This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

Convolutional Neural Network with Automatic Learning Rate Scheduler for Fault Classification

Long WEN, Liang GAO, *Senior Member, IEEE*, Xinyu LI and Bing ZENG

***Abstract*—Fault classification is vital in smart manufacturing, and Convolutional Neural Network (CNN) has been widely applied in fault classification. But the performance of CNN heavily depends on its learning rate. As the default setting on learning rate cannot guarantee its performance, the learning rate tuning process becomes essential. However, the traditional learning rate tuning methods either cost much time consumption or rely on the experts' experiences, so it is a considerable barrier for the users. To overcome this drawback, this paper proposes a CNN with automatic learning rate scheduler (AutoLR-CNN) for fault classification. Firstly, the Long Short-Term Memory (LSTM) is used to extract the features of the past loss of CNN. Then, an agent based on Deep Deterministic Policy Gradient (DDPG) is trained to automatically control the learning rate for CNN online. Thirdly, the double CNN structure is developed to enhance the stability of the proposed method. The proposed AutoLR-CNN is tested on two famous bearing datasets and a practical bearing dataset on wind turbine. The results of AutoLR-CNN are superior to six common used baseline learning rate schedulers in Tensorflow. AutoLR-CNN is also compared with other reported machine learning and deep learning methods. The results show that AutoLR-CNN has achieved the state-of-the-art performance in fault classification.**

***Index Terms*—Fault Classification, Learning Rate Scheduler, Deep Deterministic Policy Gradient, Convolutional Neural Network.**

I. Introduction

FAULT diagnosis is vital in modern manufacturing industries [1]. With the development of smart manufacturing, the huge amount of data can be collected. The massive data brings great opportunities and challenges in fault diagnosis field, making the data-driven fault diagnosis receiving increasing attentions from both engineering and academic fields [2]. The procedure of data-driven fault diagnosis includes data acquisition, feature extraction and fault classification [3]. Deep Learning (DL) methods have been widely applied on the fault classification to avoid the risk of catastrophic failures and to ensure the safety and reliability on machinery system [4].

Learning from the massive data to extract the efficient data features and establish the effective fault classification models is the key for DL based fault classification. As one of the most popular DL methods, Convolutional Neural Network (CNN) has the powerful feature extraction ability from raw data and it has gained increasing attentions on fault classification [5]. Lei et al. [6] reviewed the applications of DL on the intelligent fault classification and pointed out that DL has reformed intelligent fault classification by establishing the relationship between the increasingly-grown monitoring data and the health states of machines. Waziralilah et al. presented the review of CNN on fault classification [7], and it show that the CNN models have achieved remarkable results on fault classification. However, training a powerful CNN model for fault classification is hard, and the CNN training process highly depends on its hyper-parameters [8], especially the learning rate.

Learning rate is one of the most important hyper-parameters for CNN training process, and it greatly impacts the final performance of CNN models. Several theoretical and empirical evidences [9][10] have been investigated on the effect of the learning rate on training neural networks. Even though there are several previous studies giving the empirical suggestions, but tuning for finding the proper learning rate is still difficult for the CNN methods on real-world applications [11]. Especially on the CNN based fault classification, the datasets are totally different with the benchmark dataset on computer vision, making the learning rate tuning process becoming essential.

The main dominant approaches for learning rate tuning are the 'trial and error' or manual search [12]. Unusually, the tuning process should repeatedly train and evaluate CNN models attempting to find the optimal learning rate until obtaining the good choices. However, the turning process is a considerable barrier for the users in fault classification. Firstly, the default settings on learning rate for CNN methods cannot guarantee its performances [13], which makes the learning rate tuning processes time consumption and labor intensive. Secondly, the tuning process heavily depends on the experts' experience, and it is hard to decide which learning rate values are potential to obtain the good CNN model. Thirdly, the situation becomes

This work was supported in part by the National Key R&D Program of China (2019YFB1704600), the Natural Science Foundation of China under Grant 51805192 and 51775216, and sponsored by the State Key Laboratory of Digital Manufacturing Equipment and Technology (DMET) of Huazhong University of Science and Technology (HUST) under Grant No. DMETKF2020029. (Corresponding author: Xinyu Li)

Long WEN is with School of Mechanical Engineering and Electronic Information, China University of Geosciences, No. 388 Lumo Road, Wuhan, 430074, China (e-mail: wenlong@cug.edu.cn).

Liang GAO and Xinyu LI are with the State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science and Technology, 1037#, Luoyu Road, Wuhan, 430074, China (gaoliang@mail.hust.edu.cn, lixinyu@mail.hust.edu.cn)

Bing ZENG is with Big Data Center, XEMC Windpower Co. Ltd., No. 68 Ji'an road, Xiangtan, 411102, China (e-mail: zengbing2016@126.com)

even worse when facing new dataset on fault classification. No prior experience for learning rate tuning can be learnt from, and a complete full tuning process become necessary. So the learning rate tuning is challenging in CNN based fault classification.

On the other branch, some researchers investigated the self-adaptive learning rate algorithm to adjust the learning rate following certain rules. These rules are denoted as the learning rate schedulers in this research. The learning rate schedulers are effective. But most learning rate schedulers are designed manually or kept static during the training process, which limits their scalability on the application of fault classification. They still hardly provide the good generalization performance for all datasets [14]. So it is vital to find the more powerful learning rate scheduler to adjust the learning rate properly.

In this research, an Automatic Learning Rate Scheduler (AutoLR) is designed for the CNN based fault classification, named AutoLR-CNN. The AutoLR is a Reinforcement Learning (RL) agent, which is trained online using the historical training information of the CNN model, and it can adjust the learning rate on each training step. Firstly, the long short-term memory (LSTM) is used to extract the features of the past training loss of CNN. Secondly, a RL agent based on Deep Deterministic Policy Gradient (DDPG) is trained to control the learning rate for the CNN model online. Thirdly, in AutoLR-CNN, the double CNN structure is constructed for the pre-exploration of the RL agent and avoiding the over-estimation of the leaning rate. The proposed AutoLR-CNN is tested on two public bearing datasets and one practical bearing dataset on wind turbine. The results show that AutoLR-CNN can achieved the state-of-the-art performance on fault classification by compared with six baseline learning rate schedulers in Tensorflow, as well as other reported Machine Learning (ML) and DL methods. As AutoLR-CNN can adjust the learning rate automatically, it is easier to be used and also efficient.

The rest of this paper is organized as follows. Section II introduces the related work. Section III presents the proposed AutoLR. Section IV gives the methodologies of AutoLR-CNN. Section V presents the experimental results of AutoLR-CNN on two case studies and one practical experiments. Finally, Section VI shows the conclusion and future research.

II. Related Work

The related work is described in this section, including CNN based fault classification and learning rate tuning on DL models based fault classification.

A. Convolutional Neural Network based Fault Classification

With the development of smart manufacturing, data driven fault classification has been boosting in recent year [15]. CNN has been widely applied in the field of fault classification [7][8].

Shao et al. [16] studied a highly accurate deep transfer learning by fine-tuning CNN models for machine learning fault classification and achieved good results on induction motor dataset and gearbox dataset. Yang et al. [17] studied a double CNN architecture for the remaining useful life prediction and demonstrated its effectiveness by comparing with the state-of-the-art methods. Huang et al. [2] proposed the decoupling capsule CNN network for the compound fault decoupling. Wang et al. [18] investigated an enhanced CNN for fault classification with multi-sensor image confusion technique and the proposed method has achieved remarkable results. Chen et al. [19] studied the intelligent fault classification method for rotary machinery using transferable CNN and tested the method on the bearing dataset and gearbox dataset.

However, training a powerful CNN model for fault classification highly depends on the learning rate [8]. It can greatly impact the final CNN performance, so the tuning process for leaning rate is necessary for the CNN based fault classification. In this research, a CNN model with AutoLR is proposed, which can adjust the learning rate automatically, and it is efficient and easy to be used.

B. Learning Rate Tuning on DL in Fault Classification

Learning rate has a great impact on the performances of DL methods [11]. He et al. [9] studied the theoretical evidence that the generalization ability of deep neural network has a negative correlation with the ratio of batch size to learning rate. Li et al. [10] investigated the effect on the initial learning rate on the generalization of the CNN models and demonstrated that the large learning rate could achieve better generalization soon after the learning rate is annealed. Daniel et al. [11] presented a set of features suitable to learn a controller for the step size of CNN training algorithms.

