Energy 236 (2021) 121492

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

A novel asynchronous deep reinforcement learning model with adaptive early forecasting method and reward incentive mechanism for short-term load forecasting

Wenyu Zhang [a], Qian Chen [a], Jianyong Yan [b], Shuai Zhang [a, *], Jiyuan Xu [a]

[a] School of Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, Hangzhou, 310018, China
[b] Development and Planning Office, Zhejiang University of Finance and Economics, Hangzhou, 310018, China

ARTICLE INFO

Article history:
Received 3 January 2021
Received in revised form
8 May 2021
Accepted 12 July 2021
Available online 14 July 2021

Keywords:
Load forecasting
Deep reinforcement learning
Deep learning
Deep deterministic policy gradient

ABSTRACT

Accurate load forecasting is challenging due to the significant uncertainty of load demand. Deep reinforcement learning, which integrates the nonlinear fitting ability of deep learning with the decision-making ability of reinforcement learning, has obtained effective solutions to various optimization problems. However, no study has been reported, which used deep reinforcement learning for short-term load forecasting because of the difficulties in handling the high temporal correlation and high convergence instability. In this study, a novel asynchronous deep reinforcement learning model is proposed for short-term load forecasting by addressing the above difficulties. First, a new asynchronous deep deterministic policy gradient method is proposed to disrupt the temporal correlation of different samples to reduce the overestimation of the expected total discount reward of the agent. Further, a new adaptive early forecasting method is proposed to reduce the time cost of model training by adaptively judging the training situation of the agent. Moreover, a new reward incentive mechanism is proposed to stabilize the convergence of model training by taking into account the trend of agent actions at different time steps. The experimental results show that the proposed model achieves higher forecasting accuracy, less time cost, and more stable convergence compared with eleven baseline models.

1. Introduction

The load refers to the power consumption from power system plants [1]. The optimal balance between electricity generation and load demand must be maintained to avoid fatal interference to the grid. Therefore, load forecasting has played a critical role in power system operations, which reveals a cost-effective, efficient, and reliable technique within the energy management framework [2]. The new trend in the sustainable energy development and entrepreneurship is to utilize the energy that has satisfied the human needs, which requires the knowledge of the load forecasting to utilize the available energy by a smart way with smart decision [3].

Load forecasting helps to estimate future loads from recent loads using various techniques in efforts to save energy, reduce costs, perform power management, and implement economic dispatch plans [3]. According to California Renewable Energy Committee, most of the electricity is generated by wind energy, solar energy, and fire energy, which are important parts of energy market. In addition, load forecasting can provide a reliable indicator for managing the complex pricing strategies in liberalized and deregulated energy markets with higher financial benefits [4]. It has been reported that there is a growth of 10 million dollars operating costs annually associated with 1% raise in load forecasting error. Due to the importance of load forecasting, load forecasting has been a vibrant research topic over the past decade, and has attracted a lot of attention from people and companies in the field of energy [5].

Load forecasting can be divided into three types according to the time horizon. Long-term load forecasting predicts the load along a year or more than one year. Mid-term load forecasting predicts the load from one week to one year. Short-term load forecasting (STLF) predicts the load from 1 h to one week [6] and underlies the stable operation, safety assessment, and efficient dispatching of the power systems. STLF is also closely related to social economic

[*] Corresponding author.
E-mail addresses: wyzhang@e.ntu.edu.sg (W. Zhang), qianchen@zufe.edu.cn (Q. Chen), yjy@zufe.edu.cn (J. Yan), zhangshuai@zufe.edu.cn (S. Zhang), straka@zufe.edu.cn (J. Xu).

https://doi.org/10.1016/j.energy.2021.121492
0360-5442/© 2021 Published by Elsevier Ltd.

development after the deregulation of the power systems since market operators set day-ahead market prices based on the forecasted load [7]. However, accurate STLF is challenging because of the significant uncertainty of load demand.

Recently, deep learning models, such as artificial neural network (ANN), deep neural network [8], convolutional neural network (CNN) [9], and recurrent neural network (RNN), have made considerable contributions in many fields, including STLF, owing to its superior nonlinear fitting ability. In our previous work [10], long short-term memory (LSTM) neural network and CNN were combined to predict time series wind speed by extracting the temporal and spatial correlation features of wind.

Reinforcement learning is an artificial intelligence technology that seeks the optimal strategy and maximizes reward through continuous interaction with the environment [11]. It has been widely used in solving various optimization problems owing to its superior decision-making ability [12]. However, traditional reinforcement learning models such as Q-learning [13] are primarily used to solve optimization problems in low-dimensional and discrete state space. Hence, these models cannot solve the STLF problem, which is essentially an optimization problem in high-dimensional and continuous state space, because of the complexity for the agent to traverse the high-dimensional and continuous state space to find the optimal strategy with the maximum reward.

Deep reinforcement learning integrates the nonlinear fitting ability of deep learning with the decision-making ability of reinforcement learning, thereby obtaining effective solutions to complicated problems in high-dimensional and continuous state space [14]. The agents of deep reinforcement learning models seek the optimal actions by using deep neural network, as is different from the traditional reinforcement learning. However, the high temporal correlation of STLF results in the overestimation of the expected total discount reward of the agent, which severely reduces the forecasting accuracy. In addition, the high convergence instability of deep reinforcement learning models restricts their applications in STLF. Therefore, the powerful deep reinforcement learning has not been reported in the field of STLF up to now.

To bridge the above research gap, a novel asynchronous deep deterministic policy gradient model with adaptive early forecasting method and reward incentive mechanism (ADDPG-AEF-RIM) is proposed to address the aforementioned difficulties. The main contributions of this study are described as follows.

(1) A novel deep reinforcement learning model is proposed to solve the STLF problem for the first time. The proposed ADDPG-AEF-RIM model achieves accurate and robust forecasting results in multiple comparison experiments.
(2) A new asynchronous deep deterministic policy gradient (ADDPG) method is proposed to disrupt the temporal correlation of different samples by asynchronously selecting different critic networks, which are the unique neural networks of deep reinforcement learning models, so as to reduce the overestimation of the expected total discount reward of the agent.
(3) A new adaptive early forecasting (AEF) method is proposed in this study. It determines whether to bypass the subsequent training in each episode by adaptively judging the training situation of the agent, thereby significantly reducing the time cost of model training.
(4) A new reward incentive mechanism (RIM) is proposed to stabilize the convergence of model training by rewarding an incentive bonus to the agent according to the trend of agent actions at different time steps.
(5) Two real-world datasets, Independent System Operator-New England (ISO-NE) and Global Energy Forecasting Competition 2017 (GEFCom2017) datasets, four evaluation metrics, and eleven baseline models are used to verify the performance of ADDPG-AEF-RIM. The experimental results demonstrate that the proposed model achieves high forecasting accuracy, low time cost, and stable convergence.

The remainder of this paper is organized as follows. Section 2 provides a review of previous research on STLF and reinforcement learning. Section 3 presents the basic concept of deep reinforcement learning and the proposed ADDPG-AEF-RIM model. Section 4 describes the implementation details and experimental results of ADDPG-AEF-RIM on two datasets. Section 5 outlines conclusions and future works.

2. Related work

This section briefly summarizes previous research regarding the STLF problem and reinforcement learning models.

2.1. Short-term load forecasting

Many researchers have conducted a lot of studies to improve the accuracy of STLF. Generally, the models used for STLF can be divided into three categories: statistics-based models, machine learning-based models, and deep learning-based models [15].

Vaghefi et al. [16] proposed a statistics-based model that combined multiple linear regression model with a seasonal autoregressive moving average model to predict the electricity load. Xing et al. [17] proposed a hybrid forecasting framework to improve the load forecasting accuracy by using multi-variable quantile regression. Zhang et al. [18] conducted a comprehensive comparison among different discrete wavelet transformation and empirical mode decomposition techniques for building load forecasting. The advantage of statistics-based models is that the temporal features of the time series can be effectively utilized. However, the forecasting accuracy of statistics-based models is low owing to the nonlinear correlation of load data [19].

To address the shortcomings of statistics-based models, machine learning-based models for STLF have been used in numerous studies. For example, Jurado et al. [20] combined feature selection method, fuzzy inductive reasoning, random forest, and neural network to predict the hourly electricity load of buildings. Chen et al. [21] employed the support vector regression model and achieved higher stability and accuracy for predicting the load of four typical office buildings compared with seven other statistics-based models. Yang et al. [22] proposed a hybrid model that combined feature selection method and least squares support vector machines for predicting half-hour electricity load. Massaoudi et al. [23] proposed a machine learning-based model for STLF, by combining light gradient boosting machine, extreme gradient boosting machine, and multi-layer perceptron. However, with the increase of data volume along with the complexity of nonlinear correlation, overfitting often occurs in machine learning-based models, decreasing the forecasting accuracy [15].

Recently, it has been widely acknowledged that deep learning-based models can automatically extract important features and complex nonlinear correlations from large amounts of data so as to alleviate overfitting [24]. For example, Shi et al. [25] proposed a pooling-based RNN for household load forecasting and addressed overfitting by increasing data diversity and volume. Atef and Eltawil [26] demonstrated the effectiveness of deep-stacked bidirectional LSTM in predicting the electricity load. Huang et al. [27] proposed an improved CNN with universal load range discretization for probabilistic load forecasting and proved the superiority of the proposed method over seven baseline models. Li et al. [28] developed a convolutional LSTM neural network with selected autoregressive features to improve the forecasting accuracy of short-term household electricity load. However, deep learning-based models often require large time cost and computer memory for model training, which is not conducive to practical STLF application.

