Applied Thermal Engineering 254 (2024) 123857

Contents lists available at ScienceDirect

Applied Thermal Engineering

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

Combustion optimization study of pulverized coal boiler based on proximal policy optimization algorithm

Xuecheng Wu [a,b], Hongnan Zhang [a,b], Huafeng Chen [a], Shifeng Wang [c,*], Lingling Gong [d]

[a] Ningbo Innovation Center, Zhejiang University, Ningbo 315100
[b] State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027
[c] Hangzhou Vocational & Technical College, Hangzhou 310019
[d] Zhejiang Technical Institute of Economics, Hangzhou 310018

ARTICLE INFO

Keywords:
Pulverized coal boiler
Energy saving and emission reduction
Reinforcement learning
Proximal policy optimization algorithm
Online combustion optimization

ABSTRACT

In most industrial sectors, large coal-fired boilers are a source of carbon and pollutant emissions, so it is important to carry out combustion adjustment and optimize energy-saving operation of coal-fired boilers. Traditional combustion adjustment relies on human intervention, but manual adjustment is difficult to achieve synergistic optimization of NO_x and thermal efficiency at the same time, so there is a large adjustment space for boiler combustion optimization. Artificial intelligence technology can explore the potential of combustion optimization from boiler operation data. Currently, the boiler combustion optimization method based on supervised learning modeling and optimization algorithms has good optimization effect and high application value. At present, there are problems such as the combination of dynamic model and optimization algorithm is difficult and the optimization time is long, etc. This paper adopts feature classification and multi-model coupling to build a static-dynamic composite prediction model of boiler performance indicators, dynamic prediction model of boiler thermal efficiency and nitrogen oxides (NO_x) is established by using long short-term memory (LSTM) and one-dimensional convolutional neural network (1D_CNN). The model is categorized into static and dynamic models based on the input features, and the dynamic model is coupled with BP neural network to establish a static-dynamic composite prediction model and further couples the proximal policy optimization (PPO) reinforcement learning algorithm to establish a boiler in-place optimization strategy. Through the experimental validation of 5619 test cases, the strategy successfully achieves 63.5 % co-optimization of NO_x and thermal efficiency, with thermal efficiency increase ranging from 0-0.61 % and NO_x reduction ranging from 0-65 mg/m^3. Meanwhile, comparing the optimization effect of the PPO algorithm with that of the genetic algorithm (GA) shows that the PPO strategy has a more significant effect on NO_x reduction while keeping the thermal efficiency optimized. Moreover, the online decision-making speed of the PPO strategy is much higher than that of the GA, with an average time consumption of only 0.015 s, while the GA requires about 3 min for a single optimization, which indicates that the combustion optimization strategy of the PPO algorithm coupled with the composite prediction model has a significant advantage in realizing high-efficiency and accurate optimization.

1. Introduction

Large coal-fired boilers are necessary components in most industrial sectors, realizing industrial production, power supply and district heating, etc. Coal-fired boilers are the major source of carbon emissions and pollutant emissions in related industries, and NO_x is one of the main pollutants emitted from boilers. Most of the large coal-fired boilers are equipped with distributed control systems (DCS), recording a large amount of boiler operation history data. Based on the rich operating data of boilers, artificial intelligence technology can explore the potential of boiler combustion optimization from the data, and provide the boiler combustion adjustment plan by manual by computer with lower risk. It can improve the thermal efficiency of boilers as much as possible while ensuring the NO_x emission meets the standard, thus improving the boiler operation economy.

Establishing a boiler combustion optimization method generally consists of two steps: establishing a high-precision boiler performance prediction model; establishing a boiler combustion optimization algorithm based on the prediction model.

* Corresponding author.
E-mail address: wsf@hzvtc.edu.cn (S. Wang).

https://doi.org/10.1016/j.applthermaleng.2024.123857
Received 1 August 2023; Received in revised form 5 March 2024; Accepted 30 June 2024
Available online 1 July 2024
1359-4311/

Nomenclature

DCS	distributed control systems
PPO	proximal policy optimization
LSTM	long short-term memory
1D_CNN	one-dimensional convolutional neural network
NO_x	nitrogen oxide
GA	genetic algorithm
BPNN	back propagation neural network
BLM	bidirectional learning machine
SVR	support vector regression
ELM	extreme learning machine
PSO	particle swarm algorithm
ACO	ant colony algorithm
HDP	heuristic dynamic planning
A2C	advantage actor critic
DBQ	deep Q-learning
RNN	recurrent neural network
RMSE	root mean square error
Val_MSE	mean square error on the validation set
1D_MaxPool	one-dimensional maximal pooling
a	reinforcing learning action
s	reinforcing learning state
$r;R$	reinforcing learning reward
R_{dis}	reinforcing learning cumulative discount reward
tra	single-round interaction trajectory length
γ	cumulative reward discount factor
d_i	dynamic features
c_j	combustion tuning parameters
x_l	remaining static features except for combustion tuning parameters
$perf$	boiler performance indicators
$step$	dynamic features sequence length
$series$	dynamic features merging
$static$	static feature merging
k	number of dynamic features
n	number of combustion tuning parameters
p	number of combustion tuning parameters
ρ_e	thermal efficiency increase amplitude
ρ_n	NO_x increase amplitude
$\Delta action_j$	amplitude adjustment of combustion tuning parameters
Δeff	thermal efficiency increment
ΔNO_x	NO_x increment

In terms of predictive modeling, the "black box" models for data modeling are mainly based on supervised learning, and the models can be divided into static and dynamic categories [1,2]:

1) Static models: including back propagation neural network (BPNN) [3,4], bidirectional learning machine (BLM) [5] support vector regression (SVR) [6–8], extreme learning machine (ELM) [9–11], GRNN [12], etc. The inputs of such models have no time dimension and are therefore considered as static models. Although the static model can better predict the boiler performance under steady state conditions, it still has some limitations, and the dynamic prediction is relatively weak for variable operating conditions. In addition, the actual combination of optimization algorithms to achieve boiler combustion optimization also needs to be optimized for steady state conditions, this is contradictory to the reality of frequent changes in boiler loads.

2) Dynamic models: including one-dimensional convolutional neural network (1D_CNN) [13,14], long short-term memory (LSTM) [15–18], etc. Such models require the input data to contain a temporal dimension, and are therefore regarded as dynamic models. Dynamic models can predict boiler performance indicators online and dynamically, and better meet the needs of online boiler combustion optimization. Li N et al. [13] built a NO_x prediction model for coal-fired boilers based on 1D_CNN, and the root mean square error of the model was only 1.06–1.11 mg/m^3 on the test data. Guotian Yang et al. [17] built a NO_x prediction model for pulverized coal boilers based on LSTM with a root mean square error of 2.271 mg/m^3 on the test data. In terms of dynamic prediction on continuous data, Pan H et al. [16] compared LSTM with BP and SVR models in detail; Peng Tan et al. [15] also compared LSTM with SVR, and the results all showed that dynamic models such as LSTM and RNN have better dynamic prediction on continuous data.

Dynamic models are currently used for soft measurements and condition warnings, and relatively little research has been done on combining them with optimization algorithms to achieve online boiler combustion optimization. The key problem is the difficulty of combining dynamic models with optimization algorithms and whether the algorithms meet the speed requirements of online optimization. Cheng Y et al. [14] serialized the combustion control parameters and processed them using 1D_CNN, and coupled them with LSTM to build a prediction model to discretize the combustion tuning scheme to optimize the combination with the algorithm. Zhan X [18] et al. separated the input of the LSTM neural network and input various boiler operation parameters and combustion adjustment parameters into the LSTM network layer in stages, thus enhancing the influence of combustion adjustment parameters on the dynamic prediction model.

In terms of combining predictive models with optimization algorithms to achieve combustion optimization, the two main types of combined optimization algorithms are decision iteration and strategy training:

1) Decision iteration: Decision iteration method is based on heuristic algorithms, including ant colony algorithm (ACO) [7,8], genetic algorithm (GA) [3,4], particle swarm algorithm (PSO) [19,20], etc. L. Zheng et al. [24] compared in detail the optimization speed of GA, PSO, and ACO when combined with the SVR prediction model; the algorithm takes three to four minutes to compute, and even with the optimization improvements of algorithm, the speed is usually difficult to meet the demand of online optimization. Therefore, the decision iterative method is generally combined with a static model to optimize mainly for steady-state operating conditions.
2) Strategy training: The combustion optimization strategy contains the combustion optimization adjustment scheme for all boiler operating conditions. The strategy training method aims to first parameterize the combustion optimization strategy, and then pre-train the strategy through the parameters, and use the trained strategy directly for the boiler operating conditions that change in real time without online iterative calculation. Currently, the main ones are heuristic dynamic planning (HDP), case-based reasoning (CBR) [21], and reinforcement learning (RL) [14,22–24]. Niu Y et al. [21] used CBR to reason about the decision actions to be taken under a new combustion task by learning previous cases of combustion optimization tasks. Reinforcement learning algorithms including deep Q-learning (DQN), advantage actor critic(A2C), have been widely used in recent years in the fields of games [25], robotics [26], and power and energy [27,28]. Cheng Y et al. [14] established a discrete optimization method for boiler combustion tuning parameters based on the DQN algorithm. Adams D et al. [5] used A2C algorithm for the optimization of continuous variables such as primary and secondary airflow. Zhan X et al [18] built an online combustion optimization method based on A2C algorithm, and finally successfully applied it to power plants after the improvement of the training method and the dynamic prediction model. Since the strategy is trained offline and

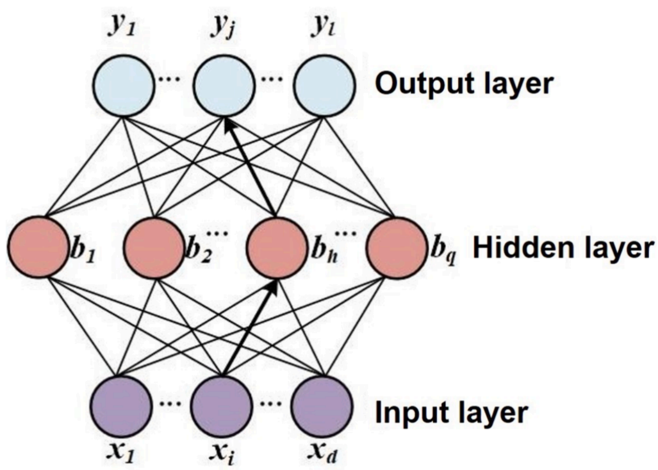

Fig. 1. BPNN structure.

decided online, the actual decision speed can reach millisecond, so it can quickly provide combustion adjustment plan to realize online optimization according to the real-time changing boiler status.