Even though there are several previous studies giving the empirical suggestions on the selection of learning rate, but tuning for finding the proper learning rate is still the key to ensure the performance of CNN on the real-world applications. Traditional ‘trial and error’ or manual search methods have been applied to find the optimal learning rate until they obtain the good choices. Xia et al. [20] tried the learning rate value from 0.005 to 0.05 on CNN based fault classification and selected the best learning using ten repeated running. Wang et al. [21] conducted the ‘trial and error’ to tune the hidden layers, the hidden neurons, the learning rate, and the training times on LSTM. Gai et al. [22] applied grasshopper optimization algorithm to optimization the hyper-parameters of deep belief network (DBN) including leaning rate.

The self-adaptive learning rate algorithms have been used on the fault classification field. Shao et al. [23] proposed an adaptive learning rate algorithm to adjust the learning rate at each iteration and the results showed that the proposed method is effective. Tang et al. [24] proposed an adaptive learning rate for the DBN based fault classification on rotating machinery to ensure the generalization. Lu et al. [25] studied the adaptive learning rate algorithm on CNN for the bearing fault classification, and they achieved improved results. The self-adaptive learning rate algorithms are manual designed and they still hardly provide the good generalization performance for all datasets [14] .

Hyper Parameters Optimization (HPO) [12][26] has been applied to find the optimal learning rate values. Snoek et al. [27] applied the pseudo-Bayesian neural network to tune the learning rate of another deep neural network, but the learning

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

rate is fixed in all iterations. Han et al. [28] investigated the methods to tune the learning rate, the batch size, the number of epochs, and the momentum coefficient of the CNN methods for motor fault classification. Saufi et al. [29] studied a review on HPO for DL applications, and pointed out several potential algorithms for the learning rate selection.

In this research, an Automatic Learning Rate Scheduler (AutoLR) is designed for the CNN based fault classification. AutoLR uses the historical CNN training information to construct a RL agent to adjust the learning rate of CNN model at each step, which can make the full use of the training information to guide the control on learning rate automatically. The experiments show its performance in fault classification.

III. Proposed Automatic Learning Rate Scheduler

This section presents the automatic learning rate scheduler (AutoLR) based on DDPG. Firstly, the introduction of DDPG is presented. Secondly, the definition of AutoLR is given.

A. Deep Deterministic Policy Gradient

RL is the method which trains an agent to perform a sequence of decision actions within the environment in order to maximize a cumulative reward. It has been widely studied in artificial intelligent field [30]. Suppose that at time *t*, RL trains the agent to take an action ($a_t \in A$) according its state ($s_t \in S$) and policy ($\pi : S \to A$). The agent will receive its reward (r_{t+1}) from environment and then arrive at the next time step *t+1* with the next state s_{t+1} . RL balances the immediate reward and the ultimate goal through the action-value function (also denoted as *Q*-function) $Q^\pi(s,a):(S,A)\to\mathbb{R}$, which is the discounted expected return of the rewards as shown in equation (1). *T* is the maximum number of time steps. γ is the discounter factor, which is used to prioritize the earlier rewards over later ones.

$$Q^\pi(s,a) = E_\pi\left\{\sum_{k=t}^{T}\gamma^k r_k \,|s_0 = s, a_0 = a\right\} \quad (1)$$

The optimal action-value function $Q^*(s,a)=\max_\pi Q^\pi(s,a)$ satisfies the well-known Bellman equations, as shown in equation (2). When the number of the states and actions are finite, the *Q* function can be viewed as a look-up *Q* table. Otherwise, the state and action space should be discretization in Markov Decision Process (MDP) and Q-learning [30].

$$Q^*(s_t,a_t) = E\left\{r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1},a') \,|\, s_t = s, a_t = a\right\} \quad (2)$$

DDPG [31] is an improved RL algorithm which can handle with the continuous state space and action space based on policy gradient and Deep Q Network (DQN). It is capable to learn from the raw observations and has been applied for solving many tasks in OpenAI Gym and Atari. DDPG is improved from the actor-critic algorithm, and it has two main parts: the actor network and the critic network. Both of the actor network and the critic network has its online subnet and target subnet. But the online subnet and target subnet have the same network structure.

In this research, the actor online subnet is denoted by θ^μ and its target subnet is $\theta^{\mu'}$. The critic online subnet is denoted by ω^Q and its target subnet is $\omega^{Q'}$. The actor network is trained to estimate the deterministic optimal policy instead of choosing the action from a specific distribution. The policy can be denoted by $\mu_\theta(s_t) = a_t$. The critic network is to simulate the Q table for action network. The estimate Q table using critic network can be denoted by $Q(s_t,\mu(s_t)\,|\,\omega^Q)$. It can be seen that DDPG can effectively handle with the continuous state space and continuous action space by simulating the real Q table using neural network. It can avoid the "curse of dimensionality" occurred in MDP and Q-learning which require the discretization of the state and action.

The loss of the critic network is measured by the distance between the two side of the Bellman equation, which can be formulated as equation (3). The optimal Q-value is defined as equation (4). The action policy of actor network is updated by means of the critic network, and it is calculated as equation (5).

$$L = \frac{1}{N}\sum_t\left(y_t - Q(s_t,a_t\,|\,\omega^Q)\right)^2 \quad (3)$$

$$y_t = r_{t+1} + \gamma Q\left(s_{t+1},\mu(s_{t+1})\,|\,\omega^{Q'}\,|\,\theta^{\mu'}\right) \quad (4)$$

$$\nabla_{\theta^\mu} J(\theta) = \frac{1}{n}\sum_{j=1}^{n}\nabla_a Q(s_i,a_i\,|\,\omega^Q)\nabla_{\theta^\mu}\mu(s\,|\,\theta^\mu) \quad (5)$$

Then, the DDPG softly updated the target actor subnet and target critic subnet with the factor $\tau \in (0,1]$.

$$\omega^{Q'} \leftarrow \tau\omega^Q + (1-\tau)\omega^{Q'} \quad (6)$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \quad (7)$$

B. Proposed AutoLR using DDPG

Since DDPG works well on the continuous state space and continuous action space, it is applied for the learning rate adjustment for CNN model in this research. The framework of the proposed AutoLR is shown in Fig. 1.

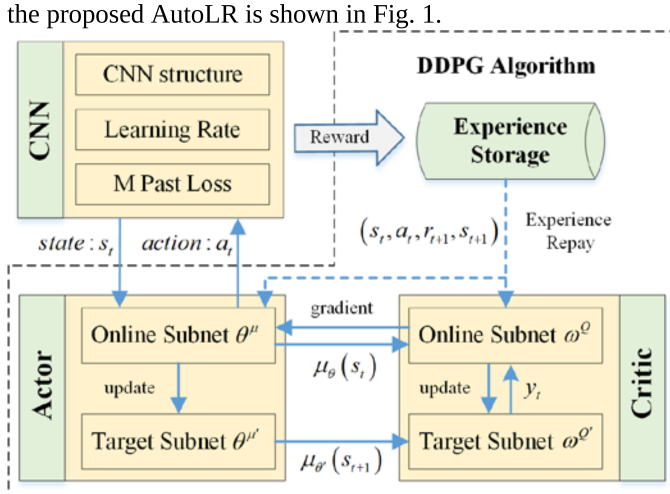

Fig. 1. The actor and critic network in AutoLR

For the framework, it can be seen that CNN network would generate its state s_t and then the actor online subnet yields the action a_t to adjust the learning rate of the CNN at each step.

In order to avoid the pre-defined state, the state of CNN is defined as the past *M* historical training loss of the CNN on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

mini-batch data. The actor networks are constructed using LSTM to extract the feature of the loss information, as well as the critic networks. The definition of actor network and critic network are shown in TABLE I.