In this study, a novel asynchronous deep reinforcement learning model is proposed for solving the STLF problem while maintaining the excellent nonlinear fitting ability of deep learning and alleviating its drawback of computational complexity by leveraging the decision-making ability of reinforcement learning. The experimental results show that the proposed model achieves higher forecasting accuracy, less time cost, and more stable convergence compared with eleven baseline models including machine learning-based models, deep learning-based models, and traditional deep reinforcement learning models.

2.2. Reinforcement learning

Since DeepMind used reinforcement learning to play Atari games in 2013, reinforcement learning has been receiving increasing attention in both academia and industry [29]. Reinforcement learning models can obtain information and update model parameters by receiving feedback from the environment. The mainstream reinforcement learning models, including Q-learning [30], state-action-reward-state-action (SARSA) [31], and policy gradients [32], have achieved considerable success in the fields of optimization, decision-making, and scheduling. For example, Šemrov et al. [33] proposed a train rescheduling method based on Q-learning to solve a single-lane track scheduling problem with three trains within reasonable computational time.

Reinforcement learning models have also been used for STLF in some studies. For example, Feng et al. [30] developed a two-step STLF model based on Q-learning to achieve the dynamic selection of forecasting models for deterministic and probabilistic load forecasting. Ma et al. [34] proposed a dynamic integration approach based on Q-learning for load forecasting, in which reinforcement learning was employed to select the forecasting models adaptively. However, these studies only used reinforcement learning to determine the weights of different forecasting models in low-dimensional and discrete state space. They did not use reinforcement learning as a direct forecasting model to solve the STLF problem, which is essentially an optimization problem in high-dimensional and continuous state space.

In recent years, it has been acknowledged that deep reinforcement learning models, including deep Q-network [35], and deep deterministic policy gradient [36], perform well in the field of energy scheduling. For example, Wan et al. [37] formulated electric vehicle charging scheduling as a Markov decision process and proposed a model-free approach based on deep Q-network to optimize the charging scheduling. Mocanu et al. [38] demonstrated the effectiveness of deep Q-network and deep policy gradients for building energy optimization. Wei et al. [39] presented a recommender system, based on deep reinforcement learning, that reduces the energy consumption of commercial buildings. However, to the best of our knowledge, no study using deep reinforcement learning for STLF has been reported. This is primarily because of the difficulties in handling high temporal correlation and high convergence instability.

In this study, a novel deep reinforcement learning model with AEF method and RIM is proposed to address the above difficulties in the field of STLF, which inspires a new insight for future load forecasting. Through multiple comparison experiments, the proposed model was found to achieve the accurate and robust forecasting results.

3. Methodology

The basic concept of deep reinforcement learning is briefly described in sub-section 3.1. The proposed ADDPG-AEF-RIM model, comprising the ADDPG method, AEF method, and RIM, is elaborated in sub-section 3.2.

3.1. Basic concept of deep reinforcement learning

3.1.1. Reinforcement learning method

The reinforcement learning method is based on the Markov decision process (MDP) and is a process that allows agents to continuously interact with the environment through trial and error over discrete time steps [11]. MDPs are generally defined as $<\mathscr{S},\mathscr{A},\mathscr{P},\mathscr{R}>$ [38], where:

- $\mathscr{S}$ is the state space, $\forall s\in\mathscr{S}$,
- $\mathscr{A}$ is the action space, $\forall a\in\mathscr{A}$,
- $\mathscr{P}:\mathscr{S}\times\mathscr{A}\times\mathscr{S}\rightarrow\mathbb{R}+$ is the transition probability distribution, which is defined as the probability that the agent will move to state s' at time t+1 by executing action a in state s at time t, denoted as $p_a(s,s^{'})=p(s_{t+1}=s^{'}|s_t=s,a_t=a)$, and
- $\mathscr{R}:\mathscr{S}\times\mathscr{A}\times\mathscr{S}\rightarrow\mathbb{R}$ is the reward function, and r_t denotes the immediate reward that the agent receives from the environment after executing action a at time t, $\forall r\in\mathscr{R}$.

Based on these definitions, the basic process of reinforcement learning is described as follows: To complete a certain task, the agent needs to execute the action a_t according to the policy $\pi(a_t|s_t):\mathscr{S}\times\mathscr{A}\rightarrow\mathbb{R}+$ in the current state s_t and obtains the corresponding reward r_t. The ultimate goal of the agent in traditional reinforcement learning models is to seek the maximum return R_t, obtained by adopting the optimal policy π^*, which is the sum of discount rewards, as shown in Equation (1) [39]:

$$R_t=\sum\nolimits_{k=0}^{\infty}\gamma^k r_k \tag{1}$$

where k means different time steps, and γ is a discount factor, which indicates the tradeoffs between the importance of immediate rewards and future rewards [40].

In deep reinforcement learning models such as deep deterministic policy gradient (DDPG) models, the agent aims to maximize state value $V_\pi(s)$ or state-action value $Q_\pi(s,a)$. $V_\pi(s)$ is defined as the expected total discount reward that the agent obtains starting from state s under policy π, as shown in Equation (2) [40]:

$$V_\pi(s)=\mathbb{E}_\pi\left(\sum\nolimits_{k=0}^{\infty}\gamma^k r_{t+k}\middle|\ s_t=s\right) \tag{2}$$

$Q_\pi(s,a)$ is defined as the expected total discount reward that the agent obtains after executing action a starting from state s under policy π, which can be converted from $V_\pi(s)$, as shown in Equation (3) [40]:

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}_\pi\left(\sum\nolimits_{k=0}^{\infty}\gamma^k r_{t+k}\middle|\ s_t=s, a_t=a\right) \\ &= \mathbb{E}_\pi\left(r_t + \gamma\underbrace{\mathbb{E}_\pi[Q_\pi(s_{t+1},a_{t+1})]|s_t=s,a_t=a}_{V_\pi(s_{t+1})}\right) \end{aligned} \tag{3}$$

3.1.2. Deep deterministic policy gradient

The DDPG model integrates the deep learning method from artificial intelligence and the actor-critic method from reinforcement learning [41]. It includes an online actor network and an online critic network, which are unique neural networks of deep reinforcement learning models, denoted by $\mu(s|\theta^{\mu})$ and $Q(s,a|\theta^{Q})$, where μ and Q are the deterministic policies of the online actor and critic networks, respectively, and θ^{μ} and θ^{Q} are the weight parameters of the online actor and critic networks, respectively. The online actor network maps a state to an action at time t, and the online critic network calculates the state-action value by combining the action output from the online actor network at time t.

In addition, the target actor network and target critic network are established in DDPG, which are denoted by $\mu'(s|\theta^{\mu'})$ and $Q'(s,a|\theta^{Q'})$, respectively, where μ' and Q' are the deterministic policies of the target actor and critic networks, respectively, and $\theta^{\mu'}$ and $\theta^{Q'}$ are the weight parameters of the target actor and critic networks, respectively. The target actor network maps a state to an action at time t+1, and the target critic network calculates the state-action value by combining the action output from the target actor network at time t+1. However, the high temporal correlation of different samples results in the overestimation of state-action values, which reduces the forecasting accuracy.