The current static model based on steady-state operating data combined with iterative optimization algorithm for combustion optimization is limited to the optimization of boiler steady-state operating conditions, and the optimization speed is slow; the dynamic model based on raw operating data has relatively good dynamic prediction effect on continuous operating conditions, but it is mainly used for soft measurement or status warning of boilers. In the combination of dynamic models and optimization algorithms, multi-model coupling, discretization of combustion adjustment parameters, and phasing of dynamic model inputs are mostly used. In terms of multi-model coupling, there is still lack of research on the effect of coupling on boiler dynamic prediction and on boiler combustion optimization when combined with optimization algorithms. In the aspect of improving the speed of algorithms, the current research mainly adopts the method of strategy pre-training, including heuristic dynamic planning, case inference and reinforcement learning, but the related research is still relatively few, among which, in reinforcement learning, the research on online combustion optimization of boilers using PPO algorithm is still lacking.

In this paper, the dynamic prediction model of boiler performance index is established by feature classification and multi-model coupling, and the boiler combustion online optimization strategy is established by coupling the prediction model with PPO reinforcement learning algorithm. The main research elements are constructed as follows:

- According to the boiler combustion mechanism and correlation analysis, the input features of the neural network are filtered and classified. The composite static-dynamic neural network prediction model based on feature classification and the dynamic neural network prediction model based on feature unclassified are built respectively, and compare the dynamic prediction performance of the composite model and the dynamic model.
- Composite and dynamic prediction models built are coupled with genetic algorithms to establish decision-based iterative boiler combustion optimization algorithms to study the combustion optimization effects of the algorithms, and to investigate the effects of the composite model structure based on feature classification on boiler combustion optimization through comparative analysis.
- The PPO reinforcement learning algorithm is coupled with a composite prediction model to establish an online optimization strategy for boiler combustion, and the time spent in actual decision making is significantly reduced by pre-training and online decision making of the strategy. The combustion optimization performance of the strategy is studied in detail, and the optimization effects of the strategy and GA are compared and analyzed under the same working conditions.

2. Methodology

2.1. Deep neural network

2.1.1. Back propagation neural network

Back propagation neural network (BPNN) is a basic neural network using back propagation algorithm, also known as fully connected neural network. the general structure of BPNN is shown in Fig. 1, which contains input layer, hidden layer and output layer, each layer network nodes are interrelated, and the network weight correction generally consists of forward propagation and error backward transmission.

2.1.2. Long short-term memory neural network

LSTM is a variant of recurrent neural network (RNN), which has a more elaborate internal “gating” mechanism than RNN [34]. LSTM can remember the valid information for a long time and forget the invalid information. When there are obvious temporal correlations between variables, LSTM can play a better prediction effect, and it is used more often when processing temporal data in practice.

The LSTM implicit layer consists of multiple LSTM neurons, and the hidden layer inputs include internal self-looping memory units in addition to external inputs and recurrent outputs. LSTM has internal gating units to control the flow of information, including input gates, forgetting gates and output gates. The input gate controls the computation of new states and how much is updated into the memory unit; the forgetting gate controls how much is forgotten in the current computation; and the output gate controls how much of the current output depends on the internal self-looping memory unit.

2.1.3. One-dimensional convolutional neural network

One-dimensional convolutional neural network (1D_CNN) can process sequence data, treating time as a spatial dimension and extracting features by sliding convolution on sequence data, and its effectiveness in processing certain sequence data is comparable to RNN and LSTM, while the computational cost is much smaller and the computation is much faster. At present, 1D_CNN has achieved great success in sequence processing tasks such as audio and text [29]. With a common convolutional kernel, the 1D_CNN can recognize local patterns in a sequence with sliding invariance, i.e., the pattern can be recognized at other locations in the sequence as well, which ultimately allows for efficient use of the data and serves as a feature filter.

2.2. Reinforcement learning

2.2.1. Actor-critic algorithm

Reinforcement learning is a constant interaction and continuous learning approach to decision optimization. The fundamentals of reinforcement learning: The intelligent agent is responsible for behavioral decisions and interacts with the environment. At the current time t, intelligent agent receives the state of the environment s_t, makes an immediate decision to give the action a_t, and the environment receives it after making state s_t shift to s_{t+1}, and gives *reward* r_t.

Stochastic policy search includes strategy gradient method, actor critic method, etc. The stochastic policy search method has been applied well to large state-space, behavior-space decision problems (e.g., robot control [35]). The stochastic policy is defined as $\pi(\theta)$, i.e., for one state s_t, the behavior a_t derived from the policy $\pi(\theta)$ satisfies the probability distribution $p(a_t|s_t;\theta)$.

For a Markov decision process (MDP), an interaction trajectory is defined as $\tau = (s_0, a_0, r_0, s_1, ..., s_t, a_t, r_t, s_H)$, γ is the discount factor, and $R(\tau)$ is the cumulative discounted reward for the remaining trajectory

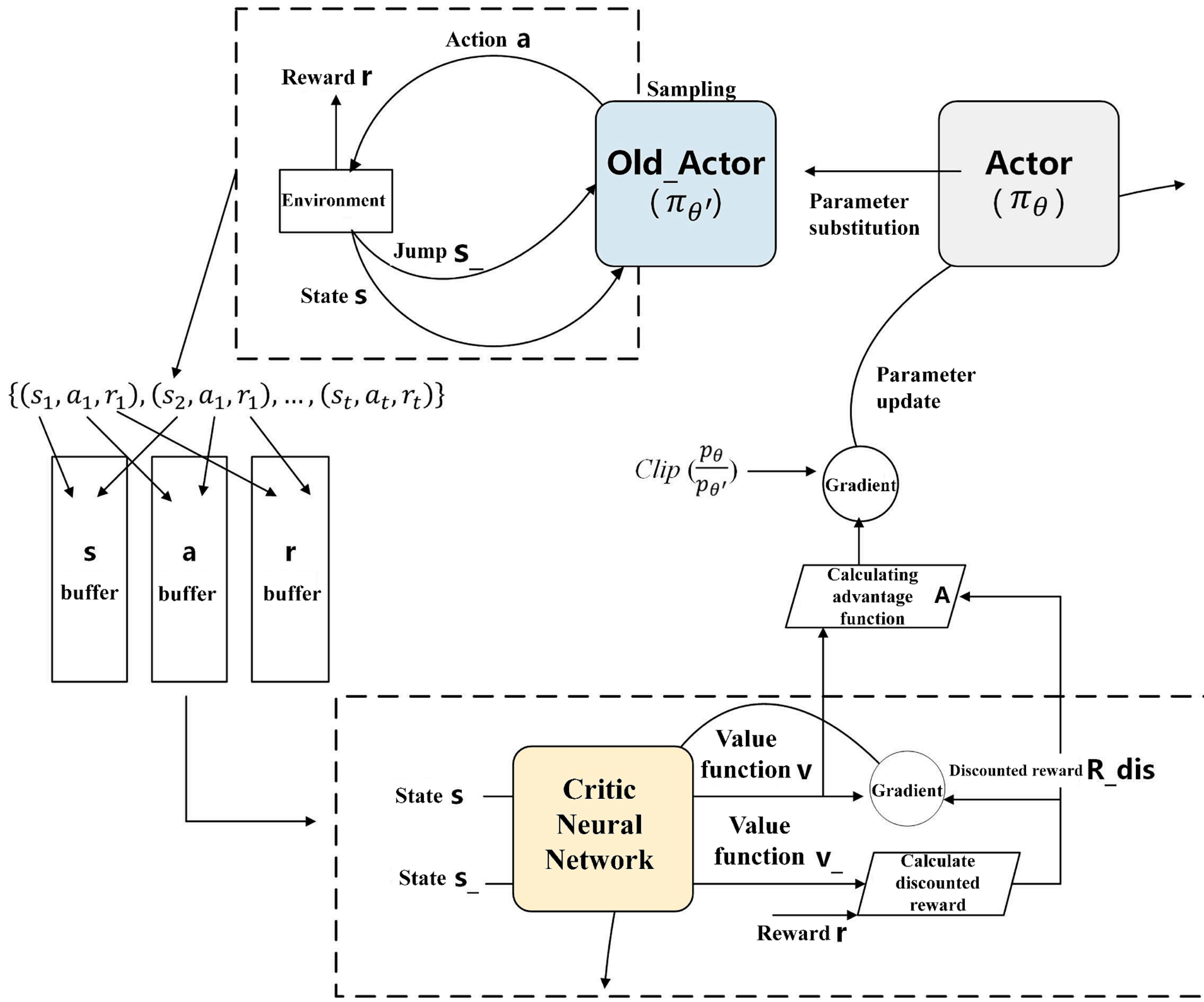

Fig. 2. Theory of PPO algorithm.

after the actor a_t is taken:

$$R(\tau) = \sum_{i=t}^{T} \gamma^{i-t} r_i \tag{1}$$

The goal of reinforcement learning is to maximize the total cumulative discounted payoff expectation $U(\theta)$, and the policy search problem is transformed into an optimization problem. Optimized sampling gradient method, gradient $\nabla_\theta U(\theta)$ calculated as follows:

$$\nabla_\theta U(\theta) = \sum_{\tau} R(\tau)\nabla p_\theta(\tau) \approx \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{t=0}^{H} R(\tau^i)\nabla_\theta \log p\left(a_t^i|s_t^i;\theta\right)\right) \tag{2}$$