TABLE I
THE LAYER CONFIGURATIONS OF ACTOR AND CRITIC NETWORK

Actor Network	Critic Network	
Input state: *M* past loss	Input 1 state: *M* past loss	Input 2: action
LSTM (units: 32)	LSTM (units: 32)	
-	Concatenate layer	
FC1 (neurons: 16, sigmoid)	FC1 (neurons: 16, sigmoid)	
FC2 (neurons: 1, sigmoid)	FC2 (neurons: 1)	

The activation function of Fully-Connected (FC) layer in actor network is sigmoid, so the output (out) of actor network is limited to $(0,1)$. out is scaled according to equation (8), so the output interval of actor network, which is also the range of action, would be $\beta \subset (\alpha, 2-\alpha)$, α is the factor to control the range of action. Meanwhile, the FC2 in critic networks has not activation function, and it directly predicts the reward according to the state and action.

$$\beta = 2(1-\alpha) \times out + \alpha \tag{8}$$

In this research, the action of actor network is used to upgrade the learning rate η of the CNN in fault classification, and it is modeled as equation (9).

$$\eta = \eta \times \beta \tag{9}$$

The reward is defined as the improvement of loss to ensure that the RL agent can find the optimal objective value within the smallest number of iterations. The reward can be presented in equation (10). As reward is the improvement of the loss of CNN models, the convergence of the reward (denoted as the reward curve) can reflect both the convergence of the agent and CNN models.

$$R = loss_t - loss_{t+1} \tag{10}$$

IV. PROPOSED AUTOLR-CNN FOR FAULT CLASSIFICATION

This section presents the proposed AutoLR-CNN for fault classification, including the data preprocessing, the CNN structure and the framework of AutoLR-CNN.

A. Data Preprocessing

S-Transform, which is proposed by Stockwell [32], has been shown to be effective in fault classification, so it is applied for the CNN based fault diagnosis in this research. It can convert the time-domain fault signals to time-frequency domain and suitable for the non-stationary fault signals. S-Transform can be viewed as an extension of short-time Fourier transform (STFT). The STFT of a signal *x(t)* is defined as equation (11).

$$STFT(\tau, f) = \int_{-\infty}^{+\infty} x(t) w(t-\tau) e^{-j2\pi ft} \mathrm{d}t \tag{11}$$

Suppose τ is the time of spectral localization, *f* is the Fourier frequency, and *w(t)* is the window function. S-Transform replaces the window function using the Gaussian function as $w(t) = |f| / \sqrt{2\pi}\, e^{-t^2 f^2/2}$. Then S-Transform can be given by equation (12).

$$S(\tau, f) = \int_{-\infty}^{+\infty} x(t) \frac{|f|}{\sqrt{2\pi}} e^{-(t-\tau)^2 f^2/2} e^{-j2\pi ft} \mathrm{d}t \tag{12}$$

B. CNN Structure

As LeNet-5 has been widely applied into fault classification, an improved CNN structure based on LeNet-5 is studied in this research. The improved CNN structure applied many alternative convolutional and maxpool layers to enhance its feature extraction ability. The CNN structure is presented in TABLE II. 'Conv(7×7×64)' means that it is a convolution layer with the filter size is 7×7, the depth is 64. Maxpool layer is adopted and its pool size is 2×2. The hidden neurons of FC layers are 2560 and 512. The step sizes (the strides parameter in Tensorflow) of Conv and Maxpool layers are all 1×1, and the zero-padding is "SAME" type. Then, two layers of fully-connected (FC) layer are followed. At last, a output FC layer contains a softmax classifer for the fault classification.

TABLE II
THE LAYER CONFIGURATIONS OF CNN MODELS

Layer Name	CNN Operator	Layer Name	CNN Operator
L1	Conv(7×7×64)	L9	Conv(3×3×256)
L2	MaxPool(2×2)	L10	MaxPool(2×2)
L3	Conv(5×5×92)	L11	Conv(3×3×256)
L4	MaxPool(2×2)	L12	MaxPool(2×2)
L5	Conv(3×3×128)	FC1	2560
L6	MaxPool(2×2)	FC2	512
L7	Conv(3×3×256)	FC3	health conditions softmax classifier
L8	MaxPool(2×2)		

C. The Framework of AutoLR-CNN

In this research, the proposed AutoLR based Learning Rate Scheduler is applied on the CNN for fault classification, named AutoLR-CNN. The framework of the proposed AutoLR-CNN is shown in Fig. 2.

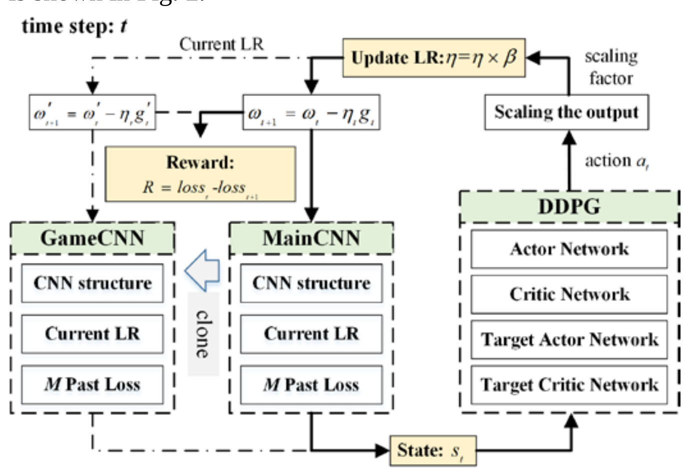

Fig. 2. The framework of the proposed AutoLR-CNN

The structure of AutoLR-CNN is inspired by DDPG. It has double CNN subnets to enhance its stability, named MainCNN network and GameCNN network. The MainCNN is the online version of the CNN for fault classification, and the GameCNN is the clone version of MainCNN. GameCNN is used to pre-exploration for the RL agent to avoid the over-estimation of the trend of learning rate.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

The training of GameCNN would be firstly implemented whenever the MainCNN is trained. Since the GameCNN is the clone version of MainCNN, so it shares the same workflow as MainCNN as shown in the dash line in Fig. 2.

The main working flow of the AutoLR-CNN is as followed. At each step *t*, firstly generate the state s_t of the MainCNN or GameCNN network. Then, the actor network in DDPG predicts its action $a_t = \mu_\theta(s_t)$. Thirdly, the agent will update the learning rate according the action using equation (8-9), and then conduct one mini-batch training step. Finally, the agent will receive its reward r_{t+1}, and arrive at the next time step *t+1*.

The training of the AutoLR-CNN is shown in Algorithm 1. Let ω denotes the weight of Main-CNN and ω' denotes the weight of Game-CNN. *D* is the storage for experience replay.

Algorithm 1: the training algorithm of AutoLR-CNN

Data Pre-processing using S-Transform as equation (12)
Initialize MainCNN network and DDPG network
Initialize the mini-batch of the dataset
for t=1, totalstep **do**
  Clone GameCNN from MainCNN
    clone the network from ω to ω'
    clone the past *M* loss and learning rate from η to η'
    clone the mini-batch of the dataset x_t to x'_t
  for *k* = 1, step4game **do**
    generate the state of GameCNN as s'_{t+k}
    predict the action $a'_{t+k} = \mu_\theta(s'_{t+k})$ and update η' using equation (9)
    Sample the mini-batch x'_{t+k+1}, and update ω'
    Calculate the received reward r'_{t+k+1}
    Store $(s'_{t+k}, a'_{t+k}, r'_{t+k+1}, s'_{t+k+1})$ in experience storage *D*
  end for
  Train DDPG, train critic network using equation (3) and train actor network using equation (5)
  Update the target network in the critic and actor networks
  for *k* = 1, step4game **do**
    generate the state of MainCNN as s_{t+k}
    predict the action $a_{t+k} = \mu_\theta(s_{t+k})$ and update η using equation (9)
    Sample the mini-batch x_{t+k+1}, and update ω
    Calculate the received reward r_{t+k+1}
    Store $(s_{t+k}, a_{t+k}, r_{t+k+1}, s_{t+k+1})$ in experience storage *D*
  end for
  t=t+ step4game
end for

As shown in Algorithm 1, AutoLR-CNN applies the LSTM network to learn the feature of the past *M* loss on mini-batch training and then trains the agent to control the learning rate of CNN automatically. In this research, the proposed AutoLR-CNN will be tested on two famous bearing dataset and one practical wind turbine bearing dataset to validate its potential.