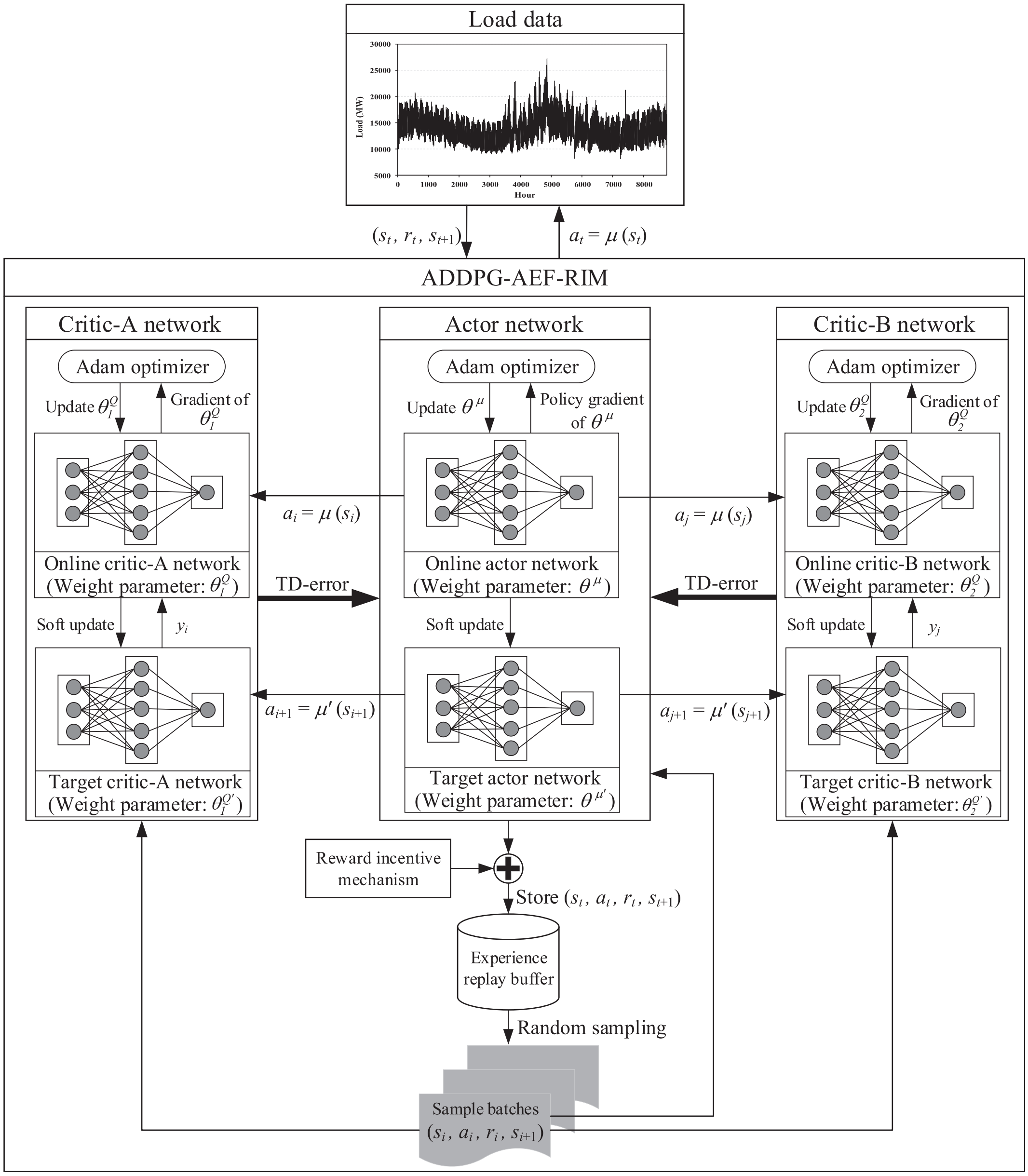
Load data
30000
25000
20000
15000
10000
5000
Load (MW)
0
1000
2000
3000
4000
5000
6000
7000
8000
Hour
(s_t, r_t, s_{t+1})
$a_t = \mu(s_t)$
ADDPG-AEF-RIM
Critic-A network
Actor network
Critic-B network
Adam optimizer
Adam optimizer
Adam optimizer
Update θ_1^Q
Gradient of θ_1^Q
Update θ^μ
Policy gradient of θ^μ
Update θ_2^Q
Gradient of θ_2^Q
$a_i = \mu(s_i)$
$a_j = \mu(s_j)$
Online critic-A network (Weight parameter: θ_1^Q)
Online actor network (Weight parameter: θ^μ)
Online critic-B network (Weight parameter: θ_2^Q)
TD-error
TD-error
Soft update
y_i
Soft update
Soft update
y_j
$a_{i+1} = \mu'(s_{i+1})$
$a_{j+1} = \mu'(s_{j+1})$
Target critic-A network (Weight parameter: $\theta_1^{Q'}$)
Target actor network (Weight parameter: $\theta^{\mu'}$)
Target critic-B network (Weight parameter: $\theta_2^{Q'}$)
Reward incentive mechanism
Store (s_t, a_t, r_t, s_{t+1})
Experience replay buffer
Random sampling
Sample batches (s_i, a_i, r_i, s_{i+1})

Fig. 1. Overview of ADDPG-AEF-RIM

3.2. Proposed model

The proposed ADDPG-AEF-RIM model is shown in Fig. 1 and detailed in the following sub-sections.

3.2.1. Asynchronous deep deterministic policy gradient

As shown in Fig. 1, the ADDPG contains one actor network and two critic networks (denoted by critic-A and critic-B). The two critic networks are used to disrupt the temporal correlation of different samples so as to reduce the overestimation of state-action values. Each actor and critic network are divided into one online and one target network. Therefore, the ADDPG has six neural networks in total. The ADDPG selects the critic-A or critic-B network at different time steps to calculate the Temporal-Difference error (TD-error) [42] for updating the weight parameters of the actor network. The pseudocode of the ADDPG is given in Algorithm 1.

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}\right) \tag{4}$$

$$L = \frac{1}{N}\sum_i^N \left(y_i - Q\left(s_i, a_i|\theta^Q\right)\right)^2 \tag{5}$$

where y_i is the output of target critic-A or critic-B network, $\theta^{\mu'}$ is the weight parameter of the target actor network, $\theta^{Q'}$ is the weight parameter of the target critic-A or critic-B network, and N represents the size of the transitions.

(4) The policy of the online actor network is updated through the sampled policy gradient, as shown in Equation (6) [43]:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i^N \nabla_a Q\left(s, a\middle|\theta^Q\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_i} \tag{6}$$

Algorithm 1: ADDPG

```
Initialize online actor network $\mu(s|\theta^\mu)$, online critic-A network $Q_1\left(s, a|\theta_1^Q\right)$, and online critic-B network $Q_2\left(s, a|\theta_2^Q\right)$ with weights $\theta^\mu$, $\theta_1^Q$, and $\theta_2^Q$
Initialize target actor network $\mu'(s|\theta^{\mu'})$, target critic-A network $Q_1'\left(s, a|\theta_1^{Q'}\right)$, and target critic-B network $Q_2'\left(s, a|\theta_2^{Q'}\right)$ with weights $\theta^{\mu'} \leftarrow \theta^\mu$, $\theta_1^{Q'} \leftarrow \theta_1^Q$, and $\theta_2^{Q'} \leftarrow \theta_2^Q$
Initialize experience replay buffer $B$ and asynchronization variable $V$
for $episode = 1, \ldots, M$ do
  Receive initial state $s_1$
  for $t = 1, \ldots, T$ do
    Select action $a_t = \mu(s_t|\theta^\mu)$ according to the current policy of the online actor network
    Execute action $a_t$, get reward $r_t$ and reach the next state $s_{t+1}$
    Store transition $(s_t, a_t, r_t, s_{t+1})$ in $B$
    Sample random mini-batch of $N$ transitions from $B$
    if $t \% V = 0$ then
      Obtain the output of target critic-A network using Equation (4)
      Update online critic-A network by minimizing the loss using Equation (5)
      Update the policy of the online actor network through the sampled policy gradient using Equation (6)
      Update the target actor and critic-A networks through soft update using Equations (7) and (8)
    else
      Obtain the output of target critic-B network using Equation (4)
      Update online critic-B network by minimizing the loss using Equation (5)
      Update the policy of the online actor network through the sampled policy gradient using Equation (6)
      Update the target actor and critic-B networks through soft update using Equations (7) and (8)
    end if
  end for
end for
```

Fig. 1 and Algorithm 1 show the main process of ADDPG, which can be described as follows:

(1) At each time step t, the agent executes action a_t according to the current policy of the online actor network, receives reward r_t, and reaches next state s_{t+1}.
(2) Transition (s_t, a_t, r_t, s_{t+1}) is stored in the experience replay buffer B, and a random mini-batch (s_i, a_i, r_i, s_{i+1}) is drawn from B, where i represents different time steps.
(3) After the target actor network output the next action a_{t+1} in state s_{t+1}, the online critic-A or critic-B network is updated by minimizing loss L, as shown in Equations (4) and (5) [43]. The decision to use the critic-A or critic-B network depends on time step t and asynchronization variable V, as shown in Algorithm 1. If asynchronization variable V is divisible by the value of time step t, ADDPG selects critic-A network; otherwise critic-B network is selected. The value of asynchronization variable V is obtained by using grid search [44].

where θ^μ is the weight parameter of the online actor network, and θ^Q is the weight parameter of the online critic-A or critic-B network.

(5) The target networks are soft updated with update speed $\tau \in (0, 1)$, as shown in Equations (7) and (8) [43]:

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

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \tag{8}$$

ADDPG can reduce the overestimation of state-action values by selecting the critic-A or critic-B network asynchronously, which effectively addresses the difficulty of high temporal correlation presented in STLF. The performance evaluation of ADDPG is discussed in Section 4.

3.2.2. Adaptive early forecasting method

With excessive training of deep reinforcement learning model, the agent may explore the non-beneficial or less beneficial actions, which may destruct the promising action trajectory and decrease the forecasting accuracy of the model. To shorten the training time cost of the deep reinforcement learning model and avoid the reduction of forecasting accuracy, an AEF method is proposed.

In the proposed model, there are T time steps in each episode. As described in the previous sub-section, the agent executes an action in response to the current environment in each time step and obtains a reward for this action. As the time step proceeds, the rewards, which are set as negative numbers in this study, will be accumulated as the cumulative reward of this episode, as shown in Equation (9):

$$ep\_r = \sum_i^T r_i \tag{9}$$

When the cumulative reward in a certain time step is smaller than the reward threshold, which is obtained using grid search, the agent in this episode is considered to be trained well. Then, the subsequent time steps in this episode will be bypassed adaptively, followed by the next episode training directly. By determining whether the subsequent time steps are worth training according to the cumulative reward and reward threshold in each episode, the agent can both reduce the training time cost and avoid the reduction of forecasting accuracy. The performance evaluation of the AEF method will be discussed in Section 4.

3.2.3. Reward incentive mechanism

The reward in deep reinforcement learning is important as it determines how the agent modifies the next action based on the previous action. Herein, the action is set as the forecasted value, and reward is set as the difference between the action and the actual value. The RIM is proposed to stabilize the convergence of the model training by taking into account the trend of actions at different time steps.

As shown in Fig. 2, at time steps t-1 and t, the agent executes actions $a_{t\text{-}1}$ and a_t according to the current state, respectively. The difference between the action and actual load at different time steps were denoted by $\Delta_{t\text{-}1}$ and Δ_t, respectively. When Δ_t is less than $\Delta_{t\text{-}1}$, indicating that the forecasted value is closer to the actual load over time, the trend of action is considered to be positive, and this trend is worth being stimulated. Therefore, the reward at time step t will be enhanced with a positive number as the additional incentive bonus, which is a random positive number no bigger than Δ_t obtained using grid search.