Actor-Critic algorithm consists of Actor and Critic components. $\nabla_\theta \log p\left(a_t^i|s_t^i;\theta\right)$ plays the role of Actor, performing a certain behavior according to the current state; $R(\tau^i)$ plays the role of Critic, evaluating the current behavior according to reward. Actor is responsible for the parameterization of the policy, Critic outputs the state behavior value function $Q(s_t, a_t)$. Due to the addition of Critic, Actor-Critic algorithm allows for single-step updating and faster training. Actor network updating uses stochastic policy gradients for gradient ascent; Critic network updating performs gradient descent based on the temporal differential error (TD_{error}), as defined below:

$$TD_{error} = r_t + \gamma \max_a Q(s_t, a_t) - Q(s_t, a_t) \tag{3}$$

The policy gradient formula based on Critic imports the advantage function $A^\theta(s_t, a_t)$, and stochastic policy gradient is redefined as:

$$\nabla_\theta U(\theta) = E_{\tau\sim p_\theta(\tau)}\left[A^\theta(s_t, a_t)\nabla \log p_\theta(\tau)\right] \tag{4}$$

2.2.2. Proximal policy optimization

Actor-Critic algorithm belongs to the policy gradient method, but the policy gradient method has inherent drawbacks, i.e., unstable training and difficult convergence of the algorithm. Proximal Policy Optimization [30] (PPO) solves the problem of learning step size in the policy gradient method and works well on high-dimensional decision optimization problems.

The PPO algorithm is based on the Actor-Critic framework and therefore consists of Actor and Critic, the theory is shown in Fig. 2. where there exists the current Actor($\pi_{\theta'}$) and the previous Actor (Old_Actor($\pi_{\theta'}$)), Old_Actor interacts with the environment sampling trajectories (states, behaviors, rewards), and calculate the corresponding advantage function $A^\theta(s_t, a_t)$. Actor is trained offline using the trajectory adopted by Old_Actor, and stochastic policy gradient of the PPO's Actor update policy is defined as:

$$\nabla_\theta U(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}A^{\theta'}(s_t, a_t)\nabla \log p_\theta\left(a_t^n|s_t^n\right)\right] \tag{5}$$

At this point, the objective function of PPO algorithm is Eq.6, and the optimal parameters are found by optimizing the objective function:

$$J^{\theta'}(\theta) \approx E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}A^{\theta'}(s_t, a_t)\right] \tag{6}$$

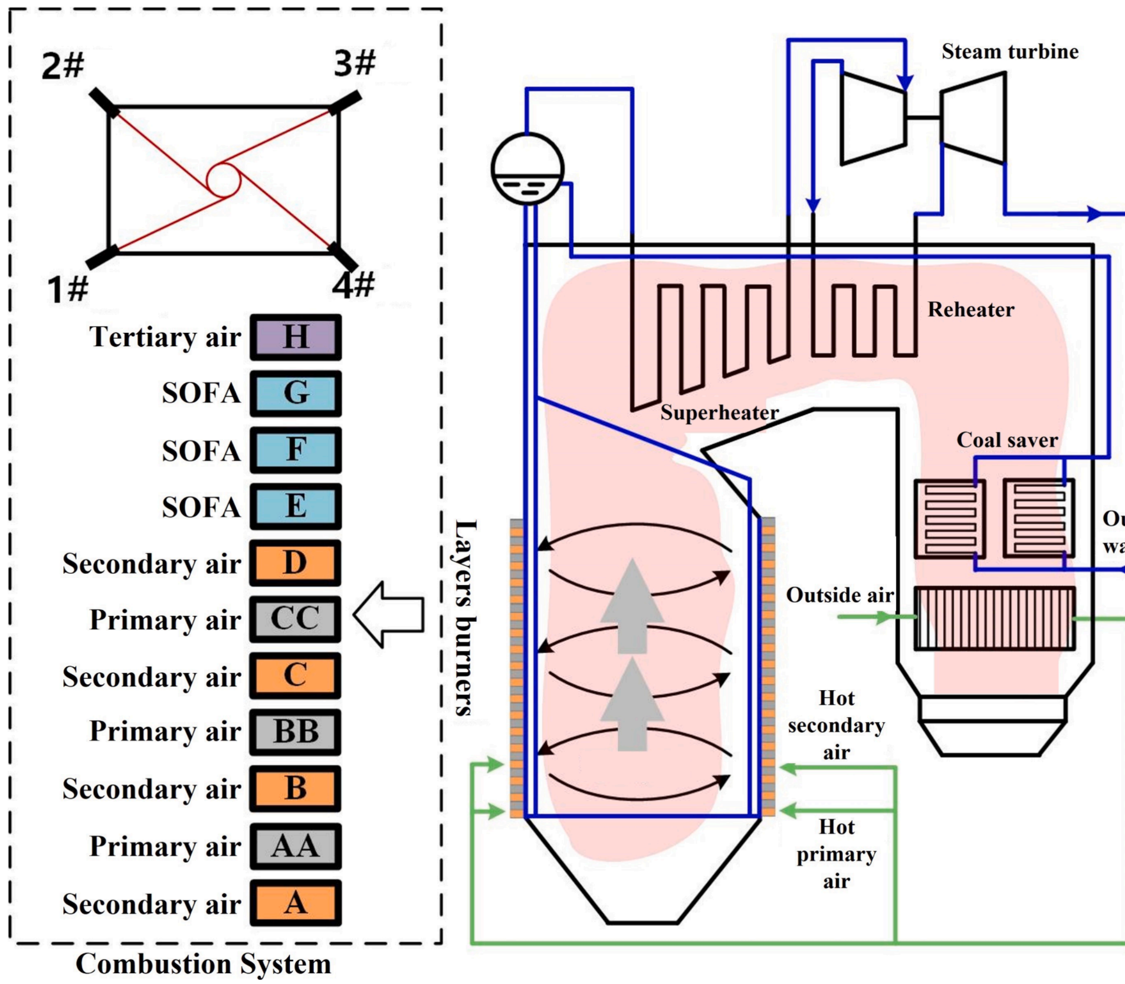

Fig. 3. Schematic diagram of the case four-corner cut circle pulverized coal boiler system.

PPO algorithm used in this paper is based on clipping method, where $\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}$ in the objective function are clipped, the upper and lower bounds are determined by ε. For updating Critic network, updating method is used like TD_{error}, and the gradient used for updating is defined as follows:

$$L(\phi) = -\sum_{(s_t)}\left(\sum_{i=t}^{T}\gamma^{i-t}r_i - V_\phi(s_t)\right)^2 \tag{7}$$

3. Boiler performance modeling

3.1. Description of the coal-fired boiler

The data studied in this paper are derived from the original DCS historical data of 410 t/h pulverized coal boiler of a coal-fired unit in a thermal power plant from December 1, 2021 to January 1, 2022, containing a total of 44,623 operating conditions.

Fig. 3 shows the schematic diagram of the thermal system of the case pulverized coal boiler. The system is mainly composed of steam ladle, water-cooled wall, each layer of burner, superheater, reheater, coal saver, air preheater, etc. The pulverized coal is preheated by the hot primary air, transported by the primary air to the furnace chamber for combustion, and completely burned to produce high-temperature flue gas by secondary air of each layer, separate over fire air (SOFA). Among them, the boiler is four-corner tangential circle combustion, using layered combustion system, burners are arranged in the four corners of the boiler, so that pulverized coal airflow in the furnace to form tangential circle combustion. A, B, C, D for four layers of secondary air; AA, BB, CC for three layers of primary air; E, F, G for three layers of separate over fire air (SOFA), led from the total secondary air duct; H is three air. The generated high temperature flue gas flows through the water-cooled wall, superheater, reheater, coal saver and air preheater in turn and then enters the flue gas super clean system. The saturated steam in the ladle becomes superheated steam through the superheater and is sent to the turbine to do work, and finally discharged to the condensing system to recover latent heat.

3.2. Thermal efficiency

For large power station pulverized coal boilers, boiler operation thermal efficiency has two calculation methods: positive balance method and counterbalance method. Among them, the positive balance method is relatively complex, requiring the calculation of the effective heat absorption of the mass in each process; the counter-balance method is relatively simple, from the point of view of the heat loss of thermal efficiency, the calculation formula is as follows:

$$\begin{aligned}\eta_t &= (1-\frac{Q_{loss}}{Q_{in}})\times 100\% \\ &= 100 - q_2 - q_3 - q_4 - q_5 - q_6\end{aligned} \tag{8}$$

In the above equation, Q_{loss} indicates the total heat loss, accordingly, including exhaust heat loss q_2, %, gas incomplete combustion heat loss q_3, %, solid incomplete combustion heat loss q_4, %, boiler heat loss q_5, %, ash physical sensible heat loss q_6, %. Q_{in} indicates the amount of heat released from the fuel, kJ/kg.

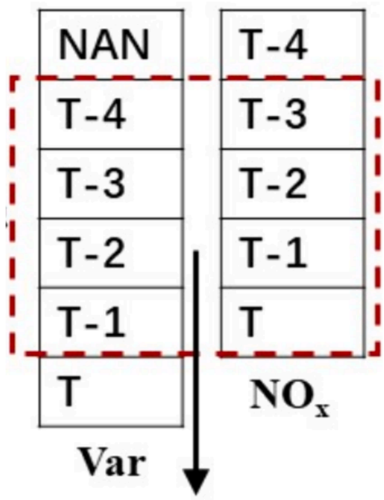

Fig. 4. Schematic diagram of time series.

Table 1
Feature classification results.