V. Case Studies and Practical Experiment

In this section, the proposed AutoLR-CNN based fault classification is implement by Tensorflow. It is conducted on two famous bearing datasets and one practical bearing dataset, including Case Western Reserve University (CWRU) bearing dataset, bearing damage dataset provided by KAT datacenter in Paderborn University, and the bearing data of a practical wind turbine.

A. Experiment Setup and Baseline

The AutoLR-CNN is based on the stochastic gradient descent (SGD). The parameters of AutoLR-CNN are: batch size is 40, SGD is applied while it learning rate is controlled by AutoLR. *step4game*=5. The parameters for actor and critic network are: epoch is 50, AdamOptimizer is applied to train actor and critic network and its learning rate is 1e-4. During the experiments, the five-fold Cross validation (CV) is applied, as CV it is a popular technique to obtain the reliable performance evaluation of fault classifiers [33].

In order to validate the performance of AutoLR-CNN, six most popular learning rate schedulers are selected as the baseline methods. These learning rate schedulers have been widely applied and implemented in Tensorflow software. AutoLR-CNN is compared with them to show its efficient and potential. The results of AutoLR-CNN are also compared with the other DL and ML methods to validate its performance on fault classification further.

The selected six baseline learning rate schedulers are:
1) C-CNN: Constant Learning Rate;
2) ED-CNN: Exponential Decay;
3) NED-CNN: Natural Exponential Decay;
4) ITD- CNN: Inverse Time Decay;
5) PD-CNN: Polynomial Decay;
6) CD-CNN: Cosine Decay.

The running environment of all the experiments for AutoLR-CNN and the baseline learning rate schedulers are the Ubuntu 16.04 desktop with RTX 2080Ti GPU. The CPU is Intel Core i9-9900X 3.5GHz*20. The memory is 128GB.

B. Case Study 1: CWRU Motor Bearing Dataset

In this case study, the proposed AutoLR-CNN is tested on the motor bearing dataset from Case Western Reserve University (CWRU) [34]. In this dataset, there are three fault patterns, namely roller fault (RF), outer race fault (OF) and inner race fault (IF), and each fault pattern has three damage severity which are 0.18mm, 0.36mm and 0.54mm. These nine fault patterns and the normal condition consist total ten health conditions for the bearings. The driven end vibration signals are collected for the fault classification. The experiments are conducted on four load conditions which the motor load is set to be 0, 1, 2, 3 *hp* respectively, as shown as TABLE III.

TABLE III
The Configuration of CWRU Dataset

No	Motor Load (HP)	Approx. Motor Speed (rpm)
0	0	1797
1	1	1772
2	2	1750
3	3	1730

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

TABLE IV
COMPARISON RESULTS OF AUTOLR-CNN WITH BASELINE IN CASE 1 (%)

Method	1-2		1-3		2-1		2-3		3-1		3-2		*AVG*
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	
AutoLR-CNN	**99.88**	**0.0005**	**98.64**	**0.0013**	**97.92**	**0.0026**	**98.72**	**0.0074**	**92.69**	**0.0246**	**97.80**	**0.0189**	**97.61**
C-CNN	99.90	0.0004	98.42	0.0037	98.04	0.0010	98.19	0.0161	90.89	0.0203	96.01	0.0185	96.91
ED-CNN	99.89	0.0006	95.68	0.0092	98.01	0.0006	98.55	0.0140	90.86	0.0088	97.91	0.0095	96.82
NED-CNN	99.92	0.0002	97.77	0.0100	97.95	0.0008	97.27	0.0337	91.60	0.0137	98.25	0.0067	97.13
ITD- CNN	99.77	0.0027	95.02	0.0258	97.95	0.0012	98.59	0.0048	90.06	0.0215	96.51	0.0187	96.32
PD-CNN	99.92	0.0002	98.67	0.0040	98.07	0.0006	98.47	0.0066	91.29	0.0099	97.63	0.0094	97.34
CD-CNN	99.89	0.0002	98.88	0.0034	97.94	0.0011	98.39	0.0097	91.14	0.0181	97.42	0.0122	97.28

The samples are randomly selected from the vibration signals from ten health conditions. The sample size is 1200 for each health condition, while 1000 samples are used for training dataset and the rest 200 are testing dataset. So there are 10000 samples in training dataset and 2000 in testing dataset totally. Each sample contains 1024 points. It would be handled by the S-Transform which is convert to the 2D matrix format, and then be resized to 224*224 before feeding to AutoLR-CNN. The task of this case study is to verified the performance of the AutoLR-CNN on cross working load. The configuration of this case studies is form Zhu et al. [34], which used 1, 2, 3 working conditions, and the notation of '1-2' means that AutoLR-CNN is trained on load 1 and tested on load 2.

1) Compared with Baseline Methods

In this subsection, the proposed AutoLR-CNN is compared with the six baseline learning rate schedulers. Except the mean prediction accuracies (*Mean*), its standard deviation (*Std*) is also selected as the comparison term to validate the performance of AutoLR-CNN. The average prediction accuracies (*AVG*) is also presented. The comparison results of AutoLR-CNN and the baseline learning rate schedulers are shown in TABLE IV.

From TABLE IV, it can be seen that AutoLR-CNN and all six baseline learning rate schedulers methods achieve very close results on the first four configuration pairs, including the 1-2, 1-3, 2-1 and 2-3. But the AutoLR-CNN obtains dominant good results on the 3-1 and 3-2 configuration pairs. The *AVG* of AutoLR-CNN is 97.61%, which outperforms these values achieved by the six baseline learning rate schedulers, and they are 96.91%, 96.82%, 97.13%, 96.32%, 97.34% and 97.28%. The *Std* values of AutoLR-CNN is also comparable with other baseline learning rate schedulers, which means that the AutoLR-CNN is as stable as the baseline learning rate schedulers.

2) Analysis and Discussion on Reward and Learning Rate

Fig. 3 shows the reward curves of the AutoLR-CNN. Four typical reward curves are presented. From the results, it can be seen that the rewards values are increased sharply at the beginning, indicating that AutoLR-CNN has been optimized well. Then, the values of reward decrease along with the training process. At the end of training, the reward values are near zero. As the reward is defined as the loss improvement of the CNN models, these results validate the convergence of AutoLR-CNN.

Fig. 4 presents the learning rate curves of the AutoLR-CNN. It can be seen that the learning rate schedulers have several different types. As all training processes have been conversed, the reasonable of the proposed AutoLR is validated. Unlike most baseline learning rate schedulers, which always set a big value to learning rate and decay it during the training process, AutoLR can increase or decrease the learning rate during the training process online. As shown in Fig. 4, all learning rate curves obtained by AutoLR have the increase-decrease or decrease-increase parts, which show that AutoLR is trying to control the learning rate to lead the training process toward to convergence.

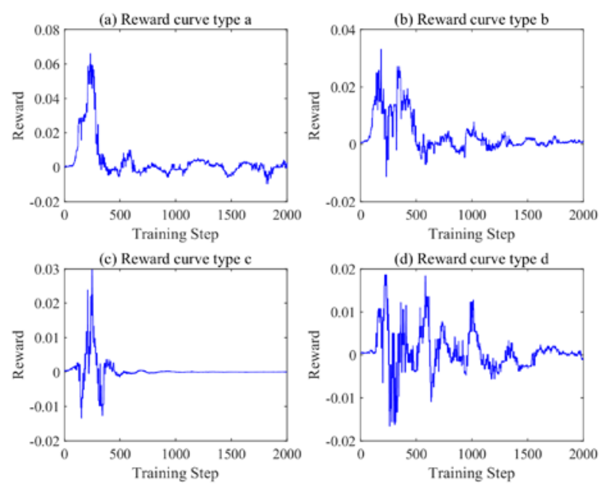

Fig. 3. The reward curves of AutoLR-CNN in Case 1

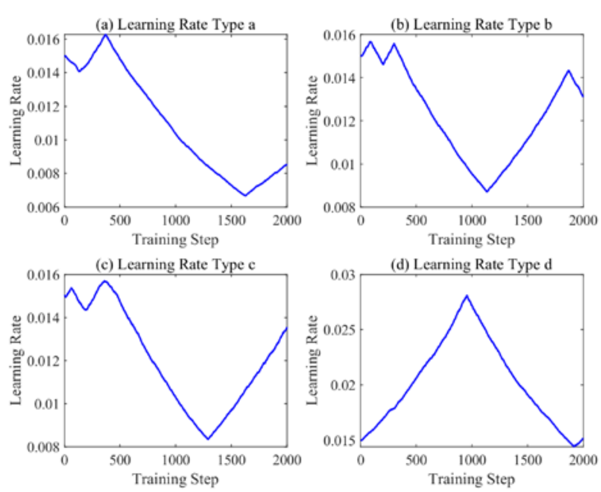

Fig. 4. The typical learning rate curves of AutoLR-CNN in Case 1

3) Compared with Other DL and ML Methods

The results of AutoLR-CNN are also compared with the published DL and ML methods in the literature to validate its performance further. The comparison methods are the CNN based on a capsule network with an inception block (ICN)[34], hierarchical CNN (HCNN)[35], Deep CNN with Wide First-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

layer Kernels with AdaBN (WDCNN-AdaBN) [36], Ensemble CNN with Training Interference (Ensemble TICNN) [37]. Other famous DL and ML methods, including ResNet, AlexNet, SVM and ANN, are also selected for the comparison. The comparison results are presented in TABLE V.

