Available online at www.sciencedirect.com

ScienceDirect

Energy Reports 00 (2017) 000–000

2022 The 3rd International Conference on Power Engineering (ICPE 2022), December 09-11, 2022, Sanya, Hainan, China

A Novel Framework Based on Adaptive Multi-Task Learning for Bearing Fault Diagnosis

Jierui Zhang[a], Jianjun Chen[a], Huiwen Deng[a], Weihao Hu[a]

[a]*School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, No.2006, Xiyuan Avenue, West Hi-tech Zone, Chengdu 611731, China*

Abstract

Bearing fault diagnosis is very important for security and efficiency of electric machine. In recent years, the newly emerging deep learning methods has risen bearing fault diagnosis as a research hotspot again. To achieve better performance and interpretability and absorb novel methods, this paper proposes a novel framework based on adaptive Multi-Task Learning for bearing fault diagnosis, GM-CNN, including data augmentation, image encoding methods and adaptive Multi-Task Learning (MTL). Firstly, data augmentation (DA) methods are applied in data preprocessing for tackling the problems of lack of data and weak generalization ability. They include cropping, flipping and noise injection. Secondly, Gramian Angular Fields (GAF), Markov Transition Fields (MTF) and their combination GAF-MTF are used to encode time series into images which are more proper for convolutional neural network (CNN) to extract features and classify. Then, the processed data are fed into the propose MTL framework, and tasks for classifying the fault type and severity are trained jointly as they share some general knowledge and this saves more time. Besides, attention is applied to make the MTL adaptive, which is helpful for more balanced training. Some experiments are carried out. The experiment results show that the proposed framework is relatively simple but more effective, classifying the fault type and severity with high accuracy (basically higher than 99%). With DA, accuracy is increased by 15.06% and 12.05% for 2 tasks on average. The proposed method has higher accuracy of 1.27% and 2.30% than the other best methods. Adaptive MTL has higher accuracy of 4.36% and 3.16% than the other best methods, and highly improves efficiency. It is shown that every step of the framework is important and essential. This paper provides a reference for future study on bearing fault diagnosis in the perspective of feature extraction and CNN. It could also be applied to other time series classification situations which could be promising directions.

Peer-review under responsibility of the scientific committee of the Name of the Conference, Conference Organizer Name, Year or Edition of Conference.

Keywords: bearing fault diagnosis, Multi-Task Learning (MTL), Convolutional Neural Network (CNN), Gramian Angular Fields (GAF), Markov Transition Fields (MTF), Data Augmentation (DA)

© YEAR The Authors. Published by Elsevier Ltd.

Peer-review under responsibility of the scientific committee of the Name of the Conference, Conference Organizer Name, Year or Edition of Conference.

1. Introduction

Bearing is an important component, widely used in many industries like aircraft, mining and mechanism. There is not a fixed lifetime of individual bearing. And bearing fault is one of the main sources of mechanical fault which leads to hidden danger, efficiency reduction and economic loss in industrial production. That is why the diagnosis and maintenance of bearing are very important. There are methods like listening the sound and using lubricant to detect the fault but they are highly dependent on experience. In practice, the vibration signals of bearing are used most widely to diagnose the fault by researchers, which is also the topic of this paper. [1], [2]

Machine learning is most direct way to be applied in vibration signals and has been studied widely. Some traditional machine learning algorithms like Support Vector Machine (SVM) and (Kernel) Principal Component Analysis ((K)PCA) have shown good results [3], [4]. But there seems to be an accuracy bottleneck which traditional methods can hardly break through and the potential of data has not been exploited fully. Benefiting from increasing volume of data and greater computing power, deep learning has shown stronger power in many areas, especially computer vision (CV) and natural language processing (NLP) [5]. Many researchers have applied deep learning to bearing fault diagnosis, and tend to have better results than traditional ones. They include the application of Long Short-Term Memory (LSTM), 1-D CNN classifier, Capsule Neural Network, etc. [6]-[11]

Basically, there are 3 directions using deep learning in the existing work. First and most naively, an end-to-end deep learning method is proposed as in [6], [10], which is somehow useful by applying some tricks. But this method shows a drawback of deep learning, lack of interpretability. Besides, the model’s parameters could vary a lot when the ways of data slicing differ, which leads to problems in generalization. Secondly, use signal processing techniques like short-time Fourier transform (STFT) to generate 2-D images, and apply CNN or its variants on them [8]. It is considered to be better because general features are extracted which are not influenced by the ways of data slicing. However, it’s not very explainable to apply CNN in such images, as the x-axis and y-axis are in different physical dimensions, thus still lacking interpretability. Thirdly, transfer learning or few-shot learning is introduced to handle the problems of lack of samples, different distribution of training and test data, etc. [11]-[13] Though related work performs well when facing few or different-domain data, the aforementioned problem is not tackled as well.