	Static		Dynamic	
	Feature category	number	Feature category	number
NO_x model input feature classification	Secondary air baffle Opening (A-D (#1-#4))	16	Primary air baffle Opening (AA(#1), BB(#1))	2
	Air preheater outlet flue temperature	4	Load	1
	Primary air outlet temperature (AA(#1-#4), BB(#1,#2,#4))	6	Main steam pressure	1
	Flue gas oxygenation	2	Total air pressure (#1-#3)	3
			Primary air temperature	1
			Furnace outlet flue gas temperature	1
			Primary air pressure (AA#3)	1
	Total	28	Total	10
Thermal efficiency model input feature classification	Secondary air baffle Opening (B-D (#1-#4))	12	Primary air baffle Opening (AA(#1, #4),BB(#1,#4),CC (#4))	5
	Air preheater outlet flue temperature	4	Load	1
	Primary air outlet temperature (AA-BB(#1-#4))	8	Main steam pressure	1
	Flue gas oxygenation	2	Total air pressure (#4)	1
	Primary air mixing box temperature	2	Furnace outlet flue gas temperature	1
			SOFA dampers opening (E(#1, #2))	2
	total	28	total	11

3.3. Feature engineering

3.3.1. Feature selection

The input to the neural network should contain as many features as possible that are strongly correlated with the output target. When establishing the boiler performance index prediction model, the model input features can be screened according to the boiler combustion mechanism, Pearson correlation coefficient [6,16], gray relation analysis (GRA) [16] or gradient enhancement tree [31], etc., and the features that are strongly correlated with the model performance index can be selected to improve the prediction performance of the model.

Depending on the combustion principle, the combustion parameters affecting the thermal efficiency and NO_x concentration of the boiler are selected as part of the model input characteristics. The secondary air has a large impact on both the concentration of raw generated NO_x and the thermal efficiency of the boiler; the primary air mainly plays the role of preheating and conveying pulverized coal, providing the oxygen for the combustion, generally accounting for a small proportion of the total air volume, which has a small impact on NO_x concentration and boiler thermal efficiency; the combustion exhaust air mainly plays the role of reducing the incomplete combustion products, which is part of the total secondary air, has a certain impact on NO_x and boiler thermal efficiency; the total air volume determines the size of each layer of air, reacting through the flue gas oxygen content, which has a greater impact on NO_x and thermal efficiency.

The correlations between different input characteristics and thermal efficiency and NO_x are different. In this paper, the Pearson and Spearman coefficients of all the input features of the initial screening with thermal efficiency and NO_x are calculated. After the significance test, the Pearson and Spearman corresponding features with significance less than 0.01 are selected, and the features with absolute values of Person or Spearman coefficients greater than 0.2 are selected among the remaining features, the NO_x model and the thermal efficiency model have a total number of 38 and 39 screening features.

In the air-graded combustion of pulverized coal boiler, the secondary air volume, the combustion air volume and the total air volume can affect the thermal efficiency and NO_x concentration of the boiler at the same time, and the effect of the combustion air swing opening is not considered because the variation of the SOFA nozzle swing openings in each layer of this boiler is not large and this is not conducive to data modeling. Finally, the final combustion adjustment parameters are determined by combining the mechanism screening (correlation prioritization), specifically the 12 secondary air door openings (B-D) and flue gas oxygen content.

3.3.2. Feature classification

Due to the delayed nature of boiler combustion, there is a time correlation between some of the features and NO_x and thermal efficiency, therefore, it is necessary to delay the features by a certain time series, i.e., sliding the features into the past by a certain time step to form a time series. Time lagged cross correlation [32,33] can be obtained by gradually sliding one of the time series and iteratively calculating the Person correlation coefficient between the two series. The time series sliding principle is shown in Fig. 4 (taking NO_x as an example), and when the sliding time step is 1, the calculated Pearson correlation coefficient is the correlation coefficient between the current NO_x series and the characteristic series before a time step.

Based on the screened input features, the NO_x or thermal efficiency sequences are kept constant in this process, and the feature sequences are slid downward with an interval of 1 time step (1 min), and the calculated Pearson coefficients are absolute values, and the feature sequences decrease more significantly with the increase of the sliding time step, implying that the correlation between the NO_x sequences and thermal efficiency sequences at the current moment, and these feature sequences at the past moment is poor, indicating that such features are more suitable to be kept intact and considered as static features. The

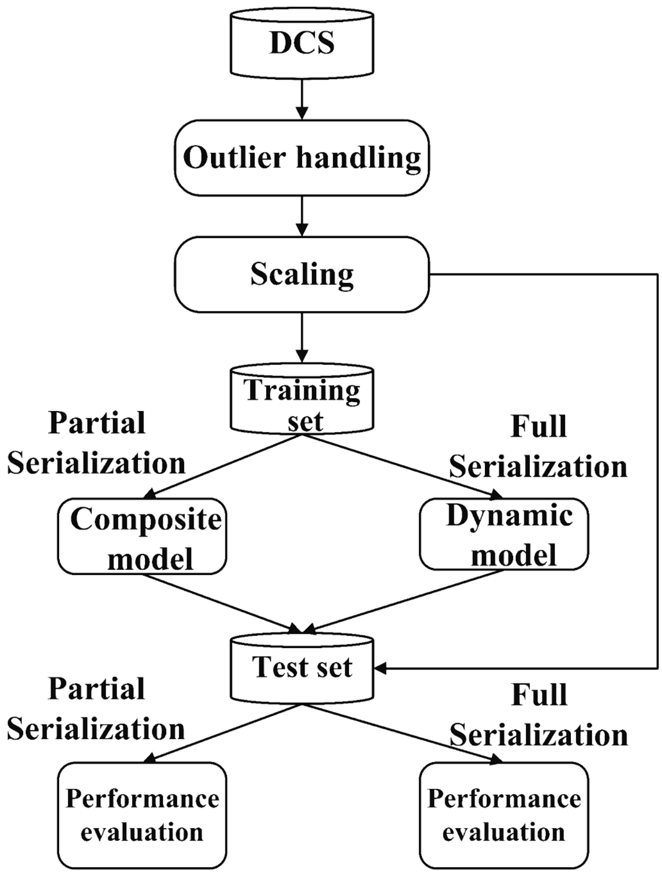

Fig. 5. Model training process.

correlation of other feature sequences with NO_x and thermal efficiency is almost constant with the increase of sliding time step, which means that the correlation between NO_x sequences at the current moment and these feature sequences at the past moment is still good, and such features can be input to the neural network for time-delayed serialization and can be regarded as dynamic features. The final classification results are shown in Table 1.

3.4. Data preprocessing

3.4.1. Exception handling

There are a certain number of outliers in the original data. In this paper, the 3sigma criterion is used to eliminate the outliers, and the 3sigma function is as follows:

$$f(x) = \begin{cases} 1, & x < \mu - 3\sigma \\ 1, & x > \mu + 3\sigma \\ 0, & others \end{cases} \tag{9}$$

In the above equation, x is the thermal efficiency or NO_x value of the current operating condition of the boiler, μ and σ is the sample mean and sample standard deviation of the thermal efficiency or NO_x. 1 indicates that the data is abnormal, and 0 indicates that the data is normal. Finally, 201 anomalies were eliminated from the 44,623 data.

3.4.2. Standardization

Raw data standardization is a necessary step in supervised learning regression modeling, and unifying the data to the same order of magnitude can facilitate model convergence and improve model prediction accuracy. In this paper, maximum-minimum normalization is used to convert all the features of the original data to the [0,1] interval, and when combining with the optimization algorithm to adjust the size of the feature value, it cannot exceed the [0,1] interval, so as to ensure that it is in the same data space.

$$X_i = \frac{X_{i,\max} - X_i}{X_{i,\max} - X_{i,\min}} \tag{10}$$

where X_i is an input feature,$X_{i,\max}$ and $X_{i,\min}$ are the maximum and minimum values of the feature.

3.4.3. Feature serialization

Dynamic features are considered to be time-delayed and need to be serialized. The so-called serialization is to form a time series by delaying the features by a certain time step, the dynamic features are delayed by the same time step, and the length of the sequence is 7. All the dynamic features are delayed and merged into a series, and the current moment is defined as follows:

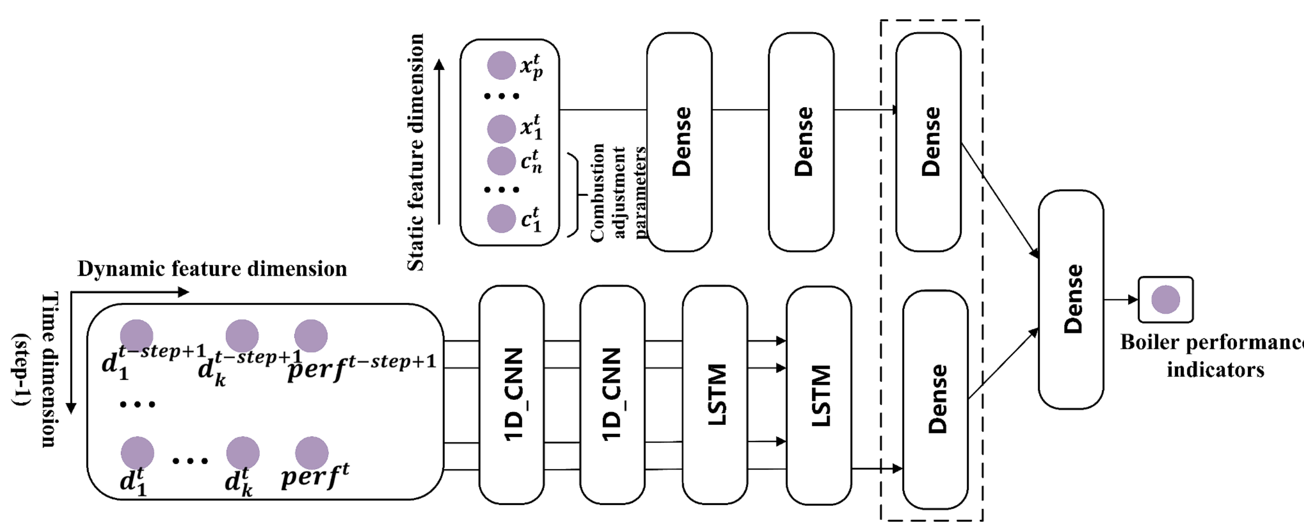

Fig. 6. Static-dynamic composite neural network model structure.