TABLE V
COMPARISON RESULTS WITH OTHER DL AND ML METHODS IN CASE 1 (%)

No.	1-2	1-3	2-1	2-3	3-1	3-2	*AVG*
AutoLR-CNN	**99.88**	**98.64**	**97.92**	**98.72**	**92.69**	**97.80**	**97.61**
ICN	98.23	97.17	99.80	94.71	94.93	98.10	97.15
HCNN	99.93	98.79	95.15	99.45	89.33	93.69	96.1
WDCNN-AdaBN	99.4	93.4	97.5	97.2	88.3	99.9	95.9
Ensemble TICNN	99.5	91.1	97.6	99.4	90.2	98.7	96.1
ResNet	99.70	94.40	94.87	94.33	88.70	98.47	94.58
AlexNet	98.93	92.27	95.07	94.40	88.40	96.87	94.32
SVM	68.6	60.0	73.2	67.6	68.4	62.0	66.6
ANN	82.1	85.6	71.5	82.4	81.8	79.0	80.4

From TABLE V, it can be seen that the *AVG* results of AutoLR-CNN is 97.61%, which is better than ICN, HCNN, WDCNN-AdaBN, Ensemble TICNN. The average prediction accuracies of ResNet, AlexNet, SVM and ANN are 94.58%, 94.32%, 66.6% and 80.4%, which are significantly inferior to AutoLR-CNN. These comparison results show the significant potential of the proposed AutoLR-CNN for fault classification.

C. Case Study 2: KAT Bearing Dataset

In this subsection, the proposed AutoLR-CNN method is conducted on the fault classification of bearing dataset provided by KAT datacenter in Paderborn University [34]. In this dataset, there are 15 datasets and they can be categorized as three healthy classifications. The K0-series (K001~K005) are the healthy condition. The KA-series (KA04, KA15, KA16, KA22, KA30) are the outer bearing ring with damage. The KI-series (KI04, KI14, KI16, KI18, KI21) are the inner bearing ring with damage. The experiments are conducted with four different operating parameters as shown in TABLE VI, and the vibrations signals are collected for analysis. The sampling rate is 64 kHz.

TABLE VI
THE CONFIGURATION OF KAT DATASET

No.	Rotational speed	Load torque	Radial force
4	900	0.7	1000
5	1500	0.1	1000
6	1500	0.7	400
7	1500	0.7	1000

The samples are randomly selected from the vibration signals from15 datasets, and the samples size is 1000 for each dataset. The ratio between training dataset and testing dataset are 9:1, so there are 13500 (15*1000*9/(9+1)) samples in the training dataset and 1500 (15*1000*1/(9+1)) in the testing dataset. Each sample contains 10240 points and it would be converted to 2D format by S-Transform technique. The learning task is also to verify AutoLR-CNN on the cross working load. The comparison configuration is from Zhu et al. [34], and the *AVG* of six configuration pairs of 5-6, 5-7, 6-5, 6-7, 7-5 and 7-6 is used as the performance index. The five-fold cross validation implemented for AutoLR-CNN and all the baseline methods.

1) Compared with Baseline Methods

In this subsection, the proposed AutoLR-CNN is compared with the six baseline learning rate schedulers. The results of *Mean* and *Std* of the five-fold cross validation on each configuration pair are presented in TABLE VII. The *AVG* of six configuration pairs are also presented in TABLE VII.

From TABLE VII, it can be seen that AutoLR-CNN has achieved three best *Mean* values and three best *Std* values out of six configuration pairs. The AVG of AutoLR-CNN is 91.43%, which is better than all six baseline learning rate schedulers. From the results, it can be seen that the proposed AutoLR-CNN not only can achieve better *Mean* accuracies, but also tend to have smaller *Std* values, which means it is more stable than the baseline methods in this case study.

2) Analysis and Discussion on Reward and Learning Rate

Fig. 5 shows the reward curves of the AutoLR-CNN on case 2. In this case, the reward curves are much complex than case 1. Two typical types of reward curves are presented. In type a, the reward firstly increases sharply, but then decreases to the negative value along with the training process. Finally, the reward converses to near zero. In type b, the reward firstly decreases, but then increases to a high value and also converses to near zero. These results can validate the convergence of the rewards of AutoLR-CNN.

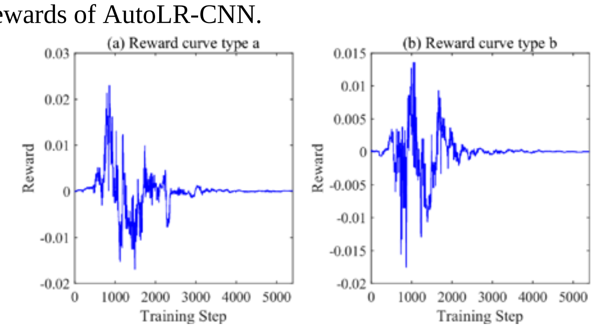

Fig. 5. The reward curves of AutoLR-CNN in Case 2

The learning rate curves of AutoLR in this case study is presented in Fig. 6. The learning rate presents almost the same curves as case study 1. The learning rate curves are various and

TABLE VII
COMPARISON RESULTS OF AUTOLR-CNN WITH BASELINE IN CASE 2 (%)

Method	5-6		5-7		6-5		6-7		7-5		7-6		*AVG*
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	
AutoLR-CNN	**86.85**	**0.0047**	**99.39**	**0.0024**	**86.55**	**0.0045**	**89.89**	**0.0067**	**96.49**	**0.0064**	**89.41**	**0.0051**	**91.43**
C-CNN	87.07	0.0061	99.31	0.0037	86.09	0.0153	88.49	0.0169	96.55	0.0053	89.65	0.0056	91.19
ED-CNN	84.49	0.0090	96.78	0.0094	84.86	0.0095	88.85	0.0117	94.51	0.0052	88.61	0.0078	89.68
NED-CNN	82.59	0.0143	96.11	0.0134	84.67	0.0071	88.13	0.0090	93.01	0.0052	86.22	0.0137	88.46
ITD-CNN	83.65	0.0093	96.93	0.0099	84.66	0.0036	88.53	0.0201	94.48	0.0077	87.33	0.0068	89.26
PD-CNN	86.74	0.0016	99.01	0.0046	86.25	0.0045	89.05	0.0058	96.36	0.0034	89.37	0.0027	91.13
CD-CNN	86.96	0.0057	98.99	0.0038	86.04	0.0033	89.35	0.0070	95.78	0.0021	89.19	0.0029	91.05

all curves contain the increase-decrease or decrease-increase parts, indicating that AutoLR is trying to find the proper learning rate during the training process. It should be noted that most learning rate are not the decay curves. But from Fig. 5, it can be seen that the CNN models of AutoLR-CNN have all been conversed.