It is noted that, in many real situations, complex problems can’t be simply divided into independent sub-problems, as this omits some shared information and representation [14], [15]. For bearing fault diagnosis, this also holds. In this topic, two important aims are the accurate classifications of fault type and severity. Some researchers either classify them separately or combine them to new classes of pair of type and severity, and tries to solve the classification task [16], [17]. By contrast, Multi-Task Learning is used to make the most of the shared knowledge between fault type and fault severity which tends to be time-saving and powerful due to the share network [18]. The latter one is considered to be the best strategy and which is applied in our framework. Besides, the training progresses of the two tasks may be different, so an adaptive method for Multi Task Learning is preferred, as [19].

In a conclusion, there are some good results in existing work regarding different aspects, but they either lacks novelty which leads to relatively poor performance, or have achieved good results however without interpreting the reason and mechanism. And few work has totally absorbed the advantages from other work. Thus, a novel framework for bearing fault diagnosis is needed, which is the main work of this paper.

The main novelty and contributions are as follows:

- A novel framework for bearing fault diagnosis is proposed, which has better interpretability and generalization ability, absorbing and improving the ideas of previous work.
- Data augmentation is applied to solve the problems of data lack and weak generalization ability, which substitutes transfer learning and, still, performs well.
- A novel method to encode time series into images, GAF-MTF, which generates better 2-D image for robust CNN classification and has better interpretability, is applied.
- Adaptive MTL with an elaborately designed function is applied, which proves to be more efficient and powerful.

The rest of this paper is organized as follows. In section 2, a brief introduction to the key techniques, including CNN, data augmentation, methods to encode time series into images and (adaptive) Multi-Task Learning, is brought out. In section 3, the data description of CRWU and the novel framework of **G**AF**M**TF-**CNN** is proposed. In section 4, experiments regarding the different parameters and settings are carried out and the results are analyzed to evaluate the framework. In section 5, the conclusion is drawn, some potential directions are proposed as well.

2. Preliminaries

2.1. Convolutional neural network

Convolutional neural network (CNN) is a new kind of neural network which has unique layers called convolutional layer and pooling layer for convolution and pooling [20].

1) Convolutional Layer

It has many convolutional kernels to extract features with translational invariance. The properties of reception field and shared weights greatly reduce the trained parameters and avoid overfitting to some extent.

Denote the original feature map (or input image) size as *ω*, kernel size as *k*, stride as *s* and padding as *p*, then the new image size *ω'* can be calculated by the following formula.

$$\omega' = \frac{\omega + 2p - k}{s} + 1 \tag{1}$$

For input X_i and kernel K_j, the output $Y_{ij} = f(b_j + \sum K_j * X_i)$, where * stands for convolution, *K* and *b* stand for the value of the kernel and bias respectively. *f* is the activation function which could be ReLu, Sigmoid, etc. [17], [20] In our paper Sigmoid is used for better fitting ability.

2) Pooling Layer

Generally, each convolutional layer is followed by a pooling layer. A pooling layer reduces the size of the picture for extracting higher level feature more easily, while preserving most space information. Pooling can be done with average or max. In our experiment, the max pooling is chosen. A pooling layer is fixed manually rather than learned.

For input feature map X_i, the output one Y_i is generated using formula (2). *r* is the size of pooling filter. [17], [20]

$$Y_i = \max_{r\times r}(X_i) \tag{2}$$

2.2. Data Augmentation

Many time series applications suffer from lack of data [21], including bearing fault diagnosis. To reduce the complexity of the framework and time, basic approaches such as cropping (slicing), flipping and noise injection are adopted in our paper [22]. Actually, to regularize the size of input data, cropping is used first and then followed by flipping or noise injection. Fig. 1 shows how they work.