In Fig. 2, slopes are used to represent the trends of the actions. After using the RIM, the slope k_{rim} of the line formed by the rewards with RIM at time steps t-1 and t (represented by the blue two-dash line) becomes steeper than the slope k of the line formed by the rewards without RIM at time steps t-1 and t (represented by the green long-dash line). The change in the slopes of the lines formed by the rewards improves the convergence stability of the training. The performance evaluation of RIM will be discussed in Section 4.

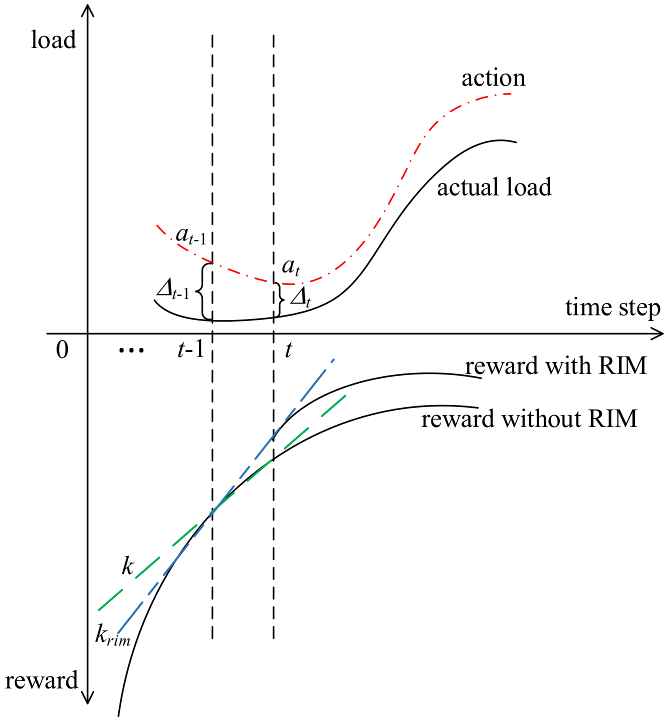

Fig. 2. Schematic diagram of RIM.

4. Experiments and analysis

To verify the forecasting accuracy, time cost, and convergence stability of the proposed ADDPG-AEF-RIM model, two real-world datasets, four evaluation metrics, and eleven baseline models, including two recent models proposed by Wen et al. [45] and Bendaoud and Farah [46]; were used to construct different comparison experiments. The hardware and software platforms used in the experiments are described in Table 1.

The parameters of ADDPG-AEF-RIM remained constant in all the comparison experiments on the different datasets. For ADDPG-AEF-RIM and the baseline models, single-step ahead and 24-step ahead forecasting with 24 sequence length (time steps) of input was performed. The main parameters of ADDPG-AEF-RIM and the baseline models are listed in Table 2. Most of the parameters were obtained using grid search, except the parameters for the models proposed by Wen et al. [45] and Bendaoud and Farah [46]; which were adopted from the references directly. Except for the relevant parameters of the critic-B networks and reward threshold, the parameters of DDPG were the same as those of ADDPG-AEF-RIM to ensure the fair comparison. The parameters have a great influence on the model performance [47]. However, the parameters of all models remain consistent in the experiments on all datasets to demonstrate the good robustness of the proposed model.

4.1. Data description

In this study, two widely used real-world datasets were employed to verify the performance of the models. No additional features were used as the input to the model, except for the electrical load data. The missing data were filled with the average values at the previous hour and the next hour, and no other data processing techniques were used in this study.

Table 1
Hardware and software platforms.

Hardware and software platform	Configuration
Operating system	Windows 10
RAM	16 GB
CPU	Intel Core i7-8700 K
GPU	GeForce RTX 2080
Programing language	Python
Deep learning framework	TensorFlow
Programing software	Spyder

Table 2
Experimental parameters of different models.

Model	Experimental parameters	Value
ADDPG-AEF-RIM	Batch size	64
	Learning rate of actor networks	0.0026
	Learning rate of critic-A and critic-B networks	0.004
	Number of hidden layer neurons in actor networks	140
	Number of hidden layer neurons in critic-A networks	140
	Number of hidden layer neurons in critic-B networks	300
	Reward discount factor	0.9
	Update speed of target networks	0.1
	Size of experience replay buffer	1000
	Number of episodes	100
	Number of steps in each episode	2000
	Asynchronization variable	50
	Reward threshold	−1.4
DDPG	Batch size	64
	Learning rate of actor networks	0.0026
	Learning rate of critic networks	0.004
	Number of hidden layer neurons in actor networks	140
	Number of hidden layer neurons in critic networks	140
	Reward discount factor	0.9
	Update speed of target networks	0.1
	Size of experience replay buffer	1000
	Number of episodes	100
	Number of steps in each episode	2000
Adaboost	Base estimator	Decision tree
	Loss function	linear
	Number of iterations	100
	Learning rate	0.1
K-nearest neighbor (KNN)	Number of neighbors	5
	Number of jobs	1
	Size of leaf	30
Decision tree (DT)	Evaluation function	Mean squared error
	Maximum depth of the tree	7
Gradient boosting decision tree (GBDT)	Loss function	Least squares
	Number of iterations	100
	Learning rate	0.1
	Maximum depth of the tree	3
Support vector regression (SVR)	Kernel function	Linear
	Regularization parameter	1
ANN/CNN/RNN	Learning rate	0.004
	Number of epochs	100
	Batch size	256
	Activation function	Relu

The first dataset is the ISO-NE dataset[1], which contains 103,776 hourly electrical load records from March 1, 2003, to December 31, 2014, of New England. The second dataset is the GEFCom2017 dataset[2], which records the load data of an unknown area in the United States. The delivery point meter of the GEFCom2017 dataset used in this study records 61,344 hourly electrical load from January 1, 2005, to December 31, 2011. The unit of the load data is megawatt (MW). All datasets are divided into training sets and test sets. The training sets are used as the input to train the proposed model, and the trained model makes predictions on the test sets.

4.2. Evaluation metric

To compare the forecasting accuracy of different models, four evaluation metrics are used in this study: mean absolute percentage error (MAPE), mean absolute error (MAE), R^2 score (R^2), and root mean squared error (RMSE). Using these metrics to evaluate the comprehensive performance of the models is more convincing because different metrics have been taken into account comprehensively. Their calculations are shown in Equations 10–13 [48]:

$$MAPE = \frac{1}{T} \times \left(\sum\nolimits_{i=1}^{T} \left| \frac{y_i - p_i}{y_i} \right| \right) \times 100\% \tag{10}$$

$$MAE = \frac{1}{T} \times \left(\sum\nolimits_{i=1}^{T} |y_i - p_i| \right) \tag{11}$$

$$R^2 = 1 - \left[\sum\nolimits_{i=1}^{T} (y_i - p_i)^2 \Big/ \sum\nolimits_{i=1}^{T} (y_i - \overline{y_i})^2 \right] \tag{12}$$

$$RMSE = \sqrt{\frac{1}{T} \times \sum\nolimits_{i=1}^{T} (y_i - p_i)^2} \tag{13}$$

where T indicates the number of forecasting points, y_i and p_i are the actual and forecasted values at time step i, respectively, and $\overline{yi}$ refers to the average value of y_i. Among them, the forecasting result would be more accurate with the smaller values of MAPE, MAE, and RMSE, or the larger values of R^2.

4.3. Performance on ISO-NE dataset

This sub-section details the comparison among ADDPG-AEF-

[1] Available at https://www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/zone-info.

[2] Available at https://www.dropbox.com/s/no9r15m9nschl63/Data.zip?dl=0.

RIM, DDPG, and different baseline models in forecasting accuracy, time cost, and convergence stability. To verify the robustness of ADDPG-AEF-RIM, the ISO-NE dataset was divided into two cases, each of which included different training sets and test sets. Multiple comparative experiments were conducted on each case.

The test set of the first case is the load data of the twelve months in 2006, and the corresponding training set is the load data from January 1, 2003, to December 31, 2005. The test set of the second case is the load data from January 1, 2011, to December 31, 2011, and the corresponding training set is the load data from January 1, 2004, to December 31, 2010. The forecasting results of ADDPG-AEF-RIM and DDPG for both cases are listed in Tables 3 and 4, respectively. All results were obtained by averaging the forecasting results of the models through five runs.

As shown in Tables 3 and 4, all evaluation metrics and the time cost of ADDPG-AEF-RIM are superior to those of DDPG. It is worth noting that, for the first case, the average time cost required for ADDPG-AEF-RIM is only 45.2% of that required for DDPG. For ADDPG-AEF-RIM, the average twelve-month MAPE, MAE, and RMSE are decreased by 9.7%, 9.6%, and 9.7%, respectively, compared to DDPG. R^2 for ADDPG-AEF-RIM is increased by 0.3% compared to that for DDPG. Similarly, in the second case, ADDPG-AEF-RIM consumes only 38.0% of the time cost of DDPG. Further, MAPE, MAE, and RMSE for ADDPG-AEF-RIM are decreased by 5.7%, 3.8%, and 4.5%, respectively, and R^2 is increased by 0.2% compared to that for DDPG. The results in Tables 3 and 4 show that the ADDPG, AEF, and RIM methods are effective in improving forecasting accuracy and reducing time cost. The comparative experiments on these two cases also verify the good robustness of ADDPG-AEF-RIM.