Table 2
Parameter definition of composite model.

Hidden layer structure		Weighting dimension	Activation function	Thermal efficiency model training weights	NO_x model training weights
Static	Dense	128	Relu	3712	3712
	Dense	64	Relu	8256	8256
	Dense	64	Relu	4160	
Dynamic	1D_CNN	64	Relu	1472	1344
	1D_CNN	64	Relu	8256	8256
	LSTM	64	/	33,024	33,024
	LSTM	64	/	33,024	33,024
	Dense	64	Relu	4160	4160
Public	Dense	128	Relu	16,512	16,512
			Total	112,576	112,448

$$series_t = \begin{pmatrix} d_i^{t-step+1} & perf^{t-step+1} \\ \vdots & \vdots \\ d_i^{t-1} & perf^{t-1} \\ d_i^t & perf^t \end{pmatrix}, i \in [1, k] \tag{11}$$

where, d_i^t denotes a certain dynamic feature at the current moment; k denotes the number of dynamic features ($k = 11$ for the thermal efficiency model and $k = 10$ for the NO_x model); and *perf* is the performance metric, i.e., NO_x or thermal efficiency itself. The predictions are for future performance metrics $perf^{t+1}$, so the upper bound of the feature sequence is *t-step* + *1*.

3.4.4. Dataset creation

Based on the original DCS data, the neural network model training dataset and test dataset were established. The dataset establishment process is shown in Fig. 5. First, the raw DCS data is anomalously processed and normalized. Second, 2000 (about 5 %) consecutive data are randomly selected from the normalized total dataset as the test set, and the remaining data are used as the training set. In this paper, we propose to build both the composite neural network model based on feature classification and the dynamic neural network model based on feature non-classification, so when building the dataset, the features are partially serialized or fully serialized according to the composite model or the dynamic model, respectively. Finally, the trained composite model and dynamic model are tested on the same test set.

3.5. Boiler efficiency and NO_x modeling

The composite model considers feature classification, coupling the dynamic neural network with the static neural network structure to handle static and dynamic features separately. The final structure of the static-dynamic composite neural network model is shown in Fig. 6, where 3 layers of fully connected network layers (Dense) are stacked to form a BP static neural network structure to handle static features, and 2 layers of 1D_CNN and 2 layers of LSTM are stacked to form a dynamic neural network structure to handle dynamic features. 1D_CNN first extracts potential features from the time series and inputs the extracted feature sequences into the LSTM network layer for refinement. The static network structure and dynamic network structure are first decoupled and then coupled, connecting the output of the LSTM network at the current moment with the Dense layer and merging it with the final Dense layer of the BP network, and then stacking the Dense layers for transition after the merger, finally constituting a complete multi-input, single-output composite neural network structure.

Since the number of static and dynamic input features of the thermal efficiency prediction model and the NO_x prediction model are not very different, the two prediction models have the same composite neural network structure and hyperparameter settings, as shown in Table 2. Among them, the sliding window size of both 1D convolutional networks is 2; the final Dense network layer of both static and dynamic network structures was L2 regularized to reduce the overfitting of the models.

The dynamic neural network model is also built in this paper to constitute a comparison with the composite neural network model. The dynamic neural network model retains only the dynamic feature processing structure based on the conforming model, i.e., two layers of 1D_CNN and 2 layers of LSTM, and two layers of Dense network for transition in the tail, and the final dynamic neural network model structure is shown in Fig. 7.

Although the dynamic model has the same dynamic feature processing structure as the composite model, the number of input dynamic features is increased due to the complete serialization of the dynamic model based on features, so the hyperparameters of the dynamic neural network need to be redefined to increase the weight dimension of the network or the number of convolution kernels to ensure the same learning capability. The hyperparameters of the established boiler thermal efficiency and NO_x dynamic neural network prediction model are defined as shown in Table 3.

The model uses the Adam optimization algorithm to update the network parameters based on the gradient. The model training method is small batch training, which is executed according to the built-in function of Tensorflow. Randomly selected data from the batch were fed into the neural network with a batch size of 128. 20 % of the data in the training set was randomly used as a validation set to observe

Table 3
Parameter definition of dynamic model.

Hidden layer structure	Weighting dimension	Thermal efficiency model training weights	NO_x model training weights
1D_CNN	128	10,112	9856
1D_CNN	64	16,448	16,448
LSTM	64	33,024	33,024
LSTM	64	33,024	33,024
Dense	128	8320	8320
Dense	128	16,512	16,512
	Total	117,440	117,184

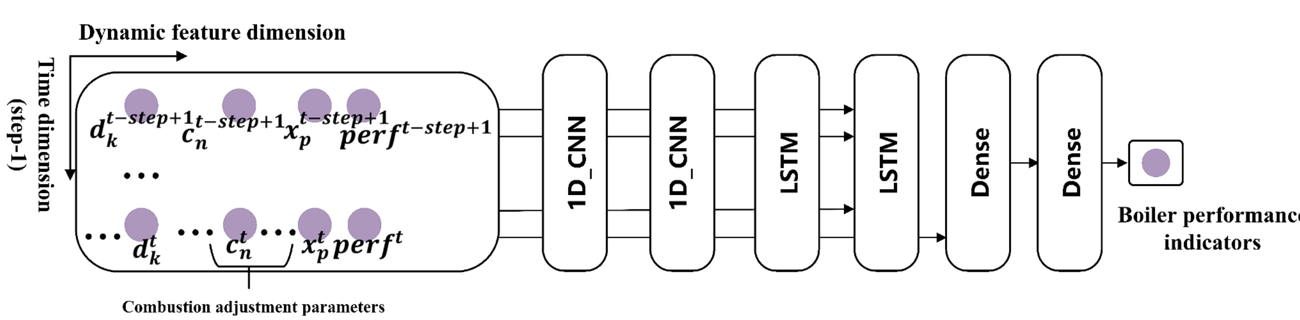

Fig. 7. Dynamic neural network model structure.

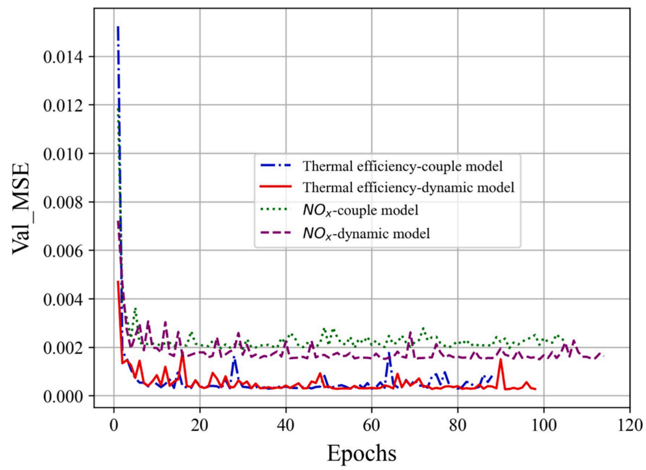

Fig. 8. All model training losses.

whether the model was overfitting. The total number of cycles of model training (max_epochs) was 500, and the model training was set to stop training early. After each round of training, the model would calculate the mean square error on the validation set (*Val_MSE*), the definition is as follows. When the number of times that the increase of *Val_MSE* was not more than 0.0001 consecutively exceeded 80 rounds, it was determined that the model was no longer optimized, and the training was stopped at this time to prevent the model from over-training and overfitting. The model parameters with the smallest *Val_MSE* were saved during training.

$$Val\_MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i' - y_i\right)^2 \tag{12}$$

where, y_i' is the model predicted value; y_i is the sample true value; n is the number of samples in the test set.

Eventually, the changes in the mean square error of the validation set during training of the composite and dynamic models are shown in Fig. 8. The *VAL_MSE* of both models on the validation set decreases rapidly and then remains almost constant without obvious signs of rebound, indicating that the models converge and do not show obvious signs of overfitting under the effect of L2 regularization and early stopping of training.

3.6. Superiority discussion

3.6.1. Comparison of predicting performance

Root mean square error (RMSE) is the standard deviation of the prediction error and is defined as in Eq.8. RMSE is more sensitive to larger errors in the sample, and the smaller RMSE, the better comprehensive performance of the model prediction:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i' - y_i\right)^2} \tag{13}$$

where y_i' is the model prediction value; y_i is the sample true value; n is the number of samples in the test set.