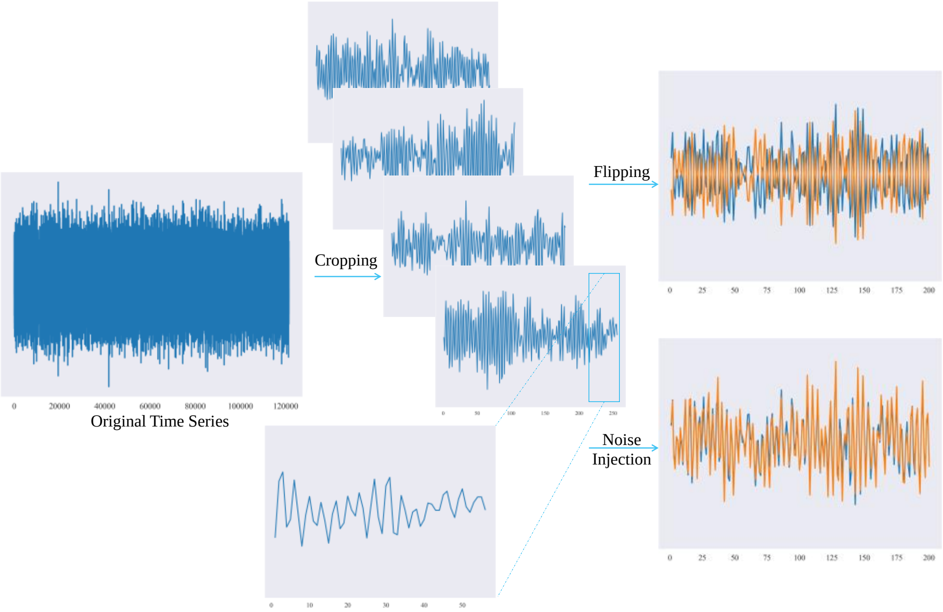

Fig. 1. Methods for data augmentation.

1) Cropping

Cropping or slicing means cutting a sub continuous data slice from the original time series data, which has the identical label as the original data [23]. We cut the data with some patterns rather than randomly, i.e., cutting a slice with length 256 with interval 100 from the start of the series. The length is fixed to make sure the input data of the neural network have the same size. And the specific length 256 is chosen as a small length leads to insufficient information and low accuracy while a large length leads to poor timeliness and high computation burden, and 256 proves to be the optimal trade-off in square numbers with the form of 2^{2n}. Overlapping is allowed in cropping. For model using, just crop a 256-length sample from the data to be tested, and use the trained model to predict its class.

2) Flipping

Let $\bar{X}$ denote the average of time series X=x_1,x_2,…,x_n, then generate a new time series X' where $x_t' = 2\bar{X} - x_t$, and the label of new generated series is the same as the original one.

3) Noise injection

This method means injecting some noise with small magnitude into the time series while the label holds. The noises include Gaussian noise, spike, slope-like trend, etc [24]. In this paper, Gaussian noise is adopted because it’s more general in nature and some work has proved that the model trained using data with added Gaussian noise has better generalization ability [25]. Let S denote the standard variance of time series X. The formulas are as follows.

$$S^2 = \frac{\sum_i (x_i - \bar{X})^2}{n-1} \tag{3}$$

$$x_t' = x_t + \varepsilon, \varepsilon \in N(0, S^2/10) \tag{4}$$

2.3. Encoding Time Series as Images

Though there are many methods for processing time series immediately, it is believed by many researchers that encoding it to images and applying computer vision methods has much potential [26].

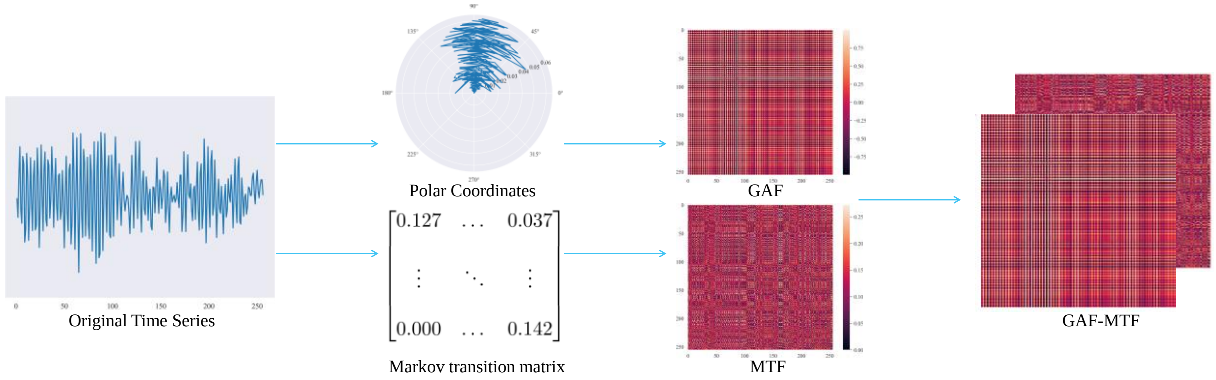

Fig. 2. GAF, MTF and GAF-MTF.

