 95.73              | 92.89            | 92.89            | 96.27              | 94.22            | 93.39            | 95.73              | 93.33            | 92.51            |
| CNN           | 93.07          | 74.67         | 62.92         | 93.07              | 88.89            | 88.11            | 93.33              | 89.33            | 88.55            | 92.80              | 88.44            | 87.67            |
| SVM           | 87.87          | 68.00         | 43.22         | 87.89              | 81.83            | 78.54            | 88.09              | 80.79            | 79.74            | 89.10              | 82.38            | 81.66            |
| LVQ           | 86.67          | 65.33         | 39.84         | 86.71              | 78.22            | 77.19            | 86.80              | 78.67            | 77.63            | 87.33              | 80.00            | 78.26            |
| FFNN          | 84.40          | 56.00         | 33.33         | 83.73              | 75.11            | 71.91            | 84.00              | 75.56            | 72.34            | 86.40              | 78.22            | 76.86            |

TABLE 7  Comparison between the proposed GCNN and other methods in bearing fault classification in the PU dataset

| Methods       | Health Acc (%) | Health Sp (%) | Health Pp (%) | Inner Race Acc (%) | Inner Race Sp (%) | Inner Race Pp (%) | Outer Race Acc (%) | Outer Race Sp (%) | Outer Race Pp (%) |
|---------------|----------------|---------------|---------------|--------------------|-------------------|-------------------|--------------------|--------------------|------------------|
| GCNN          | 98.69          | 97.92         | 98.16         | 98.27              | 96.89            | 97.32            | 98.27              | 96.89            | 97.32            |
| Adaptive CNN  | 90.86          | 85.33         | 87.00         | 90.82              | 85.22            | 87.02            | 90.36              | 84.67            | 87.02            |
| ACNN          | 90.64          | 85.00         | 86.66         | 90.52              | 84.83            | 86.55            | 89.96              | 84.00            | 86.55            |
| CNN           | 89.36          | 83.00         | 84.77         | 89.50              | 83.25            | 84.95            | 88.94              | 82.42            | 84.95            |
| SVM           | 82.64          | 76.17         | 72.94         | 82.50              | 75.92            | 72.76            | 81.39              | 74.25            | 72.76            |
| LVQ           | 81.97          | 75.42         | 71.88         | 82.17              | 75.67            | 72.18            | 81.06              | 74.00            | 72.18            |
| FFNN          | 78.92          | 73.00         | 66.82         | 78.97              | 72.67            | 67.23            | 77.85              | 71.00            | 67.23            |

Based bearing fault detection performs poorly with far less accuracy and reliability compared with the GCNN.

5.2 | Case 1: Bearing fault classification

In the second case, the fault classification is studied based on the CWRU and PU experimental datasets.

Based on the CWRU dataset, four different classes including health, ball fault, inner race fault, and outer race fault have been considered for the bearing fault classification in the induction motors. The results are compared in Table 6 in terms of accuracy and reliability. It is clear that the proposed GCNN classification method has far more accuracy and reliability in all different four conditions based on the indices. The difference is more obvious in health condition due to the low number of testing data size in comparison with the other three classes. However, the higher accuracy, specificity, and positive predictivity of the proposed deep network are undeniable. For instance, the proposed GCNN bearing fault classifier has improved accuracy of Adaptive CNN, ACNN, and CNN as deep convolutional networks approximately 4.49%, 4.76%, and 7.01% in health condition detection, while the obtained Sp and Pp show 12.12% and 38.68% improvement, respectively, in comparison with ACNN, 32.14% and 56.81% in comparison with standard CNN and also show 10.45% and 36.60% compared to adaptive CNN method. The superiority of the proposed deep classifier in bearing fault identification problem is far more obvious compared to the shallow based networks including SVM, LVQ, and FFNN. The better reliability and accuracy in all classes can be easily verified based on the obtained results of the shallow based networks and the proposed GCNN. In inner race fault detection, for example, the GCNN network improved the performance of the SVM, LVQ, and FFNN about 11.55%, 13.21%, and 16.98% in terms of accuracy, while in comparison based on the Sp and Pp, the proposed GCNN at least shows more than 20% improvement.

Table 7 compares the results of the GCNN and three deep neural networks and three shallow neural networks in terms of accuracy and reliability in high noisy conditions based on the experimental data in the PU dataset. It is clear that the proposed GCNN network is far more accurate and reliable than other methods, including deep and shallow structures. As can be seen in Table 7, the accuracy of the adaptive CNN, ACNN, and standard CNN methods are less than 90.86%, while the proposed method performs with an accuracy higher than 98.27%. In terms of Sp and Pp indices, the lowest values of the proposed method are 96.86% and 97.32%, respectively, whereas Sp and Pp values of other deep networks are less than 85.33% and 87.00%, respectively. Even the Pp values in health class identification are less than 71%, which shows that they are not reliable in the high noisy conditions. On the other hand, the superiority of the proposed GCNN based bearing fault classification in the industrial motors is far more obvious compared to the shallow based methods, including SVM, LVQ, and FFNN. The results of the shallow based structures show that values of Ac, Sp, and Pp are less than 90% in these methods.
5.3 Computational burden

Computational time plays a pivotal role in the early bearing fault diagnosis/classification. To assess the computation time properly, the proposed GCNN and other methods repeated for 50 different times and the average time in the testing process for the GCNN is 4.86 ms, while the adaptive CNN and standard CNN requires 7.18 ms and 14.52 ms, respectively. However, ACNN performs about 0.74 ms faster than the proposed method, which can be ignored due to a more accurate performance of the proposed method, in particular, based on the obtained results on the PU dataset. On the other hand, the shallow-based methods perform faster than the proposed method, where the average computational time of the SVM, LVQ, and FFNN are 2.68 ms, 2.78 ms, and 3.94 ms, respectively. However, the proposed method performs significantly more accurately, and its computational time is proper for real-time applications.