In this paper, the dynamic neural network model and the composite neural network model built are trained with the same training set and tested in the same test set after training, and the comparison of the prediction performance of the composite model and the dynamic model is shown in Fig. 9 and Fig. 10. As shown in Fig. 9, the two models have little difference in the prediction of the true thermal efficiency and NO_x

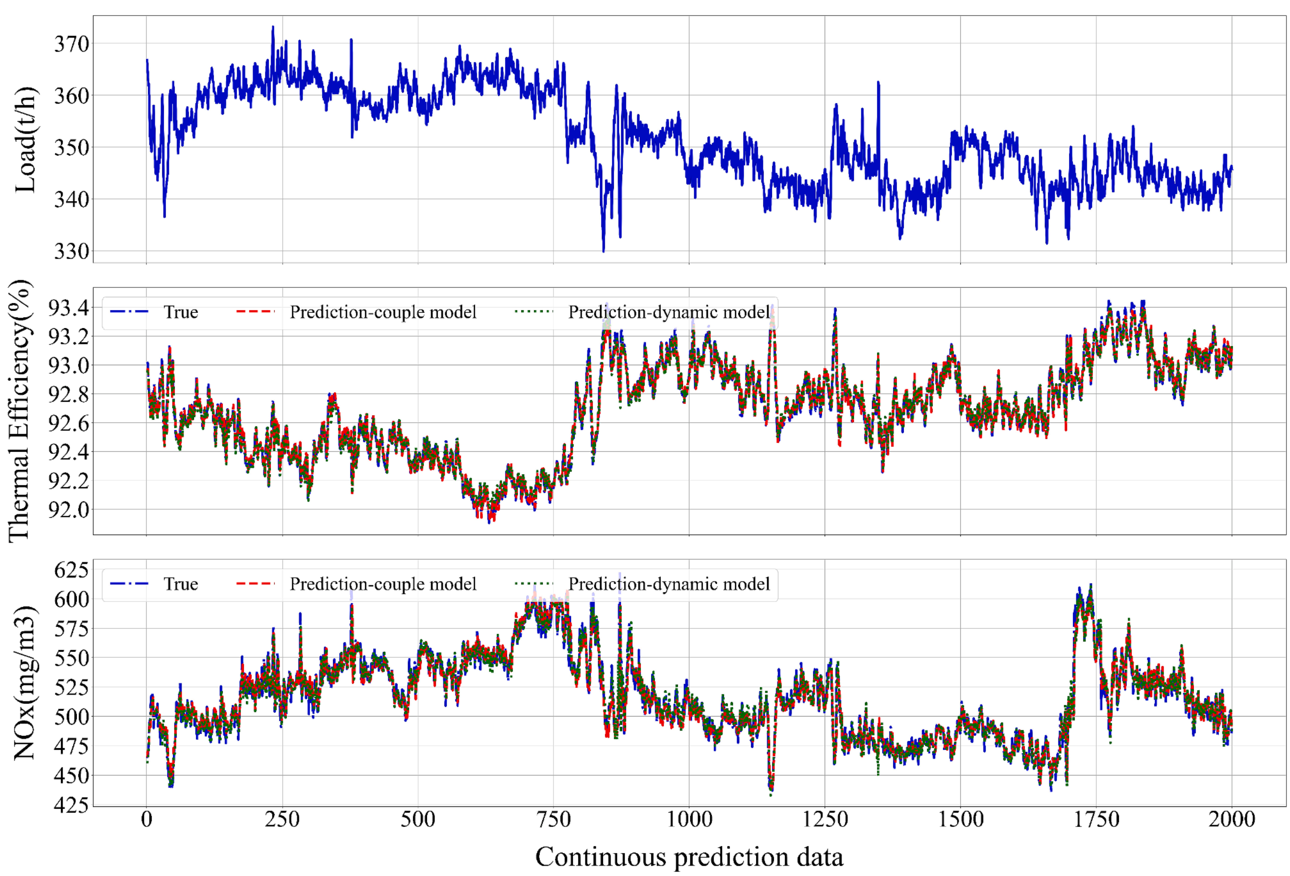

Fig. 9. Comparison of the prediction results of composite model and dynamic model.

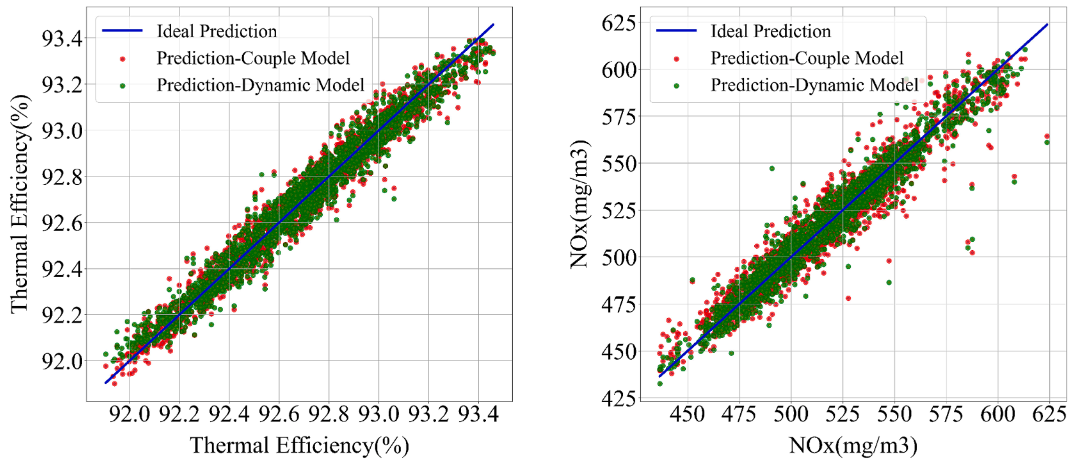

Fig. 10. Comparison of actual and ideal regression lines for composite and dynamic model.

Table 4
Composite model vs. dynamic model (performance comparison).

	Thermal efficiency predict model	NO_x predict model
	RMSE (%)	RMSE (mg/m^3)
Dynamic model	0.062	8.22
Composite model	0.063	9.28

Table 5
Composite model vs. Literature (*RMSE*).

NO_x model *RMSE* (mg/m^3)		Thermal efficiency model *RMSE* (%)	
Composite model	9.28	Composite model	0.063
Literature [6](LSSVM)	3.94	Literature [21](LSSVM)	0.094
Literature [11](ELM)	7.88	Literature [23](LSSVM)	1.510
Literature [9](SVM)	8.67	Literature[14](LSTM + 1D_CNN)	0.140

concentration values, and both can achieve better online and dynamic prediction at different loads. As shown in the *RMSE* of Table 4, the prediction performance of the composite model and the dynamic model are not very different, and the composite model predicts NO_x slightly worse, and this can also be seen from the comparison of the actual regression line with the ideal regression line in Fig. 10, the prediction points of the NO_x composite model are a bit more divergent around the ideal regression line compared to the dynamic model.

The *RMSE* evaluation index of the composite model is shown in Table 5. Since there are no explicit upper and lower bounds for *RMSE* for comparison, some relevant literature results are selected for comparison in this paper. It can be seen that the accuracy of the thermal efficiency and NO_x prediction model is not much different from the literature results, and the *RMSE* of the model on the test set is smaller, for the thermal efficiency model, the *RMSE* is 0.063 %; for the NOx model, the *RMSE* is 9.28 mg/m^3, indicating that the thermal efficiency and NO_x prediction model constructed in this paper has reached the accuracy requirements for further combining with the optimization algorithm, and thus it can be carried out boiler combustion optimization study.

3.6.2. Comparison of combustion optimization performance

To compare the performance of the dynamic and composite models on the combustion optimization task in combination with the optimization algorithm, this paper references to the relevant research [3,4], and couples GA with the composite prediction model built in the previous paper to establish the boiler combustion optimization algorithm, and tests the optimization on the same 30 operating conditions. GA is responsible for generating the initial solution set (population), which contains one individual that is decoded into a combustion adjustment scheme, including 13 adjustment ranges of combustion adjustment parameters. Since the data are pre-normalized, the upper and lower bounds of the 13 adjustment magnitudes are the same, with the lower bound being −16.7 % and the upper bound being 16.7 % (Limits the range of combustion adjustments to ensure that single-step adjustments are not too large). The population size was 150 and the individual coding method was Gray coding.

The fitness function $f(e^{t+1}, e^t, n^{t+1}, n^t, action)$ is defined as follows:

$$Max \quad f = \begin{cases} 1 + 10\rho_e - 0.01\sum_{j=1}^{n}\Delta action_j^t, & \rho_e \geqslant 1\%, \rho_n < 0 \\ -1, & others \end{cases} \tag{14}$$

$$\begin{aligned} &\rho_e = \frac{e^{t+1} - e^t}{e^t}, \quad e^t \neq 0 \\ &\rho_n = \frac{n^{t+1} - n^t}{n^{t-1}}, \quad n^t \neq 0 \\ &s.t. \quad -\frac{1}{6} < \Delta action_j^t < \frac{1}{6} \end{aligned} \tag{15}$$

where e^t and e^{t+1} denote thermal efficiency before and after combustion adjustment,n^t and n^{t+1} denote NO_x concentration before and after combustion adjustment,e^{t+1} and n^{t+1} are obtained by applying the action to the optimized conditions and inputting the thermal efficiency and NO_x prediction model,ρ_e and ρ_n denote the relative increase of thermal efficiency and the relative increase of NO_x concentration, *action* is a behavior vector, representing an individual in GA and a burn adjustment scheme, including 13 adjustment range $\Delta action_j$, defined as:

$$action = \{\Delta action_1, \Delta action_2, \ldots, \Delta action_{13}\} \tag{16}$$

The GA algorithm performs individual selection based on the fitness values of the individuals in the population, followed by crossover, mutation and retention of historical elite individuals to form a new population of the next generation, completing one population iteration. After several iterations, the individuals in the GA population are optimized, and the individual that can make the fitness value reach the global optimum is finally selected as the optimal combustion adjustment scheme under the current boiler operating conditions to be optimized. In this paper, the stopping condition of GA is set to reach the maximum number of population iterations of 80 rounds.