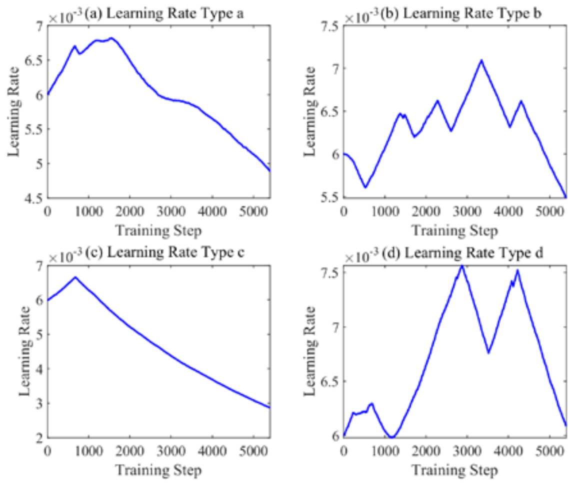

Fig. 6. The typical learning rate curves of AutoLR-CNN in Case 2

3) *Compared with other DL and ML Methods*

In this subsection, the results of AutoLR-CNN is compared with several published DL and ML methods, and they are DAN[38], ICN[34], WDCNN[34], ACDIN[34], ResNet[34], AlexNet[34], SVC[34], and KNN[34]. The comparison results are presented in TABLE VIII.

TABLE VIII
COMPARISON RESULTS WITH OTHER DL AND ML METHODS IN CASE 2 (%)

No.	5-6	5-7	6-5	6-7	7-5	7-6	*AVG*
AutoLR-CNN	**86.85**	**99.39**	**86.55**	**89.89**	**96.49**	**89.41**	**91.43**
DAN	85.7	98.4	81.58	89.29	98.00	90.50	90.58
ICN	80.67	96.97	70.23	70.67	94.27	79.50	82.05
WDCNN	72.33	94.70	69.33	69.77	93.67	70.27	78.35
ACDIN	79.43	78.73	85.07	90.53	79.53	75.60	81.48
ResNet	71.33	96.67	64.53	67.23	92.73	72.60	77.52
AlexNet	78.87	98.47	65.93	66.20	96.03	74.07	79.92
SVC	81.20	92.97	72.13	77.23	93.00	72.77	81.55
KNN	72.13	93.27	60.47	58.60	90.67	74.13	74.88

From the results, it can be seen that AutoLR-CNN has achieved significant prediction accuracy among these methods. AutoLR-CNN obtains four best values out of six configuration pairs and the best *AVG* results. The prediction of AutoLR-CNN is 91.43%, which outperforms DAN, ICN, WDCNN and ACDIN. The results of ResNet and AlexNet are 77.52% and 79.92%, which means that AutoLR-CNN has improved 19.57% and 15.98% than them. Traditional ML methods, including SVC and KNN methods, are also compared with AutoLR-CNN, and the results show that AutoLR-CNN has achieved better results than them.

D. *Practical Experiments: The Bearing Dataset from Wind Turbine (WT)*

In this section, a practical experiment is conducted for analyzing of the bearing damage of the wind turbine (WT) from a wind power equipment company in China. The drive chain of direct drive wind turbine has no gearbox, and the main bearing is connected with the generator to transfer torque. The main bearing is not only a transmission component, but also a supporting structure. Due to the low running speed of the direct drive wind turbine, three low-frequency acceleration sensors are installed to measure vibration signal of the axial, vertical, horizontal directions of the main spindle. The installation is shown as Fig. 7.

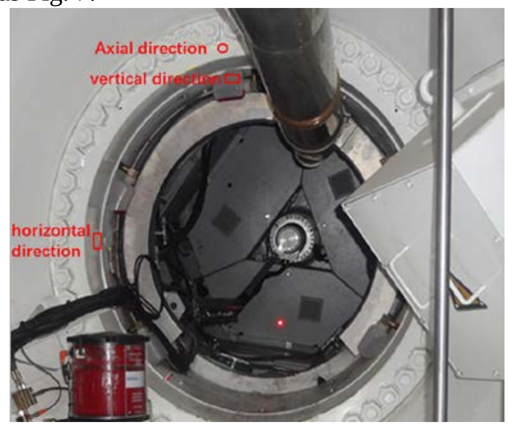

Fig. 7. The equipment of the bearing on wind turbine in practical experiment

The selected acceleration sensor is the RH113. The parameters of this sensor are 1) sensitivity: 500mv/g, 2) frequency range: 0.1~10*k*Hz, 3) range: ±10g. The sampling frequency is 2560 Hz, and the sampling time is 6.4 seconds. So there is total 16384 points for each signal. Three healthy conditions are investigated in this experiment, and they are the normal condition, slight wear and serious damage. The experiment is repeated by 4 times. The samples size is 500 for each signal, and 400 samples are used for the training dataset while the rest 100 samples are the testing dataset. So there are totally 4800 (400*3*4) samples in training dataset and 1200 (100*3*4) in testing dataset. Each sample contains 2048 points and then it would be converted to 2D format by using S-Transform technique, and handled by AutoLR-CNN. The learning task of this case study is to validate AutoLR-CNN further, and five-fold cross validation is implemented for AutoLR-CNN and the baseline methods.

1) *Compared with Baseline Methods*

In this subsection, the results of AutoLR-CNN are compared with the six baseline learning rate schedulers, and the comparison results are shown in TABLE IX.

TABLE IX
COMPARISON RESULTS OF AUTOLR-CNN WITH BASELINE IN PRACTICAL EXPERIMENT (%)

Method	*Mean*	*Std*
AutoLR-CNN	**99.52**	**0.0038**
C-CNN	99.27	0.0043
ED-CNN	99.20	0.0042
NED-CNN	86.98	0.0337
ITD-CNN	93.72	0.0338
PD-CNN	97.98	0.0023
CD-CNN	98.08	0.0012

From the results, it can be seen that the *Mean* accuracy of AutoLR-CNN is as high as 99.52%, which is the best among these baseline learning rate schedulers. The Std value of AutoLR-CNN is 0.0038, which is a small value among these

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

schedulers. These results indicate that AutoLR-CNN is better than C-CNN, ED-CNN, NED-CNN and ITD-CNN.

2) Analysis and Discussion on Reward and Learning Rate

Fig. 8 presents the reward curves of the AutoLR-CNN on practical experiment. In this experiment, the reward curves are almost the same with case 2. In type a, the reward curve is a 'increase-decrease' curve and finally converse near to zeros. Type b is a 'decrease-increase' curve. As all the reward curves have been conversed to zeros, it can validate the convergence of the rewards of AutoLR-CNN in the practical experiment.

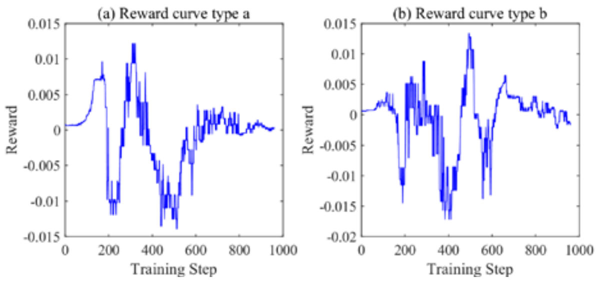

Fig. 8. The reward curves of AutoLR-CNN in practical experiment

The typical learning rate curves of AutoLR-CNN are presented in Fig. 9. The learning rate curves can be categorized into two types. The type a is the linear decrease type. The type b is the linear increase type. Both types contain some disturbances. The learning rate types in this case are simpler than case 1 and case 2. But both types of learning rate curves contain the increase-decrease or decrease-increase parts, showing that AutoLR is also trying to find a good learning rate value online in this case study.

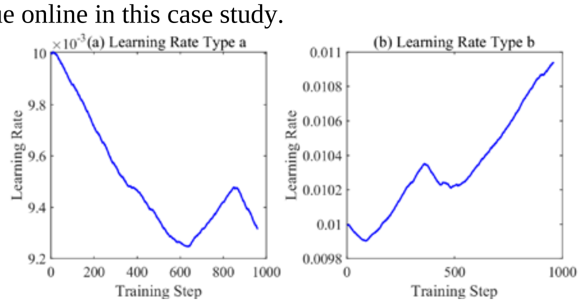

Fig. 9. The typical learning rate curves of AutoLR-CNN in practical experiment

E. Discussion

In this subsection, the time issues of AutoLR-CNN and the comparison with Grid Search (GS) are presented. Firstly, the comparison on time consumption of AutoLR-CNN with the baseline methods are presented. Secondly, AutoLR-CNN is compared with GS, which is the popular HPO method, to show that AutoLR-CNN is efficient and easy to use.