The forecasting results of year 2011 on the ISO-NE dataset among different models are listed in Table 5. Many machine learning methods, deep learning methods, and ensemble learning methods can be used for forecasting [49]. Therefore, the baseline models used in this study include Adaboost [50], KNN, ANN, DT [51], GBDT [52], SVR [53], CNN, RNN, DDPG, and the models proposed by Bendaoud and Farah [46] and Wen et al. [45]. These models cover the methods of machine learning, deep learning, ensemble learning, and deep reinforcement learning. All results were obtained by averaging the forecasting results of the models through five runs, and the significant values are boldfaced.

Table 5
Comparison of the forecasting results of year 2011 on the ISO-NE dataset among different models.

Metric	MAPE(%)	MAE	RMSE	R^2
Adaboost	3.741	542.680	680.496	0.896
KNN	2.685	398.594	511.831	0.941
ANN	1.649	236.938	**310.809**	0.978
DT	1.990	299.432	400.030	0.964
GBDT	1.671	250.849	337.598	0.974
SVR	3.032	445.396	554.341	0.931
CNN	1.495	230.051	333.812	0.975
RNN	2.550	391.001	452.602	0.954
DDPG	1.451	204.092	349.781	0.984
Bendaoud and Farah [46]	1.460	205.147	319.653	0.987
Wen et al. [45]	1.405	197.056	344.396	0.985
ADDPG-AEF-RIM	**1.368**	**196.314**	335.534	**0.988**

Note: Significant values are boldfaced.

As shown in Table 5, the MAPE, MAE, and R^2 of ADDPG-AEF-RIM are the best among the twelve models, and the RMSE of ADDPG-AEF-RIM is slightly worse than that of ANN. The proposed model can be considered to have the best performance because it performs the best on three of the four evaluation metrics.

To compare the forecasting results of ADDPG-AEF-RIM and DDPG more intuitively, Fig. 3 shows the forecasting error heat maps of ADDPG-AEF-RIM and DDPG for forecasting the load from March 1, 2006, to March 4, 2006, on the ISO-NE dataset. The experimental data are derived from the experiments in Table 3.

The different color blocks in Fig. 3 represent the different forecasting error of ADDPG-AEF-RIM and DDPG at different hours, respectively. The lighter the color, the smaller the forecasting error. It is evident that most of the forecasting error made by ADDPG-AEF-RIM are smaller than those of DDPG.

Fig. 4 shows the cumulative reward convergence of the DDPG with and without RIM. The test set is the load data from March 1, 2006, to March 31, 2006, on the ISO-NE dataset. The experimental data are derived from the experiments in Table 3.

It is clear that the fluctuation of the cumulative reward convergence of the DDPG with RIM is smaller than that of the DDPG

Table 3
Comparison of the forecasting results in different months of year 2006 on the ISO-NE dataset between ADDPG-AEF-RIM and DDPG.

Model	ADDPG-AEF-RIM					DDPG				
Metric	MAPE(%)	MAE	RMSE	R^2	Time(s)	MAPE(%)	MAE	RMSE	R^2	Time(s)
Jan	1.442	220.078	309.805	0.983	348.370	1.482	228.256	325.672	0.981	740.109
Feb	1.352	208.888	294.014	0.981	348.623	1.599	244.556	318.535	0.978	747.500
Mar	1.269	187.465	277.790	0.983	361.404	1.552	227.398	295.564	0.981	720.453
Apr	1.272	164.987	235.887	0.987	339.452	1.305	168.714	239.833	0.986	727.726
May	1.214	161.180	229.357	0.991	376.544	1.423	187.499	255.707	0.988	710.999
Jun	1.305	199.118	265.746	0.993	409.163	1.385	211.307	282.584	0.992	724.737
Jul	1.339	229.778	292.541	0.994	320.600	1.418	243.601	307.134	0.993	695.951
Aug	1.338	216.799	284.217	0.994	308.008	1.429	244.591	424.280	0.984	746.909
Sep	1.287	178.748	254.100	0.990	312.005	1.403	189.715	260.595	0.989	752.621
Oct	1.362	189.275	284.660	0.985	269.708	1.628	223.903	312.720	0.982	771.643
Nov	1.313	186.157	275.962	0.986	282.648	1.456	205.332	290.065	0.985	783.620
Dec	1.438	215.662	298.526	0.985	284.975	1.564	234.390	311.609	0.984	650.722
Ave	1.328	196.511	275.217	0.988	330.125	1.470	217.438	302.025	0.985	731.082

Table 4
Comparison of the forecasting results of year 2011 on the ISO-NE dataset between ADDPG-AEF-RIM and DDPG.

Model	ADDPG-AEF-RIM					DDPG				
Metric	MAPE(%)	MAE	RMSE	R^2	Time(s)	MAPE(%)	MAE	RMSE	R^2	Time(s)
2011	1.368	196.314	335.534	0.986	289.358	1.451	204.092	349.781	0.984	762.105

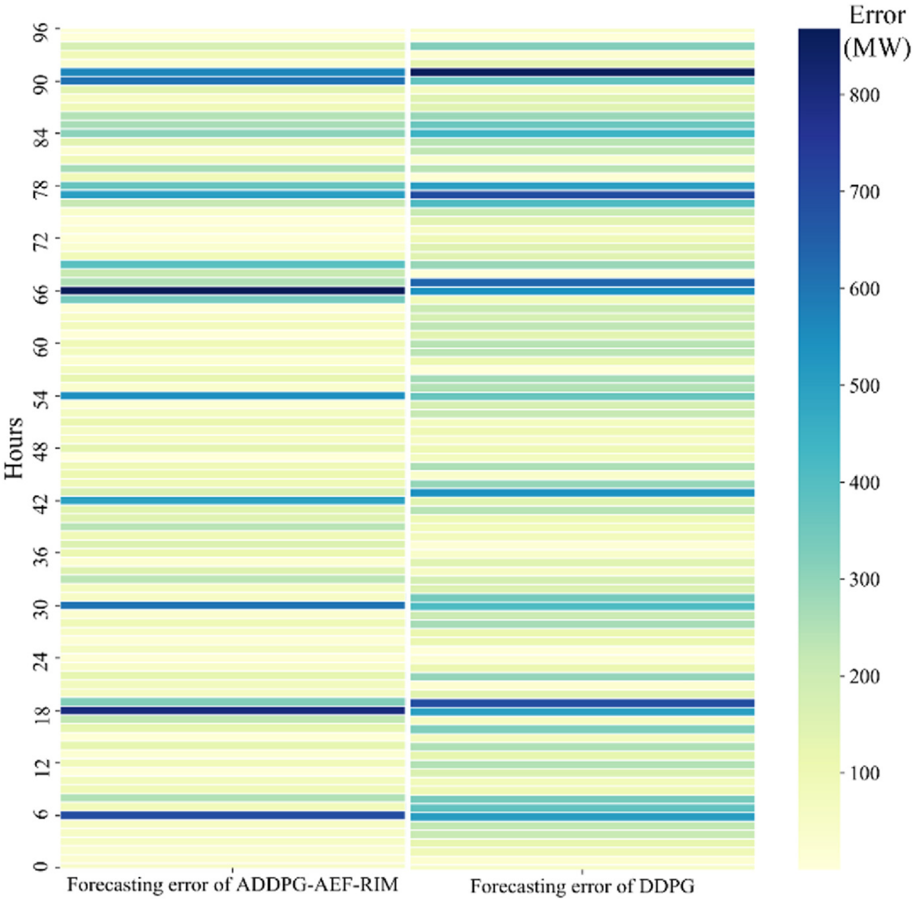

Fig. 3. Forecasting error heat maps of ADDPG-AEF-RIM and DDPG on the ISO-NE dataset.

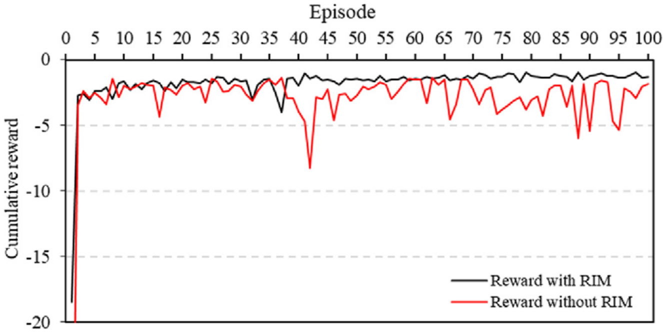

Fig. 4. Cumulative reward convergence of the DDPG with and without RIM on the ISO-NE dataset.

without RIM, and the final cumulative reward value of DDPG with RIM is higher, as shown in Fig. 4. The result verified that RIM could effectively stabilize the convergence of model training.

Figs. 5–8 show the forecasting results and error of ADDPG-AEF-RIM for January, April, July, and October on the ISO-NE dataset, respectively. It represents forecasts in the four seasons of spring, summer, autumn, and winter.

The 24-step ahead forecasting results of year 2011 on the ISO-NE dataset are listed in Table 6. All results were obtained by averaging the forecasting results of the models through five runs, and the significant values are boldfaced.