The boiler combustion optimization in this paper takes boiler energy saving as the overall goal, and improves boiler thermal efficiency without affecting NO_x or even emission reduction. A thermal efficiency optimization benchmark of 1 % is set, and when the relative increase in thermal efficiency is greater than 1 % and the relative increase in NO_x is less than 0, it is determined that the boiler thermal efficiency has been

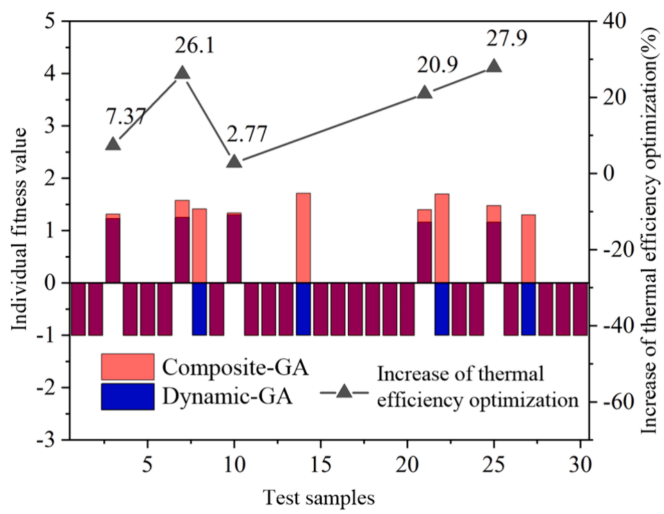

Fig. 11. Comparison of optimization effect of composite model and dynamic model combined with GA.

optimized and has no effect on NO_x generation. At the same time, setting a benchmark of 1 % reduces the impact of model prediction errors. When the boiler performance is deemed optimized, the fitness value is positive and the sum of the combustion adjustments is imposed as a penalty for over-adjustment. Other cases are considered not optimized and the fitness value is negative.

The optimal individual fitness value derived from GA based on the dynamic model is basically negative in Fig. 11. Compared with the composite model based on GA that achieves 30 % of the thermal efficiency optimization of the operating conditions, dynamic model based on GA achieves only about 16 % of the thermal efficiency optimization of the operating conditions, and the optimization divergence is relatively high. In addition, for the successful combustion optimization conditions, the thermal efficiency increase based on the composite model is 2–28 % higher than that of the dynamic model, indicating that the optimization effect of the GA algorithm based on the dynamic model is relatively poorer when successfully optimized, specifically the thermal efficiency increase is relatively low.

4. Optimization of boiler combustion

4.1. Construction of combustion optimization policy

As can be seen in Fig. 12, boiler combustion online optimization strategy consists of PPO algorithm coupled with composite prediction model. Reinforcement learning environment constructed from thermal efficiency and NO_x prediction models that give states, complete state transitions and give immediate rewards; PPO acts as a reinforcement learning intelligence, initializing the boiler combustion optimization strategy by constructing an Actor neural network, giving the corresponding combustion adjustment scheme by the strategy according to the boiler operation status; PPO agent trains Actor based on the

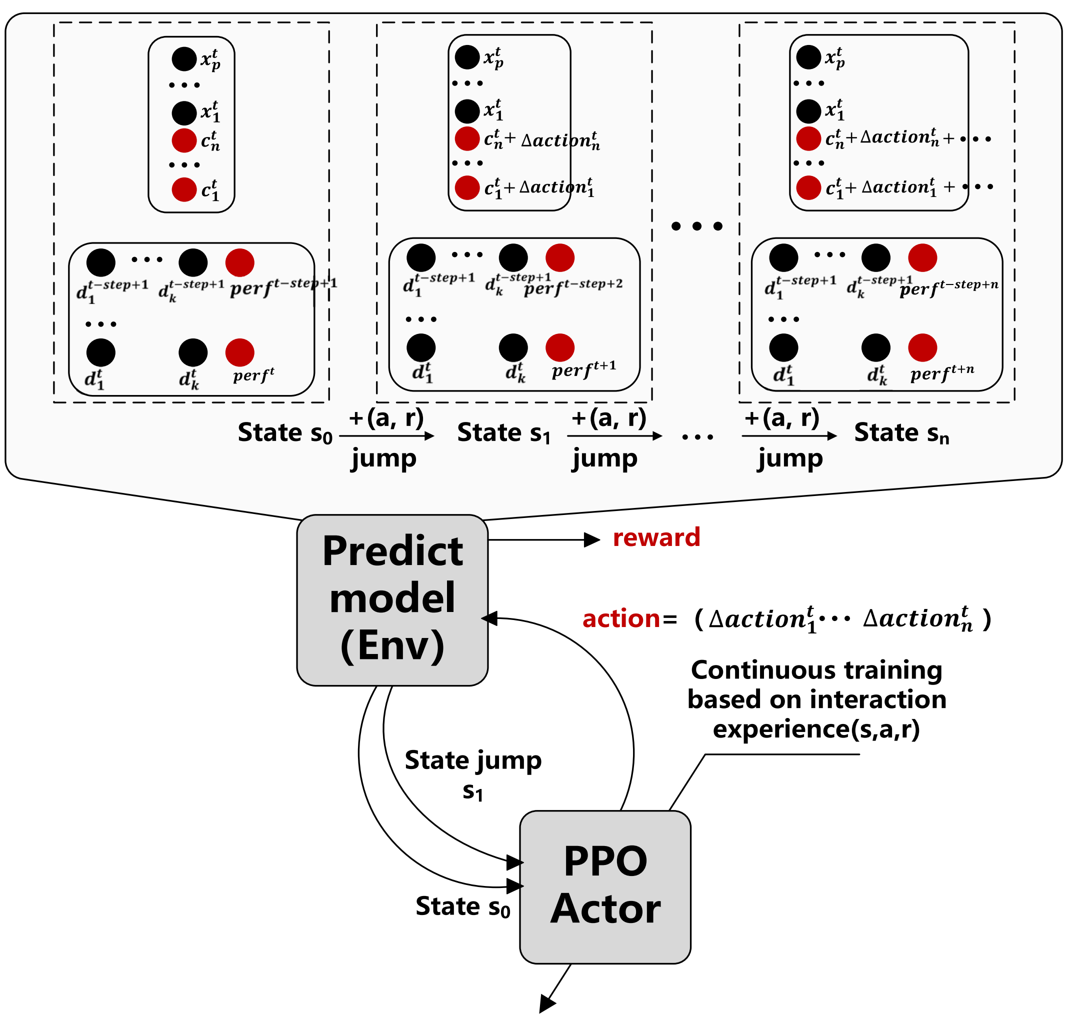

Fig. 12. Composite prediction model and PPO algorithm coupling framework.

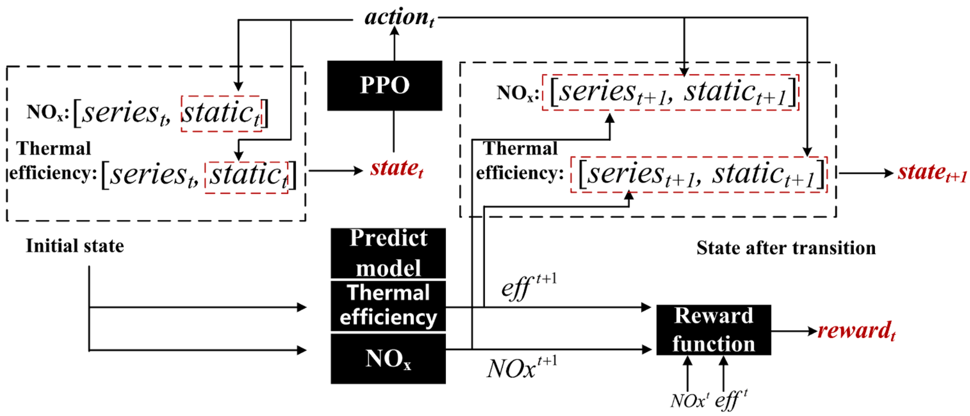

Fig. 13. Environment-to-state transition.

Table 6a
Actor and Critic neural network structure.

	Network structure	Weighting dimension	Activation function
Static feature processing	Dense	256	tanh
Dynamic feature processing	1D_CNN	64	tanh
	1D_MaxPool	/	/
	1D_CNN	64	tanh
	1D_MaxPool	/	/
	Flatten	/	/
Merger	Dense	128	tanh
Output	Dense	Actor (13) ; Critic (1)	Actor (tanh) ; Critic (None)

extensive experience of interacting with the environment, finally enabling the boiler combustion optimization strategy to complete the training. Trained strategies can be used directly for boiler operation conditions that the agent have never seen before, providing combustion optimization tuning solutions online.

4.1.1. Reinforcement learning environment

Reinforcement learning environment simulates the real environment of boiler combustion, the boiler operation state needs to be accurately transitioned next state according to the combustion adjustment scheme, and gives the reward corresponding to the scheme.

The initial state set of boiler operation is the operating load interval [332 t/h, 362 t/h] with a size of 28097, selecting 20 % (5619) as the test state set and the remaining 80 % as the training state set. When PPO agent interact with the environment, the environment selects randomly from the training state set as the initial state; when testing the trained combustion optimization strategy, the environment selects the initial state from the test state set in order.