Fig.2. shows the methods to encode time series as images, GAF, MTF, and their combination GAF-MTF.

1) Gramian Angular Field

GAF is the short for Gramian Angular Field, where time series is represented in polar coordinate. For a time series X containing n items, first normalize it such that all the values fall in interval [-1,1] which is the domain of *arccos*(x) and then transform the time series into polar coordinate form:

$$\tilde{x}_i = \frac{((x_i - \max(x_i)) + (x_i - \min(x_i)))}{\max(x_i) - \min(x_i)} \tag{5}$$

$$\begin{cases} \phi = \arccos(\tilde{x}_i), -1 \le \tilde{x}_i \le 1, \tilde{x}_i \in \tilde{X} \\ r = \dfrac{t_i}{N}, t_i \in \mathbb{N} \end{cases} \tag{6}$$

where N is a normalization factor to restrict the range of polar coordinate system. Since $cos(x)$ is monotonic in interval [0,π], the map of time series in Cartesian coordinate and polar coordinate is invertible. To measure the temporal correlation of different time stamps, the angular cosine of trigonometric sum is used, and GAF(G) is defined as follows. [26]

$$G = \begin{bmatrix} \cos(\phi_1+\phi_1) & \cdots & \cos(\phi_1+\phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n+\phi_1) & \cdots & \cos(\phi_n+\phi_n) \end{bmatrix} = \tilde{X}'\cdot\tilde{X} - \sqrt{I-\tilde{X}^2}'\sqrt{I-\tilde{X}^2} \tag{7}$$

where I is unit row vector [1,1,...,1]. The second equation holds as $\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)$.

2) Markov Transition Field

MTF is the short for Markov Transition Field. For the time series X, choose a proper Q representing the number of quantile bins and let each x fall into corresponding bin $q_j\,(j\in[1,Q])$. Iterate the time series, check the neighboring items and count the transitions between different bins. The total number of transitions between q_i and q_j is noted as w_ij, and normalize them by $\sum_j \omega_{ij}=1$. Then first-order Markov transition matrix W is constructed. To overcome too much information loss, some modifications are adopted and that's also how MTF(M) is constructed: if the data at time stamp a and b belong to quantile bins q_i and q_j respectively, then the value $m_\text{ab}=w_\text{ij}$, as follows. [26], [27]

$$M = \begin{bmatrix} w_{ij|x_1\in q_i,x_1\in q_j} & \cdots & w_{ij|x_1\in q_i,x_n\in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n\in q_i,x_1\in q_j} & \cdots & w_{ij|x_n\in q_i,x_n\in q_j} \end{bmatrix} \tag{8}$$

3) GAF-MTF

It can be concluded that G_ij denotes the directions at t_i and t_j while M_ij denotes the transition probability between the corresponding quantiles. Thus, GAF encodes the static information while MTF encodes dynamic one, and they are just like orthogonal channels. Besides, GAF and MTF has the same size, so they can be combined into new image denoted as GAF-MTF which has two channels [26]. For one thing, general features can be extracted by GAF, MTF and GAF-MTF regardless of the ways of data slicing. For another, the x-axis and y-axis of these images are in the same physical dimension. Thus, the interpretability is improved compared with previous methods [8], [10].

2.4. Multi-Task Learning

Human tend to learn multiple tasks all at once and improve faster. These tasks, like soccer and basketball, have something in common, and knowledge learned in one task could help learn another. Inspired by this, MTL is proposed to make the most of shared information and improve performance. MTL is defined as: Given m learning tasks $\{T_i\}_{i=1}^m$ where all or some of the tasks are related, MTL expects to learn the m tasks jointly to improve the learning performance for each task by using the shared knowledge in all or some of other tasks. [28]

According to this definition, generally in supervised learning, a task T_i has a corresponding training dataset D_i = $\{X_j^i, y_j^i\}_{j=1}^{n_i}$ with size n_i. Focusing on bearing fault diagnosis, the most important tasks are the classification of fault type and severity and the dataset is shared by the two tasks. MTL is applied by Xie et al. to classify the fault type and severity jointly and have some good result [18]. Here some modification is done to train the model adaptively which helps improve the balance of training two tasks. The losses of two tasks are added with weights using the formulas below, where acc_1 and acc_2 represent the accuracies of task type and task severity respectively. The ε is a very small positive value, making sure the two items are positive. In the first epoch, just let w_1= w_2=0.5.

$$\begin{cases} w_1 : w_2 = (1-acc_1+\varepsilon):(1-acc_2+\varepsilon) \\ w_1 + w_2 = 1 \end{cases} \tag{9}$$

$$loss = w_1 loss_1 + w_2 loss_2 \tag{10}$$

3. The Proposed GM-CNN Framework

In this part, the whole **G**AF**M**TF-**CNN** framework is proposed, as Fig.3. It includes the training and using.