1) Comparison on Time Comparison with Baseline Methods

The time consumption of AutoLR-CNN is presented on TABLE X. The time consumptions of AutoLR-CNN using five-fold cross validation are 7304.5, 19126.5 and 3138.0 seconds on two case studies and the practical experiment. During the training process of AutoLR-CNN, it trains an RL agent within the training process, this causes about 50%~60% of the total time consumption in AutoLR-CNN.

TABLE X
THE TIME CONSUMPTIONS OF AUTOLR-CNN AND BASELINE METHODS

Method	CWRU Dataset	KAT Dataset	WT Dataset
AutoLR-CNN	**7304.5**	**19126.5**	**3138.0**
C-CNN	1530.5**A*	3503.1**A*	546.9**A*
ED-CNN	1453.3**A*	3504.8**A*	569.8**A*
NED-CNN	1536.2**A*	3456.7**A*	554.5**A*
ITD-CNN	1474.9**A*	3714.2**A*	555.3**A*
PD-CNN	1468.5**A*	3636.9**A*	548.1**A*
CD-CNN	1466.5**A*	3642.4**A*	577.9**A*

As shown in TABLE X, the factor *A* means the time consumption for the turning process for the baseline methods. As the learning rate tuning process includes the selection of learning rate schedulers and the tuning process on the selected learning rate schedulers, the tuning space suffers from the 'curse of dimensions'. So the value of factor *A* for tuning process is always large than 10 [8][12]. But AutoLR-CNN doesn't have this factor, because it contains the automatic learning rate scheduler. These results meaning that the AutoLR-CNN is potential on the time consumption compared with the baseline methods.

2) Comparison with Grid Search

In this subsection, the AutoLR-CNN is compared with GS, which has been widely applied in the tuning process of DL and ML fields. The ED-CNN is selected for using GS to tuning its learning rate scheduler. The pre-defined initial learning rates are {1e-2, 1e-3, 1e-4} and its decay factors are {0.999, 0.995, 0.990}. These learning rate and decay values are widely applied as the default values for DL and ML. The results of GS are shown in TABLE XI. It should be noted that the decay is a hyper-parameter to control learning rate in ED-CNN, and it is as shown in equation (13). η_{init} is the initial learning rates, and *t* is the current training step while *decay_step* is the parameter to control the decay of learning rate. And decay_step is set to be 1 in this research.

$$\eta = \eta_{init} \times decay^{\frac{t}{decay\_step}} \tag{13}$$

TABLE XI
THE RESULTS OF GS ON CWRU AND KAT DATASET (%)

CWRU Dataset				KAT Dataset			
Load	1	2	3	Load	5	6	7
1	-	99.95	96.20	5	-	84.77	96.97
2	98.05	-	99.20	6	83.07	-	87.53
3	91.60	98.40	-	7	93.53	87.73	-
AVG	97.23			AVG	88.93		
Time(s)	15914.1			Time(s)	44156.1		

As shown in TABLE XI, the best *AVG* of GS on CWRU dataset is 97.23%, and the best configurations of GS are learning rate value 1e-2 and the decay value 0.999. The AVG results of GS is inferior to AutoLR-CNN, while the time consumption of GS is 15914.1 seconds, which is much higher than AutoLR-CNN. The results of GS on KAT dataset is almost the same. the best configurations of GS on KAT are also learning rate value 1e-2 and the decay value 0.999. The AVG results of GS on KAT is 88.93%, which is inferior to AutoLR-CNN too. The time consumption of GS is 44156.1 second, which is much higher than AutoLR-CNN. The AVG of GS on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2020.3048792, IEEE Transactions on Instrumentation and Measurement

WT dataset is 98.75%, and the time consumption is 5912.73 seconds. The AVG and the time consumption of GS on WT dataset are both worse than AutoLR-CNN.

The GS results with different configurations on learning rate and decay values are presented in TABLE XII. These results are the cross validation results on the GS. It can be seen that only the configuration of {learning rate value :1e-2, decay value: 0.999} obtains the good results, and its cross validation results on CWRU, KAT and WT are 97.45%, 88.14% and 97.97%. However, all these learning rate values and decay values are commonly used as default values in various applications, but most of them are not suitable for these dataset. These results can provide the solid support for the importance on the selection of the learning rate in order to obtain a powerful performance of CNN based fault classification.

TABLE XII
THE CROSS VALIDATION RESULTS OF DIFFERENT GS CONFIGURATION (%)

Learning Rate	decay	**CWRU**	**KAT**	**WT**
1e-2	0.999	97.45	88.14	97.97
	0.995	43.03	67.07	85.97
	0.990	40.11	63.46	76.35
1e-3	0.999	47.03	67.31	78.45
	0.995	35.92	57.95	70.67
	0.990	27.52	52.68	68.52
1e-4	0.999	30.91	60.07	48.38
	0.995	21.62	41.29	55.50
	0.990	18.19	35.54	36.67

From these results. it shows that AutoLR-CNN can achieve better results than the same CNN network with the default learning rate values on these three cases, and it is also efficient on the time consumption. As AutoLR-CNN can make the full use of the training information, it can search the learning rate space more efficient and avoid the effect on the manual selection on learning rate, AutoLR-CNN can obtain the state-of-the-art performance and it also easy to be used on the fault classification.

VI. CONCLUSION AND FUTURE RESEARCH

This research presents a CNN with Automatic Learning Rate Scheduler (AutoLR) for fault classification (AutoLR-CNN). The main contributions of this research are: 1) a AutoLR is designed. It applies the LSTM network to extract the features of the past loss CNN and then trains an RL agent to control the learning rate of CNN online in the automatic way. 2) a double CNN structure is developed to enhance the stability of the proposed AutoLR-CNN. The proposed AutoLR-CNN is conducted on the CWRU bearing dataset, the KAT bearing dataset and the practical bearing dataset on wind turbine. The results show that AutoLR-CNN outperform the six popular learning rate schedulers in Tensorflow. AutoLR-CNN are also compared with ML and DL methods, and the comparison results show that AutoLR-CNN is superior to them, validating its potential performance on fault classification.

The limitations of this research include the following two parts: 1) the actor network and critic network should be trained on every separated running, and it consumes 50%~60% of the total time for each runing. 2) the learning rate is control by the AutoLR-CNN, but the batch size and the regulation term are still pre-determined using traditional methods. Based on these limitations, the future research can be conducted in the following ways. Firstly, the transfer learning can be combined with AutoLR-CNN to reuse the actor network and critic network across the dataset which can significantly reduce the training times of them. Secondly, AutoLR-CNN can be extended to control the batch size and the regulation as well as learning rate further.

REFERENCES

[1] G. Xu, M. Liu, Z. Jiang, W. Shen and C. Huang, “Online Fault Diagnosis Method Based on Transfer Convolutional Neural Networks,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 509-520, 2020

[2] R. Huang, J. Li, W. Li and L. Cui, “Deep Ensemble Capsule Network for Intelligent Compound Fault Diagnosis Using Multisensory Data,” *IEEE Transactions on Instrumentation and Measurement*. vol. 69, no. 5, pp. 2304-2314, 2020.

[3] H. N Liu, C. L. Liu and Y. X. Huang, “Adaptive Feature Extraction Using Sparse Coding for Machinery Fault Diagnosis.” *Mechanical Systems and Signal Processing*, vol. 25, no. 2, pp. 558-574, 2011.

[4] W. Huang, G. Gao, N. Li, X. Jiang and Z. Zhu, “Time-Frequency Squeezing and Generalized Demodulation Combined for Variable Speed Bearing Fault Diagnosis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2819-2829, 2019.

[5] Z. P. Liu and L. Zhang, “A Review of Failure Modes, Condition Monitoring and Fault Diagnosis Methods for Large-Scale Wind Turbine Bearings.” *Measurement*, vol. 149, pp. 107002, 2020.

[6] Y. G Lei, B. Yang, X. W. Jiang, F. Jia, N. P. Li and A. K. Nandi, “Applications of machine learning to machine fault diagnosis: A review and roadmap,” *Mechanical Systems and Signal Processing*, vol. 138, pp. 106587, 2020.