As shown in Table 6, the MAPE and MAE of ADDPG-AEF-RIM are decreased by 6.83% and 1.95% respectively, compared to those of CNN. The RMSE of ADDPG-AEF-RIM is increased by 1.56% compared

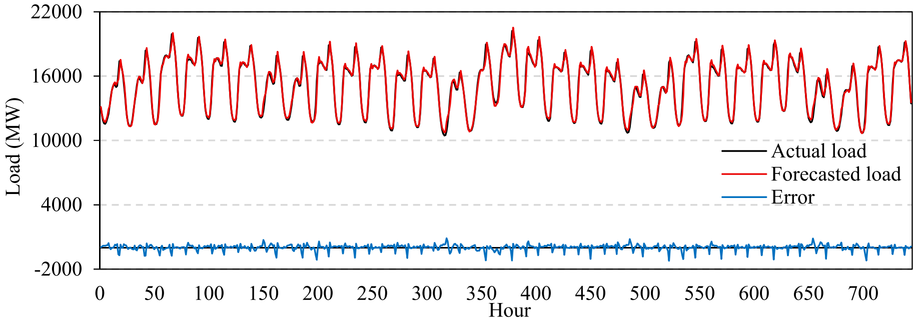

Fig. 5. Forecasting results and error of ADDPG-AEF-RIM in January on the ISO-NE dataset.

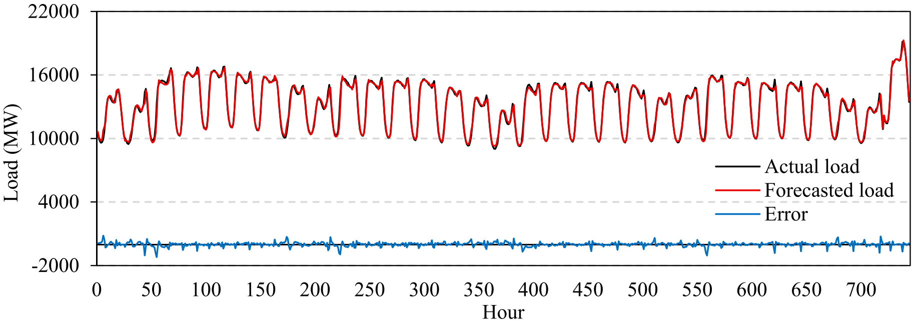

Fig. 6. Forecasting results and error of ADDPG-AEF-RIM in April on the ISO-NE dataset.

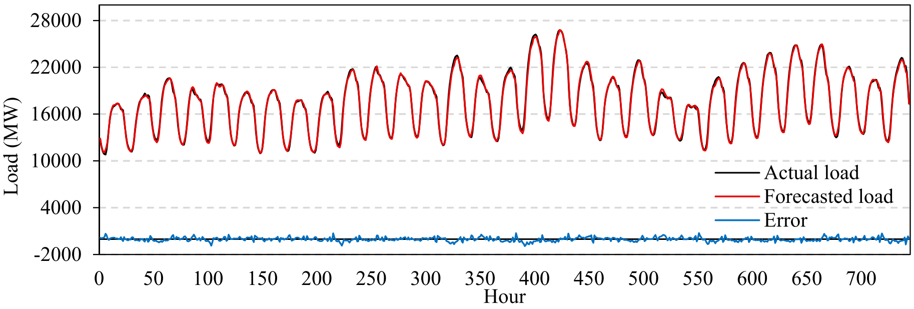

Fig. 7. Forecasting results and error of ADDPG-AEF-RIM in July on the ISO-NE dataset.

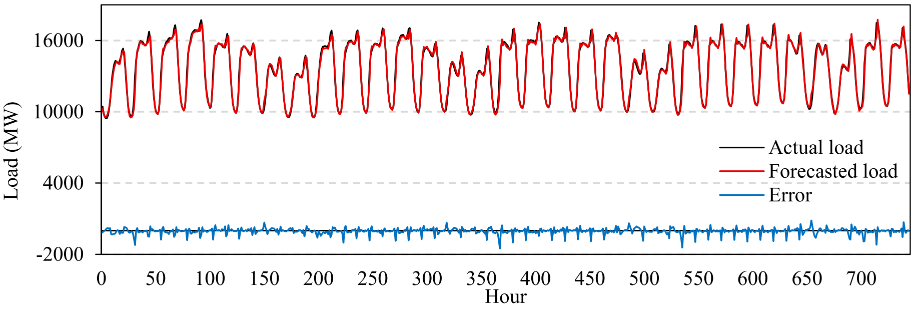

Fig. 8. Forecasting results and error of ADDPG-AEF-RIM in October on the ISO-NE dataset.

to that of CNN, and the R^2 is decreased by 0.73% compared to that of CNN. It is shown that the proposed model has the best performance in multi-step forecasting among twelve models on the ISO-NE dataset.

4.4. Performance on GEFCom2017 dataset

This sub-section describes the performance of ADDPG-AEF-RIM compared to eleven baseline models on the GEFCom2017 dataset.

The test set is the load data from December 1, 2011, to December 31, 2011, and the training set is the load data from January 1, 2005, to November 30, 2011. Table 7 and Fig. 9 display the forecasting results and scatter diagrams of the different models, respectively. All results were obtained by averaging the forecasting results of the models through five runs, and the significant values are boldfaced.

As shown in Table 7, the MAPE and MAE of ADDPG-AEF-RIM are the lowest among the twelve models, R^2 of ADDPG-AEF-RIM is slightly worse than that of the model proposed by Wen et al. [45]; and the RMSE of ADDPG-AEF-RIM is worse than that of CNN. The MAPE and MAE of ADDPG-AEF-RIM are decreased by 4.3% and 0.8%, respectively, compared to those of the model proposed by Wen et al. [45]. The R^2 of ADDPG-AEF-RIM is decreased by 0.7% compared to that of the model proposed by Wen et al. [45] because the latter's forecasting results have smaller fluctuations. However, most forecasting results of the model proposed by Wen et al. [45] tend to be larger than the actual loads. In summary, the proposed model outperforms the baseline models because it performs the best in terms of the MAPE and MAE on four evaluation metrics.

The black lines in Fig. 9 represent the ±20% error lines, respectively, which mean that the forecasting result is 20% larger or 20% smaller than the actual load. The results in Fig. 9 show that ADDPG-AEF-RIM obtains the best performance, with only few forecasting results outside of the black lines. The forecasting results of the model proposed by Wen et al. [45] are more concentrated than the other baseline models, but its most forecasting results tend to be larger than the actual loads. In addition, when the load values are relatively small, many forecasting results of the model proposed by Wen et al. [45] exceed the +20% error line.

Tables 8 and 9 display the forecasting results under peak load and valley load of different models, respectively. All results were obtained by averaging the forecasting results of the models through five runs, and the significant values are boldfaced.

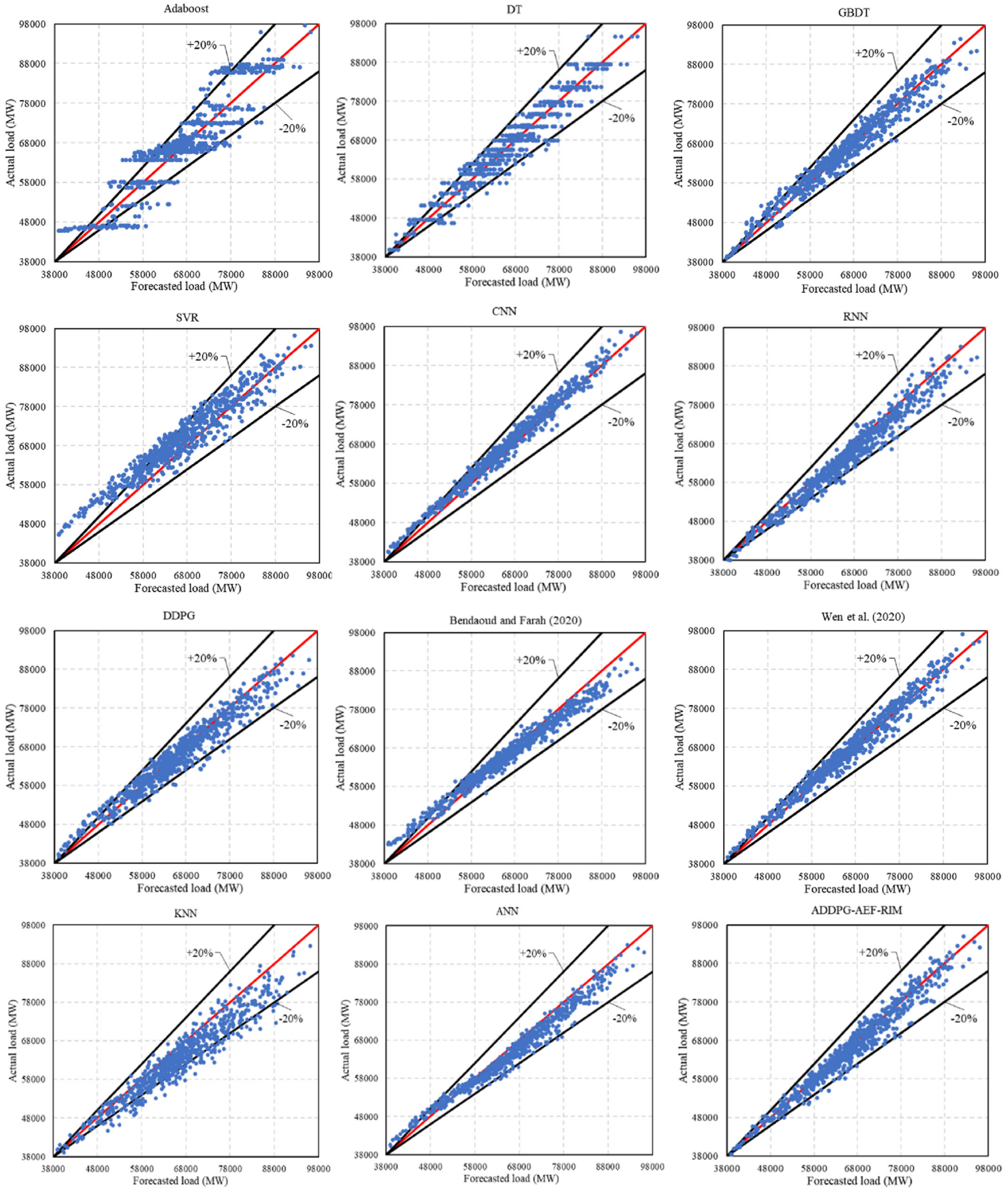

Fig. 9. Scatter diagrams of forecasting results on the GEFCom2017 dataset among different models.