For model training, data augmentation is applied first and data samples are generated by cropping, flipping and noise injection. The samples are fed into the MTL model based on CNN for training, whose parameters are shown in Table.1. The total loss is calculated by Eq. 10. while the weights are dynamically adjusted according to the two tasks' training accuracy by Eq. 9. Thus the training process by gradient descent is adaptive and balanced, which proves to be better in Section 4. After training, the expected model is obtained and is able to classify fault type or severity.

For model using, the sample to be classified should be processed, i.e., cropped first and be transformed into GAF-MTF then. Input the GAF-MTF into the obtained model and classification results are generated.

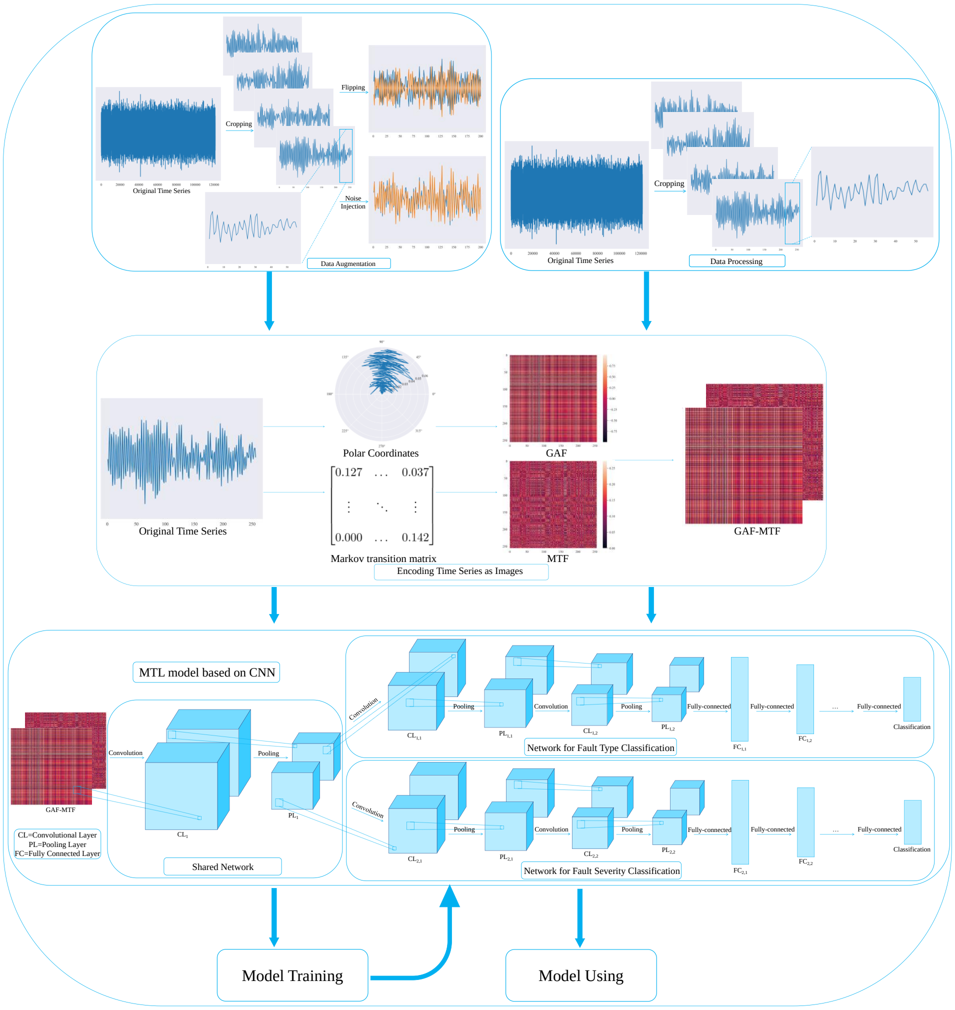

Fig. 3. The Proposed GM-CNN Framework.

Table 1. The parameters of the neural network.