[7] N. F. Waziralilah, A. Abu, M. H. Lim, L. K. Quen and A. Elfakharany, “A Review on Convolutional Neural Network in Bearing Fault Diagnosis.” *MATEC Web of Conferences*, vol. 255, p. 06002. EDP Sciences, 2019.

[8] P. Probst, A. L. Boulesteix and B. Bischl, “Tunability: Importance of Hyperparameters of Machine Learning Algorithms,” *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1-32, 2019.

[9] F. X. He, T. L. Liu and D. C. Tao, “Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence.” *Advances in Neural Information Processing Systems*, pp. 1143-1152. 2019.

[10] Y. Z. Li, C. Wei and T. Ma, “Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks.” *Advances in Neural Information Processing Systems*, pp. 11674-11685. 2019.

[11] C Daniel, J Taylor and S Nowozin, “Learning Step Size Controllers for Robust Neural Network Training,” *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1519-1525, North America, 2016.

[12] F. Hutter, L. Kotthoff, and J. Vanschoren, “Automated Machine Learning-Methods, Systems, Challenges”. *Automated Machine Learning*, 2019.

[13] K. Nar and S. Sastry, “Step Size Matters in Deep Learning.” *Advances in Neural Information Processing Systems*, pp. 3436-3444. 2018.

[14] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford and N. De Freitas, “Learning to Learn by Gradient Descent by Gradient Descent,” *Advances in neural information processing systems*, 2016, 3981-3989

[15] J. Wang, C. Xu, Z. Yang, J. Zhang and X. Li, “Deformable Convolutional Networks for Efficient Mixed-type Wafer Defect Pattern Recognition,” *IEEE Transactions on Semiconductor Manufacturing*, 2020. DOI: 10.1109/TSM.2020.3020985.

[16] S. Shao, S. McAleer, R. Yan and P. Baldi, “Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2446-2455, 2019.

[17] B. Yang, R. Liu and E. Zio, “Remaining Useful Life Prediction Based on a Double-Convolutional Neural Network Architecture,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9521-9530, Dec. 2019.

[18] H. Wang, S. Li, L. Song, L. Cui and P. Wang, “An Enhanced Intelligent Diagnosis Method Based on Multi-Sensor Image Fusion via Improved

Deep Learning Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 2648-2657, 2020.

[19] Z. Chen, K. Gryllias and W. Li, "Intelligent Fault Diagnosis for Rotary Machinery Using Transferable Convolutional Neural Network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 339-349, 2020.

[20] M. Xia, T. Li, L. Xu, L. Z. Liu and C. W. Silva, "Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101-110, 2018.

[21] W. F. Wang, X. H. Qiu, C. S. Chen, B. Lin and H. M. Zhang, "Application Research On Long Short-Term Memory Network in Fault Diagnosis," *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, Chengdu, pp. 360-365, 2018.

[22] J. B. Gai, J. X. Shen, H. Wang and Y. F. Hu, "A Parameter-Optimized DBN Using GOA and Its Application in Fault Diagnosis of Gearbox," *Shock and Vibration*, Article ID 4294095, 2020.

[23] H. D. Shao, H. K. Jiang, K. Zhao, D. D Wei and X. Q. Li, "A Novel Tracking Deep Wavelet Auto-Encoder Method for Intelligent Fault Diagnosis of Electric Locomotive Bearings," *Mechanical Systems and Signal Processing*, vol. 110, pp. 193-209, 2018.

[24] S. H. Tang, C. Q. Shen, D. Wang, S. Li, W. G. Huang and Z. K. Zhu, "Adaptive Deep Feature Learning Network with Nesterov Momentum and Its Application to Rotating Machinery Fault Diagnosis," *Neurocomputing*, vol. 305, pp. 1-14, 2018.

[25] K. W. Lu, T. R. Lin, J. Z. Xue, J. Shang and C. Ni, "An Automated Bearing Fault Diagnosis Using a Self-Normalizing Convolutional Neural Network," *2019 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, Zhangjiajie, Hunan, China, pp. 908-912, 2019.

[26] L. Wen, X. Li and L. Gao, "A New Reinforcement Learning based Learning Rate Scheduler for Convolutional Neural Network in Fault Classification," *IEEE Transactions on Industrial Electronics*, 2020. doi: 10.1109/TIE.2020.3044808.

[27] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram and R. Adams, "Scalable Bayesian Optimization Using Deep Neural Networks," *International Conference on Machine Learning*, pp. 2171-2180, 2015.

[28] J. Han, D. Choi, S. Hong and H. Kim, "Motor Fault Diagnosis Using CNN Based Deep Learning Algorithm Considering Motor Rotating Speed," *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, Tokyo, Japan, 2019, pp. 440-445.

[29] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong and M. H. Lim, "Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis: A Review," *IEEE Access*, vol. 7, pp. 122644-122662, 2019.

[30] C. Qiu, Y. Hu, Y. Chen and B. Zeng, "Deep Deterministic Policy Gradient (DDPG)-Based Energy Harvesting Wireless Communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8577-8588, Oct. 2019.

[31] J. Duan, D. Shi, R. S. Diao, H. F. Li, Z. W. Wang, B. Zhang, D. Bian and Z. H. Yi, "Deep-Reinforcement-Learning-Based Autonomous Voltage Control for Power Grid Operations," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 814-817, Jan. 2020.

[32] R. G. Stockwell, L. Mansinha and R. P. Lowe, "Localization of The Complex Spectrum: The S Transform," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 998-1001, 1996.

[33] T. W. Raube, F. Assis Boldt and F. M. Varejão, "Heterogeneous Feature Models and Feature Selection Applied to Bearing Fault Diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 637-646, 2015.

[34] Z. Y. Zhu, G. L. Peng, Y. H. Chen and H. J. Gao, "A Convolutional Neural Network based on A Capsule Network with Strong Generalization for Bearing Fault Diagnosis," *Neurocomputing*, vol. 323, pp. 62-75, 2019.

[35] L. Wen, X. Y. Li and L. Gao L, "A New Two-Level Hierarchical Diagnosis Network based on Convolutional Neural Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 330-338, 2020.

[36] W. Zhang, G. Peng, C. Li, Y. Chen and Z. Zhang, "A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals," *Sensors*, vol. 17, no. 2, pp. 425, 2017.

[37] W. Zhang, C. H. Li, G. L. Peng, Y. H. Chen and Z. J. Zhang, "A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis Under Noisy Environment and Different Working Load," *Mechanical Systems and Signal Processing*, vol. 100, pp. 439-453, 2018.

[38] Y. Song, Y. Li, L. Jia and M. Qiu, "Retraining Strategy based Domain Adaption Network for Intelligent Fault Diagnosis," *IEEE Transactions on Industrial Informatics*, DOI: 10.1109/TII.2019.2950667, 2019.

Long Wen received his Ph.D. degree in industrial engineering from Huazhong University of Science and Technology, China, 2014.
He is a Professor in the School of Mechanical Engineering and Electronic Information, China University of Geosciences. He had published more than 20 refereed papers, and his research interests include deep learning, automatic machine learning, fault diagnosis and intelligent algorithm, etc.

Liang Gao (M'08-SM'20) received Ph.D. degree in mechatronic engineering from Huazhong University of Science and Technology, China, 2002.
He is a Professor of the Department of Industrial & Manufacturing System Engineering, State Key Laboratory of Digital Manufacturing Equipment & Technology, School of Mechanical Science & Engineering, HUST. He had published more than 170 refereed papers. His research interests include operations research and optimization, big data and machine learning etc.

Xinyu Li (M'19) received his Ph.D. degree in industrial engineering from Huazhong University of Science and Technology, China, 2009.
He is a Professor of the Department of Industrial & Manufacturing Systems Engineering, State Key Laboratory of Digital Manufacturing Equipment & Technology, School of Mechanical Science & Engineering, HUST. He had published more than 80 refereed papers. His research interests include intelligent algorithm, big data and machine learning etc.

Bing Zeng received his Ph.D. degree in industrial engineering from Huazhong University of Science and Technology, China, 2017. He is director of big data center, XEMC Windpower Co. Ltd.,. His research interests include intelligent algorithm, big data, machine learning and fault diagnosis etc.