Table 6
Comparison of the 24-step ahead forecasting results of year 2011 on the ISO-NE dataset among different models.

Metric	MAPE(%)	MAE	RMSE	R^2
Adaboost	8.051	1105.163	1408.926	0.749
KNN	7.820	1202.711	1678.439	0.644
ANN	6.295	886.162	1205.385	0.816
DT	6.088	879.437	1241.619	0.805
GBDT	6.007	864.277	1206.640	0.816
SVR	7.092	983.670	1279.395	0.793
CNN	5.955	829.203	**1148.586**	**0.833**
RNN	7.634	1078.731	1413.834	0.747
DDPG	6.497	951.946	1308.142	0.783
Bendaoud and Farah [46]	6.174	874.627	1210.243	0.815
Wen et al. [45]	6.027	887.316	1250.680	0.802
ADDPG-AEF-RIM	**5.548**	**812.995**	1166.787	0.827

Note: Significant values are boldfaced.

Table 7
Comparison of the forecasting results on the GEFCom2017 dataset among different models.

Metric	MAPE(%)	MAE	RMSE	R^2
Adaboost	5.506	3594.981	4485.732	0.824
KNN	5.088	3497.613	4409.118	0.830
ANN	3.013	2029.292	2528.202	0.944
DT	3.373	2210.422	2848.591	0.929
GBDT	2.692	1759.620	2227.918	0.957
SVR	5.989	3690.391	4225.015	0.844
CNN	2.658	1695.446	**1935.776**	0.966
RNN	3.069	2068.365	2652.599	0.938
DDPG	3.060	2022.692	2656.843	0.938
Bendaoud and Farah [46]	2.648	1728.963	2208.887	0.957
Wen et al. [45]	2.437	1597.127	1940.771	**0.967**
ADDPG-AEF-RIM	**2.332**	**1583.773**	2138.569	0.960

Note: Significant values are boldfaced.

As shown in Tables 8 and 9, CNN and ADDPG-AEF-RIM obtain the best performance among the twelve models under peak load and valley load, respectively. Although the forecasting results of the proposed model under peak value are not the best, it still can be considered that the proposed model has the most stable performance among the twelve models through comprehensive evaluation in Tables 7–9.

The 24-step ahead forecasting results on the GEFCom2017 dataset are listed in Table 10. All results were obtained by averaging the forecasting results of the models through five runs, and the significant values are boldfaced. The test set in this experiment is the load data from July 1, 2010, to July 31, 2010, and the training set is the load data from January 1, 2005, to June 30, 2010.

Table 8
Comparison of the forecasting results under peak load on the GEFCom2017 dataset among different models.

Metric	MAPE(%)	MAE	RMSE	R^2
Adaboost	5.830	4747.889	5575.868	0.000
KNN	6.826	5688.873	6522.285	0.000
ANN	3.440	2854.722	3397.764	0.354
DT	3.119	2577.224	3323.053	0.382
GBDT	2.857	2374.097	2979.687	0.503
SVR	3.119	2579.750	3137.056	0.449
CNN	**1.887**	**1565.617**	**1979.694**	**0.781**
RNN	3.407	2839.611	3465.908	0.328
DDPG	3.493	2916.888	3743.840	0.216
Bendaoud and Farah [46]	3.857	3234.794	3630.741	0.262
Wen et al. [45]	2.264	1878.961	2350.372	0.691
ADDPG-AEF-RIM	2.688	2229.404	2838.713	0.549

Note: Significant values are boldfaced.

Table 9
Comparison of the forecasting results under valley load on the GEFCom2017 dataset among different models.

Metric	MAPE(%)	MAE	RMSE	R^2
Adaboost	7.370	3814.125	4695.832	0.214
KNN	3.871	2058.715	2732.277	0.734
ANN	2.334	1144.979	1382.104	0.932
DT	4.228	2178.597	2722.666	0.736
GBDT	3.707	1904.363	2308.934	0.810
SVR	10.439	5233.067	5541.542	0.000
CNN	3.717	1886.036	2065.000	0.848
RNN	2.806	1479.567	1836.255	0.880
DDPG	3.568	1785.150	2114.856	0.841
Bendaoud and Farah [46]	4.296	2135.442	2419.688	0.791
Wen et al. [45]	2.881	1463.848	1731.163	0.893
ADDPG-AEF-RIM	**1.888**	**999.143**	**1324.931**	**0.937**

Note: Significant values are boldfaced.

Table 10
Comparison of the 24-step ahead forecasting results on the GEFCom2017 dataset among different models.

Metric	MAPE(%)	MAE	RMSE	R^2
Adaboost	7.797	4568.266	5767.189	0.764
KNN	8.687	5579.172	6839.358	0.668
ANN	6.080	3688.319	4472.543	0.858
DT	6.116	3666.677	4496.768	0.857
GBDT	6.243	3684.786	4459.167	0.859
SVR	16.360	8579.792	10264.652	0.253
CNN	6.560	3754.619	4708.577	0.843
RNN	6.248	3969.036	4813.925	0.836
DDPG	6.897	4020.860	5317.786	0.798
Bendaoud and Farah [46]	6.003	3682.323	4621.465	0.846
Wen et al. [45]	6.174	**3641.012**	4696.081	0.844
ADDPG-AEF-RIM	**5.891**	3667.399	**4396.155**	**0.862**

Note: Significant values are boldfaced.

As shown in Table 10, the MAPE, RMSE and R^2 of ADDPG-AEF-RIM are the best among twelve models, the MAE of ADDPG-AEF-RIM is slightly worse than that of the model proposed by Wen et al. [45]. It is shown that the proposed model has the best performance in multi-step forecasting compared with eleven baseline models on the GEFCom2017 dataset.

5. Conclusions and future work

Accurate load forecasting has great significance for energy markets because (a) the forecasted load is considered as the basis in determining the electricity price for consumers, (b) the forecasted load can make the power system more stable and reduce costs, and (c) the forecasted load can reduce the unnecessary consumption of the energy [3]. Conversely, inaccurate forecasted load can result in economic losses and serious power outages.

In this study, a novel asynchronous deep reinforcement learning model with adaptive early forecasting method and reward incentive mechanism is proposed for short-term load forecasting. Two real-world datasets, eleven baseline models, and four evaluation metrics are used to verify the performance of the proposed model. Based on the results of multiple comparison experiments, the following four conclusions are summarized as follows.

(1) The proposed model has achieved higher forecasting accuracy, less training time cost, and more stable convergence than the baseline models. The results of different comparison experiments on different datasets demonstrate the good

performance and robustness of the proposed model, which can make the power system more stable and cost effective.

(2) The proposed asynchronous deep deterministic policy gradient method can reduce the overestimation of state-action values of the agent by disrupting the temporal correlation of different samples so as to improve the forecasting accuracy of the deep reinforcement learning model. The proposed model also inspires a new insight of reinforcement learning models for future load forecasting.

(3) The proposed adaptive early forecasting method can reduce the time cost of model training by adaptively judging the current training situation of the agent and determining whether to bypass the subsequent training in each episode, which could help energy companies save a lot of time cost.

(4) The proposed reward incentive mechanism can stabilize the convergence of the model training by rewarding an incentive bonus to agent actions, which are worth being stimulated at different time steps.

Although the proposed model has high forecasting accuracy, low time cost, and stable convergence, it has some limitations. First, the abnormal data interfered with the deep reinforcement learning models, thereby reducing the forecasting accuracy. In future work, a more effective algorithm will be designed to alleviate the adverse impact of abnormal data for deep reinforcement learning models. Second, the less efficient grid search was used to obtain the parameters of the proposed model; in future, this can be enhanced by using evolutionary optimization algorithms such as the particle swarm optimization [54] algorithm. Finally, the algorithm can be further optimized to improve the performance of the proposed model for multi-step forecasting in the future.

Credit author statement

Wenyu Zhang: Conceptualization, Methodology, Writing-Original draft, Writing- Reviewing & Editing, Supervision, Funding Acquisition. **Qian Chen**: Conceptualization, Methodology, Formal analysis, Writing- Original draft, Data curation, Software, Validation. **Jianyong Yan:** Supervision, Writing- Reviewing & Editing. **Shuai Zhang**: Supervision, Writing- Reviewing & Editing, Funding Acquisition. **Jiyuan Xu**: Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work has been supported by National Natural Science Foundation of China (No. 51875503, No. 51975512), Zhejiang Natural Science Foundation of China (No. LZ20E050001), Zhejiang Key R & D Project of China (No.2021C03153).