Layer Name	Parameter Name	Parameter Value	Activation Function	Output Dimension	Layer Name	Parameter Name	Parameter Value	Activation Function	Output Dimension
Data	/	/	/	[256,256,2]	$FC_{1,3}$	/	/	Sigmoid	[3,1]
CL_1	Kernel_size	15×15×8	Sigmoid	[256,256,8]	$CL_{2,1}$	Kernel_size	5×5×16	Sigmoid	[128,128,16]
PL_1	Pooling_size	2×2	/	[128,128,8]	$PL_{2,1}$	Pooling_size	2×2	/	[64,64,16]
$CL_{1,1}$	Kernel_size	5×5×16	Sigmoid	[128,128,16]	$CL_{2,2}$	Kernel_size	3×3×32	Sigmoid	[64,64,32]
$PL_{1,1}$	Pooling_size	2×2	/	[64,64,16]	$PL_{2,2}$	Pooling_size	2×2	/	[32,32,32]

$CL_{1,2}$	Kernel_size	3×3×32	Sigmoid	[64,64,32]	$FC_{2,1}$	/	/	Sigmoid	[1000,1]
$PL_{1,2}$	Pooling_size	2×2	/	[32,32,32]	$FC_{2,2}$	/	/	Sigmoid	[100,1]
$FC_{1,1}$	/	/	Sigmoid	[1000,1]	$FC_{2,3}$	/	/	Sigmoid	[3,1]
$FC_{1,2}$	/	/	Sigmoid	[100,1]					

4. Experiments

4.1. Data Description

The dataset is from Case Western Reserve University (CWRU) bearing center. CWRU dataset is the most wildly used public dataset for rolling bearings. The fault type includes the inner raceway, rolling element (or ball), and out raceway, denoted as IR, REB and OR respectively. The bearings have fault severity ranging from 0.007 inches to 0.040 inches in diameter, divided into 3 classes. The bearings with different fault type and severity are reinstalled into the machine and the vibration signals are recorded by sensors with different motor loads (motor speeds of 1720 to 1797 RPM, and in this experiment 1797 RPM is chosen). The vibration signals, or time series, are then stored in MATLAB(*.mat*) format. There are total 161 records divided into 4 groups: 48k normal-baseline, 48k drive-end fault (48k DE), and 12k drive-end fault (12k DE), and 12k fan-end fault (12k FE). The experiments carried on the latter 3 groups. And the average length of the samples is 411861.71, 121306.51, 121895.96 respectively. [29], [30]

4.2. Case Study Ⅰ: Effect of Data Augmentation

Separately test datasets 48k DE, 12k DE and 12k FE. In each dataset, there are 3 different fault types as well as 3 different fault severities. Without data augmentation, there are 100 samples simply sliced in each class and total 900 samples. With DA, these samples are flipped and injected with Gaussian noise, and there are 2700 samples then. The datasets are shuffled and partitioned randomly with ratio 7:3 for training and testing, which is the same in following case studies. The parameters of the neural network are shown in table 1 and adaptive MTL is adopted.

The results are shown in table 2. It suggests that DA has strong power in improving generalization ability. For task 1 (fault type classification) DA improves the accuracy by an average of 15.06%, and 12.05% for task 2 (fault severity classification). There is dramatic effect in dataset 48k DE, with an increase of 32.84% and 19.63%.

Choose 12k FE and with DA as an example and show the curves of total loss and accuracy with epoch increasing in Fig. 4. It suggests that the loss falls and the accuracy increases to an expected level in first few rounds.

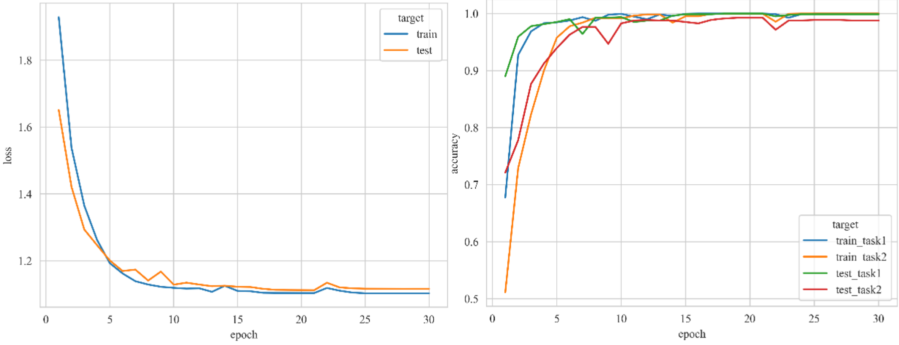

Fig. 4. (a) the curve of total loss; (b) the curve of accuracy.

Table 2. Comparison of accuracy (%) without and with data augmentation.