6 CONCLUSION

A new end-to-end deep-based structure is proposed in this paper for bearing fault identification as the most vulnerable element in the industrial motors, the most widely used rotary machine in the transportation and industries. The proposed method is designed based on the modulated Gabor filter and deep convolutional neural network to extract the features from the raw vibration signal and enhance the robustness in high noisy conditions. Furthermore, the training process is also modified to improve the learning ability as well as reducing computational time. To verify the generality and robustness of the proposed GCNN network, the results of two reliable experimental datasets, including the CWRU and PU datasets are discussed in terms of accuracy and reliability. The proposed structure benefits from the modulated Gabor filter based CNN to enhance the performance in terms of the computational time, accuracy, and reliability. The proposed GCNN based shows more robustness against noise and promotes the level of accuracy with a performance three times faster than standard CNN. It has also improved accuracy about 10% compared to deep networks, that is, adaptive CNN, ACNN, and CNN and about 20% in comparison with SVM, LVQ, and FFNN. The numerical results also verify the higher reliability, in particular in classification, with at least 15% improvement in terms of specificity and positive predictivity.

ORCID
Mohammad Mohammadi https://orcid.org/0000-0002-8953-1996

REFERENCES
1. He, M., He, D.: Simultaneous bearing fault diagnosis and severity detection using a LAMSTAR network-based approach. IET Sci. Meas. Technol. 12(7), 893–901 (2018)
2. Izadi, V., et al.: Supervisory algorithm based on reaction wheel modelling and spectrum analysis for detection and classification of electromechanical faults. IET Sci. Meas. Technol. 11(8), 1085–1093 (2017)
3. Zhang, S. et al.: Deep learning algorithms for bearing fault diagnostics—A comprehensive review. IEEE Access 8, 29857–29881 (2020)
4. Xin, Y. et al.: Intelligent fault diagnosis method for rotating machinery based on vibration signal analysis and hybrid multi-object deep CNN. IET Sci. Meas. Technol. 14(4), 407–415 (2020)
5. Afrasiabi, S. et al.: Integration of accelerated deep neural network into power transformer differential protection. IEEE Trans. Ind. Inf. 16(2), 865–876 (2020)
6. Dubey, R., Agrawal, D.: Bearing fault classification using ANN-based Hilbert footprint analysis. IET Sci. Meas. Technol. 9(8), 1016–1022 (2015)
7. Li, Y. et al.: A new rolling bearing fault diagnosis method based on multi-scale permutation entropy and improved support vector machine based binary tree. Measurement 77, 80–94 (2016)
8. Kankar, P.K., Sharma, S.C., Harsha, S.P.: Rolling element bearing fault diagnosis using wavelet transform. Neurocomputing 74(10), 1638–1645 (2011)
9. Jafari, H., Poshtan, J.: Fault detection and isolation based on fuzzy-integral fusion approach. IET Sci. Meas. Technol. 13(2), 296–302 (2019)
10. Afrasiabi, M. et al.: Power transformers internal fault diagnosis based on deep convolutional neural networks. J. Intell. Fuzzy Syst. 37(1), 1165–1179 (2019)
11. Afrasiabi, M. et al.: Deep-based conditional probability density function forecasting of residential loads. IEEE Trans. Smart Grid 11(4), 3646–3657 (2020)
12. Lu, W. et al.: Early fault detection approach with deep architectures. IEEE Trans. Instrum. Meas. 67(7), 1679–1689 (2018)
13. Zhao, R. et al.: Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Trans. Ind. Electron. 65(2), 1539–1548 (2018)
14. Afrasiabi, M. et al.: Deep learning architecture for direct probability density prediction of small-scale solar generation. IET Gener. Transm. Distrib. 14(11), 2017–2025 (2020)
15. Luan, S. et al.: Gabor convolutional networks. IEEE Trans. Image Process. 27(9), 4357–4366 (2018)
16. Loparo, K.: Case Western Reserve University Bearing Data Center (2012)
17. Lessmeier, C. et al.: Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In: Proceedings of the European Conference of the Prognostics and Health Management Society, pp. 05–08. The Prognostics and Health Management Society, Rochester (2016)
18. Afrasiabi, S. et al.: Designing a composite deep learning based differential protection scheme of power transformers. Appl. Soft Comput. 87, 105975 (2020)
19. Afrasiabi, M. et al.: Multi-agent microgrid energy management based on deep learning forecaster. Energy 186, 115873 (2020)
20. Afrasiabi, S. et al.: Real-time bearing fault diagnosis of induction motors with accelerated deep learning approach. In: 2019 10th International Power Electronics, Drive Systems and Technologies Conference (PEDSTC), pp. 155–159. IEEE, Shiraz (2019)
21. Ince, F. et al.: Real-time motor fault detection by 1-D convolutional neural networks. IEEE Trans. Ind. Electron. 65(11), 7067–7075 (2016)

How to cite this article: Afrasiabi S, Mohammadi M, Afrasiabi M, Parang B. Modulated Gabor filter based deep convolutional network for electrical motor bearing fault classification and diagnosis. IET Sci Meas Technol. 2021;15:154–162. https://doi.org/10.1049/smt2.12017