References

[1] Singh P, Dwivedi P, Kant V. A hybrid method based on neural network and improved environmental adaptation method using Controlled Gaussian Mutation with real parameter for short-term load forecasting. Energy 2019;174:460–77.

[2] Yildiz B, Bilbao JI, Sproul AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renew Sustain Energy Rev 2017;73:1104–22.

[3] Talaat M, Farahat MA, Mansour N, Hatata AY. Load forecasting based on grasshopper optimization and a multilayer feed-forward neural network using regressive approach. Energy 2020;196:117087.

[4] Van der Meer DW, Widén J, Munkhammar J. Review on probabilistic forecasting of photovoltaic power production and electricity consumption. Renew Sustain Energy Rev 2018;81:1484–512.

[5] Zhang N, Li ZY, Zou X, Quiring SM. Comparison of three short-term load forecast models in Southern California. Energy 2019;189:116358.

[6] Mocanu E, Nguyen PH, Gibescu M, Kling WL. Deep learning for estimating building energy consumption. Sustain. Energy Grids Netw. 2016;6:91–9.

[7] Chen Y, Luh PB, Guan C, Zhao YG, Michel LD, Coolbeth MA, et al. Short-term load forecasting: similar day-based wavelet neural networks. IEEE Trans Power Syst 2009;25(1):322–30.

[8] Ryu S, Noh J, Kim H. Deep neural network based demand side short term load forecasting. Energies 2017;10(1):3.

[9] Ferreira A, Giraldi G. Convolutional neural network approaches to granite tiles classification. Expert Syst Appl 2017;84:1–11.

[10] Chen Y, Zhang S, Zhang WY, Peng JJ, Cai YS. Multifactor spatio-temporal correlation model based on a combination of convolutional neural network and long short-term memory neural network for wind speed forecasting. Energy Convers Manag 2019;185:783–99.

[11] Sutton RS, Barto AG. Reinforcement learning: an introduction. The MIT Press; 2018.

[12] Wen LL, Zhou KL, Li J, Wang SY. Modified deep learning and reinforcement learning for an incentive-based demand response model. Energy 2020a;205:118019.

[13] Watkins CJ, Dayan P. Q-learning. Mach Learn 1992;8(3–4):279–92.

[14] Liu T, Tan ZH, Xu CL, Chen HX, Li ZF. Study on deep reinforcement learning techniques for building energy consumption forecasting. Energy Build 2020;208:109675.

[15] Kim TY, Cho SB. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019;182:72–81.

[16] Vaghefi A, Jafari MA, Bisse E, Lu Y, Brouwer J. Modeling and forecasting of cooling and electricity load demand. Appl Energy 2014;136:186–96.

[17] Xing YZ, Zhang S, Wen P, Shao LM, Rouyendegh BD. Load prediction in short-term implementing the multivariate quantile regression. Energy 2020;196:117035.

[18] Zhang L, Alahmad M, Wen J. Comparison of time-frequency-analysis techniques applied in building energy data noise cancellation for building load forecasting: a real-building case study. Energy Build 2021;231:110592.

[19] Chen Q, Zhang WY, Lou Y. Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. IEEE Access 2020;8:117365–76.

[20] Jurado S, Nebot A, Mugica F, Avellana N. Hybrid methodologies for electricity load forecasting: entropy-based feature selection with machine learning and soft computing techniques. Energy 2015;86:276–91.

[21] Chen YB, Xu P, Chu YY, Li WL, Wu YT, Ni LZ, et al. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. Appl Energy 2017;195:659–70.

[22] Yang AL, Li WD, Yang X. Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines. Knowl Base Syst 2019;163:159–73.

[23] Massaoudi M, Refaat SS, Chihi I, Trabelsi M, Oueslati FS, Abu-Rub H. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting. Energy 2021;214:118874.

[24] Ronao CA, Cho SB. Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models. Int J Distributed Sens Netw 2017. https://doi.org/10.1177/1550147716683687.

[25] Shi H, Xu MH, Li R. Deep learning for household load forecasting—a novel pooling deep RNN. IEEE Trans. Smart Grid 2017;9(5):5271–80.

[26] Atef S, Eltawil AB. Assessment of stacked unidirectional and bidirectional long short-term memory networks for electricity load forecasting. Elec Power Syst Res 2020;187:106489.

[27] Huang Q, Li JH, Zhu MS. An improved convolutional neural network with load range discretization for probabilistic load forecasting. Energy 2020. https://doi.org/10.1016/j.energy.2020.117902.

[28] Li LC, Meinrenken CJ, Modi V, Culligan PJ. Short-term apartment-level load forecasting using a modified neural network with selected auto-regressive features. Appl Energy 2021;287:116509.

[29] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing Atari with deep reinforcement learning. 2013. arXiv preprint arXiv:1312.5602.

[30] Feng C, Sun MC, Zhang J. Reinforced deterministic and probabilistic load forecasting via Q-learning dynamic model selection. IEEE Trans. Smart Grid 2019;11(2):1377–86.

[31] Tripathi A, Ashwin TS, Guddeti RMR. EmoWare: a context-aware framework for personalized video recommendation using affective video sequences. IEEE Access 2019;7:51185–200.

[32] Aboussalah AM, Lee CG. Continuous control with stacked deep dynamic recurrent reinforcement learning for portfolio optimization. Expert Syst Appl 2020;140:112891.

[33] Šemrov D, Marsetič R, Žura M, Todorovski L, Srdic A. Reinforcement learning approach for train rescheduling on a single-track railway. Transp Res Part B Methodol 2016;86:250–67.

[34] Ma M, Jin B, Luo S, Guo S, Huang H. A novel dynamic integration approach for multiple load forecasts based on Q-learning algorithm. Int. Trans. Electric.

Energy Syst. 2020. https://doi.org/10.1002/2050-7038.12146.

[35] Park H, Sim MK, Choi DG. An intelligent financial portfolio trading strategy using deep Q-learning. Expert Syst Appl 2020;158:113573.

[36] Pesce E, Montana G. Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. Mach Learn 2020:1—21.

[37] Wan ZQ, Li HP, He HB, Prokhorov D. Model-free real-time EV charging scheduling based on deep reinforcement learning. IEEE Trans. Smart Grid 2018;10(5):5246—57.

[38] Mocanu E, Mocanu DC, Nguyen PH, Liotta A, Webber ME, Gibescu M, et al. Online building energy optimization using deep reinforcement learning. IEEE Trans. Smart Grid 2018;10(4):3698—708.

[39] Wei P, Xia S, Chen RF, Qian JY, Li C, Jiang XF. A deep reinforcement learning based recommender system for occupant-driven energy optimization in commercial buildings. IEEE Internet Things J. 2020;7(7):6402—13.

[40] Keneshloo Y, Shi T, Ramakrishnan N, Reddy CK. Deep reinforcement learning for sequence-to-sequence models. IEEE Trans. Neural Netw. Learn. Syst. 2019;31(7):2469—89.

[41] Rahimpour Z, Verbič G, Chapman AC. Actor-critic learning for optimal building energy management with phase change materials. Elec Power Syst Res 2020;188:106543.

[42] Rolls ET, McCabe C, Redoute J. Expected value, reward outcome, and temporal difference error representations in a probabilistic decision task. Cerebr Cortex 2008;18(3):652—63.

[43] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. 2015. arXiv preprint arXiv: 1509.02971.

[44] Batten A, Thorpe J, Piegari R, Rosland AM. A resampling based grid search method to improve reliability and robustness of mixture-item response theory models of multimorbid high-risk patients. IEEE J. Biomed. Health Inform. 2019;24(6):1780—7.

[45] Wen LL, Zhou KL, Yang SL. Load demand forecasting of residential buildings using a deep learning model. Elec Power Syst Res 2020b;179:106073.

[46] Bendaoud NMM, Farah N. Using deep learning for short-term load forecasting. Neural Comput Appl 2020;32(18):15029—41.

[47] Feng ZK, Niu WJ, Tang ZY, Xu Y, Zhang HR. Evolutionary artificial intelligence model via cooperation search algorithm and extreme learning machine for multiple scales nonstationary hydrological time series prediction. J Hydrol 2021;595:126062.

[48] Fuertes AM, Izzeldin M, Kalotychou E. On forecasting daily stock volatility: the role of intraday information and market conditions. Int J Forecast 2009;25(2): 259—81.

[49] Niu WJ, Feng ZK, Feng BF, Xu YS, Min YW. Parallel computing and swarm intelligence based artificial intelligence model for multi-step-ahead hydrological time series prediction. Sustain. Cities Soc. 2021;66:102686.

[50] Liu H, Zhang XC, Zhang XT. PwAdaBoost: possible world based AdaBoost algorithm for classifying uncertain data. Knowl Base Syst 2019;186:104930.

[51] Yan JJ, Zhang ZN, Lin KH, Yang F, Luo XB. A hybrid scheme-based one-vs-all decision trees for multi-class classification tasks. Knowl Base Syst 2020;198: 105922.

[52] Sun R, Wang G, Zhang W, Hsu LT, Ochieng WY. A gradient boosting decision tree based GPS signal reception classification algorithm. Appl Soft Comput 2020;86:105942.

[53] Zhou P, Guo DW, Chai TY. Data-driven predictive control of molten iron quality in blast furnace ironmaking using multi-output LS-SVR based inverse system identification. Neurocomputing 2018;308:101—10.

[54] Asadi M, Jamali MAJ, Parsa S, Majidnezhad V. Detecting botnet by using particle swarm optimization algorithm based on voting system. Future Generat Comput Syst 2020;107:95—111.