Data Source	without DA	with DA
48k DE	66.67, 77.78	**99.51**, **97.41**
12k DE	97.78, 92.96	**99.75**, **99.14**
12k FE	89.63, 89.63	**100.00**, **99.26**

4.3. Case Study Ⅱ: Different Methods to Encoding Time Series as Images

To show the power of GAF-MTF, other four methods, i.e., the simple encoding method, Hilbert method and separate GAF and MTF, are adopted for comparation. The simple method refers to transforming the 256-length sample into 16×16 image by filling the image from top to bottom with polyline, while Hilbert method improve the transforming method with Hilbert curve. [31] There are shown in Fig. 5. The structure of neural network for simple method and Hilbert method is almost the same as that for GAF, MTF or GAF-MTF, as Fig 3.

The results are shown in table 3. Two percentages in a cell represent the test accuracy of two tasks, type and severity. It shows that the methods GAF, MTF perform better than two traditional methods generally. In dataset 12k DE and 12k FE, GAF performs better than MTF, while in dataset 48k DE it's on the contrary. And GAF-MTF performs far better than separate GAF and MTF, as it contains more comprehensive information. GAF-MTF has higher accuracy of 1.27% and 2.30% than the other best methods.

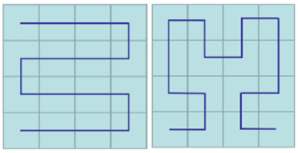

Fig. 5. (a) a simple curve; (b) the Hilbert curve.

Table 3. Comparison of accuracy (%) among different time series encoding methods.

Data Source	Simple	Hilbert	GAF	MTF	GAF-MTF
48k DE	93.81, 90.75	97.41, 87.12	90.14, 90.17	97.04, 93.59	**99.51**, **97.41**
12k DE	97.90, 63.13	94.45, 72.38	99.51, 98.15	98.25, 97.04	**99.75**, **99.14**
12k FE	89.52, 67.32	81.13, 63.38	98.52, 97.16	98.47, 92.97	**100.00**, **99.26**

4.4. Case Study Ⅲ: Effect of MTL

The GPU used is Tesla T4 and three methods are tested in the same condition. Results are shown in table 4, and the third value in each cell means the total training time, i.e., the time when the highest accuracy is reached for the first time. It suggests that in most situations single task learning has slightly higher accuracy than MTL, but MTL is more efficient. However, for MTL, when one task is almost finished, another task is not trained completely. Adaptive MTL appears to be more competitive in both accuracy and training time. On average, adaptive MTL has higher accuracy of 4.36% and 3.16% than the other best methods, and saves 22.34 minutes.

Table 4. Comparison of accuracy (%) and total training time (minute) with and without (adaptive) MTL.

Data Source	Single task	MTL	Adaptive MTL
48k DE	88.27, 89.77, 66.93	88.04, 87.79, 43.93	**99.51**, **97.41**, **30.05**
12k DE	**99.88**, 97.66, 66.07	99.14, 97.66, 43.93	99.75, **99.14**, **33.32**
12k FE	97.53, 98.89, 63.65	98.03, 98.40, 40.63	**100.00**, **99.26**, **21.13**

5. Conclusion

In this paper, a novel framework is put forward. The results of experiments show that this framework performs very well, and every step in the framework has its unique function. DA improves the generalization ability thus increasing accuracy; GAF-MTF has better feature than what other methods generate; adaptive MTL saves time and let two tasks be trained completely. And solving task 2 is more difficult than task 1, and 48k DE is more tough.

In the future, the mechanism of GAF and MTF is supposed to be explored further and more methods for classification of time series are supposed to be studied. Besides, there is little research about Multimodal Learning or Federated Learning for bearing fault diagnosis, which could be potential directions.

Acknowledgements

This work was supported by the National Science Foundation of China (Grant No. 52107074).

References

[1] Wang, Haisheng, Jian Wei, and Pengjin Li. “Research on Fault Diagnosis Technology Based on Deep Learning.” *Journal of Physics: Conference Series*. Vol. 2187. No. 1. IOP Publishing, 2022.

[2] Chen, W. “Application of Deep Learning in Rolling Bearing Fault Diagnosis.” *Southwest Jiaotong University* 621 (2018): 1-63.

[3] Shuang, Lu, and Li Meng. “Bearing fault diagnosis based on PCA and SVM.” *2007 International conference on mechatronics and automation*. IEEE, 2007.

[4] Liu, Zhiwen, et al. “A hybrid intelligent multi-fault detection method for rotating machinery based on RSGWPT, KPCA and Twin SVM.” *ISA transactions* 66 (2017): 249-261.

[5] Pouyanfar, Samira, et al. “A survey on deep learning: Algorithms, techniques, and applications.” *ACM Computing Surveys (CSUR)* 51.5 (2018): 1-36.

[6] Khorram, Amin, Mohammad Khalooei, and Mansoor Rezghi. “End-to-end CNN+ LSTM deep learning approach for bearing fault diagnosis.” *Applied Intelligence* 51.2 (2021): 736-751.

[7] Pan, Honghu, et al. “An improved bearing fault diagnosis method using one-dimensional CNN and LSTM.” *Strojniski Vestnik/Journal of Mechanical Engineering* 64 (2018).

[8] Zhu, Zhiyu, et al. “A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis.” *Neurocomputing* 323 (2019): 62-75.

[9] Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. “Dynamic routing between capsules.” *Advances in neural information processing systems* 30 (2017).

[10] Zhuang, Y., et al. “An end-to-end approach for bearing fault diagnosis based on LSTM.” *Noise Vib. Control* 13 (2019): 4-6.

[11] J. Chen, W. Hu, D. Cao, Z. Zhang, Z. Chen and F. Blaabjerg, “A Meta-Learning Method for Electric Machine Bearing Fault Diagnosis under Varying Working Conditions with Limited Data,” in *IEEE Transactions on Industrial Informatics,* doi: 10.1109/TII.2022.3165027.

[12] Zhiyi, He, et al. “Transfer fault diagnosis of bearing installed in different machines using enhanced deep auto-encoder.” *Measurement* 152 (2020): 107393.

[13] Wu, Zhenghong, et al. “An adaptive deep transfer learning method for bearing fault diagnosis.” *Measurement* 151 (2020): 107227.

[14] Zhao, Xiaoping, et al. “A multi-fault diagnosis method of gear-box running on edge equipment.” *Journal of Cloud Computing* 9.1 (2020): 1-14.

[15] Crawshaw, Michael. “Multi-task learning with deep neural networks: A survey.” *arXiv preprint arXiv:2009.09796* (2020).

[16] Huang, Ruyi, et al. “A robust weight-shared capsule network for intelligent machinery fault diagnosis.” *IEEE Transactions on Industrial Informatics* 16.10 (2020): 6466-6475.

[17] Chen, Zhuyun, et al. “A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks.” *Mechanical Systems and Signal Processing* 140 (2020): 106683.

[18] Xie, Zongliang, et al. “End to end multi-task learning with attention for multi-objective fault diagnosis under small sample.” *Journal of Manufacturing Systems* 62 (2022): 301-316.

[19] Wang, Huan, et al. “Feature-level attention-guided multitask CNN for fault diagnosis and working conditions identification of rolling bearing.” *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[20] LeCun, Yann, et al. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[21] Shijie, Jia, et al. “Research on data augmentation for image classification based on convolution neural networks.” *2017 Chinese automation congress (CAC)*. IEEE, 2017.

[22] Wen, Qingsong, et al. “Time series data augmentation for deep learning: A survey.” *arXiv preprint arXiv:2002.12478* (2020).

[23] Cui, Zhicheng, Wenlin Chen, and Yixin Chen. “Multi-scale convolutional neural networks for time series classification.” *arXiv preprint arXiv:1603.06995* (2016).

[24] Wen, Tailai, and Roy Keyes. “Time series anomaly detection using convolutional neural networks and transfer learning.” *arXiv preprint arXiv:1905.13628* (2019).

[25] Sietsma, Jocelyn, and Robert JF Dow. “Creating artificial neural networks that generalize.” *Neural networks* 4.1 (1991): 67-79.

[26] Wang, Zhiguang, and Tim Oates. “Encoding time series as images for visual inspection and classification using tiled convolutional neural networks.” *Workshops at the twenty-ninth AAAI conference on artificial intelligence*. 2015.

[27] Campanharo, Andriana SLO, et al. “Duality between time series and networks.” *PloS one* 6.8 (2011): e23378.

[28] Zhang, Yu, and Qiang Yang. “A survey on multi-task learning.” *IEEE Transactions on Knowledge and Data Engineering* (2021).

[29] Neupane, Dhiraj, and Jongwon Seok. “Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review.” *IEEE Access* 8 (2020): 93155-93178.

[30] Smith, Wade A., and Robert B. Randall. “Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study.” *Mechanical systems and signal processing* 64 (2015): 100-131.

[31] B. Moon, H. V. Jagadish, C. Faloutsos and J. H. Saltz, “Analysis of the clustering properties of the Hilbert space-filling curve,” in *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 124-141, Jan.-Feb. 2001, doi: 10.1109/69.908985.