Environment to state transition is mainly referred to the related literature [14]. In Fig. 13, for a random initial state $[series_t, static_t]$ at the current moment t, states are merged into the PPO after which the current combustion adjustment scheme $action_t$ is obtained and the combustion adjustment parameters are applied to the initial state, then static feature $static_t$ transitions to $static_{t+1}$. The state $[series_t, static_t]$ after applying the combustion adjustment scheme is fed into the thermal efficiency and NO_x prediction model to obtain eff^{t+1} and NO_x^{t+1}, and this is used to update the thermal efficiency and NO_x performance parameter sequence,so that the dynamic characteristics $series_t$ transitions to $series_{t+1}$,

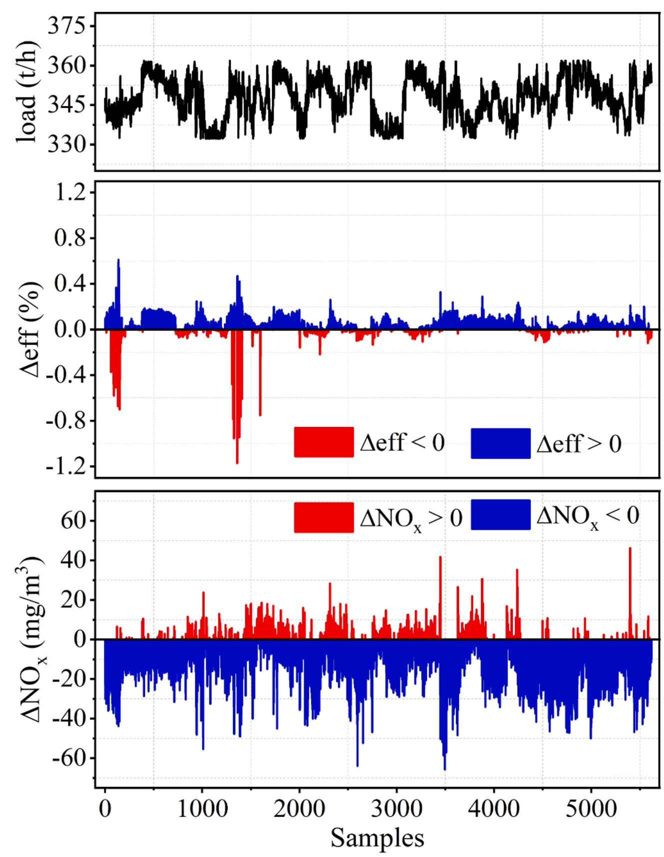

Fig. 14. Actual thermal efficiency and NO_x concentration change after policy optimization.

$$series_{t+1} = \begin{pmatrix} d_i^{t-step+1} & perf^{t-step+2} \\ \vdots & \vdots \\ d_i^{t-1} & perf^{t} \\ d_i^{t} & perf^{t+1} \end{pmatrix}, i \in [1,k] \tag{17}$$

$$static_{t+1} = (c_j^t(1+\Delta action_j^t), x_l^t), j \in [1,n], l \in [1,p] \tag{18}$$

where $\Delta action_j^t$ is adjustment range for adjustment parameter with upper and lower bounds, set to [-16.7 %,16.7 %] in the same way as GA.

The environment outputs the reward value corresponding to the

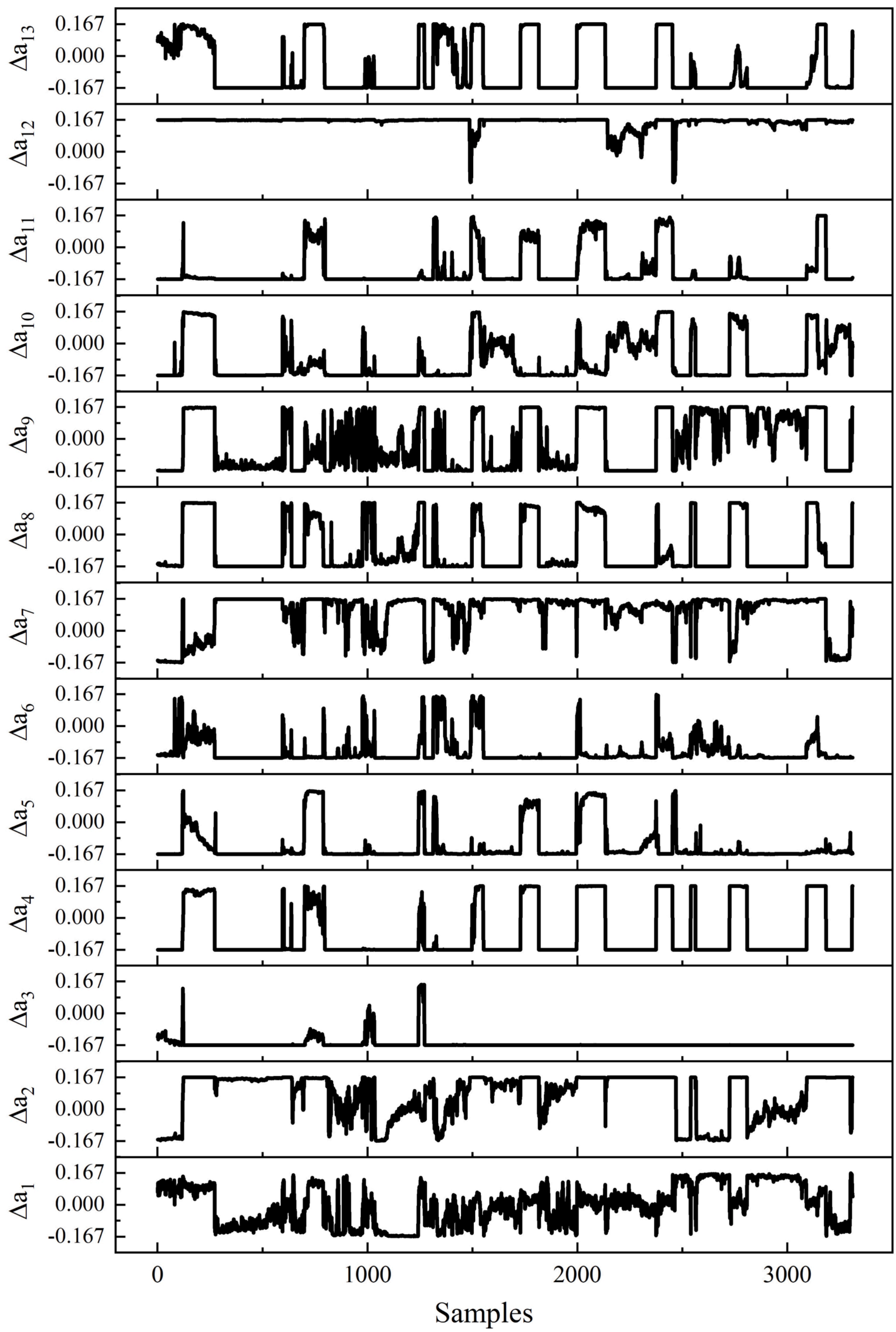

Fig. 15. Combustion adjustments proposed by the strategy for different test conditions.

Table 6b
Combustion adjustment scheme for maximum increase in thermal efficiency.

Δa_1	Δa_2	Δa_3	Δa_4	Δa_5	Δa_6	Δa_7	Δa_8	Δa_9	Δa_{10}	Δa_{11}	Δa_{12}	Δa_{13}
0.06	−0.14	−0.16	−0.16	−0.16	0.14	−0.16	−0.16	−0.16	−0.15	−0.16	0.16	0.16

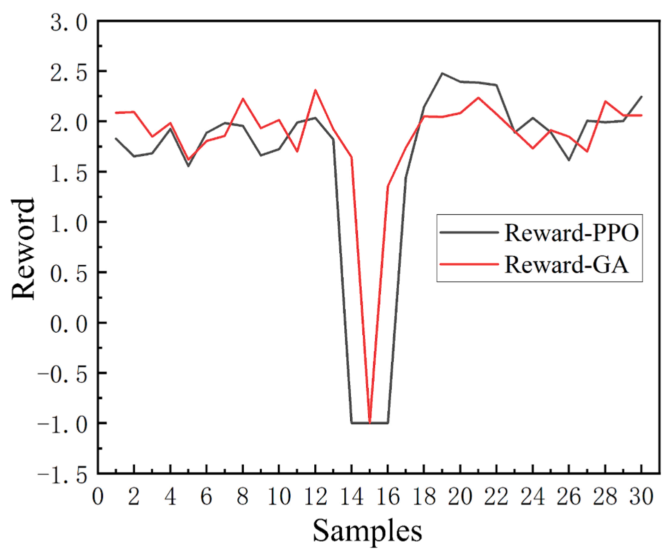

Fig. 16. Comparison of reward values obtained by the strategy and GA.

behavior based on the current state and the behavior derived from the PPO, and the reward function $R(e^{t+1}, e^t, n^{t+1}, n^t, action)$ is the same as GA fitness function built in the previous section.

4.1.2. PPO agent

Although PPO contains two Actor networks in principle, there is no difference between the two networks in terms of structure, only the difference of successive calls, so only one Actor network and one Critic network need to be built. The initialization of the Actor neural network is to complete the parameterization of the boiler combustion optimization strategy.

Actor network input is the state, the advantage function and the previous output value $action_{old}$, the output is the combustion adjustment scheme *action*, containing the adjustment magnitude of 13 combustion adjustment parameters $\Delta action_j$. the *Critic* network input is the state and the output is a state value function. Actor and Critic network structures are based on the constructed NO_x and thermal efficiency prediction models, and the network structures are shown in Table 6a. 1D_CNN and 1D_MaxPooling are used for dynamic features, and the sliding window size is 2; 1-layer Dense is used for static features. Finally, the Actor network output is activated using the tanh function so that the action is in the interval [-1, 1] and subsequently narrowed to [-1/6, 1/6]; the Critic network output has no activation function.

4.2. Offline policy training

The number of rounds of interaction between the algorithm and the environment *EPISODES* is 5 million, and the trajectory length obtained from each round of interaction *tra* is 4, so the number of interactions between the algorithm and the environment is 20 million. Before each policy training, the interaction experience is saved by buffer with a size of 2048 (*buffer_size*). Actor network and Critic network training epoch is 10, training size is 256 (*batch_size*), learning rate is 0.0001, and optimizer both use Adam algorithm. The hyperparameters of the other PPO algorithms are defined as in the original algorithm [30].

After the environment and the agent are built, the training of combustion optimization strategies is then carried out. The boiler combustion optimization strategy training is shown as follows.

Burning optimization strategy training algorithm

Initialize *episode* = 0;
While *episode* ≤ *EPISODES* do
Randomly initialize buffer, used to save *state*, current *action*, past action $action_{old}$ and discount rewards R_{dis};
Initialize $i = 0$;
While $i \le$ *buffer_szie* do
Randomly initialize $state_t$, then transition to $state_{t+1}$ according to Actor, and output rewards R_t; Multiple interactions until track length is reached, *episode* += 1;
Calculate the cumulative discounted rewards of the remaining trajectories at each moment R_{dis};
End while
The experiences in buffer are fed into the Critic and then calculate advantage function. Inputting all experiences and advantage functions into the Actor and Critic, calculating the gradient according to PPO algorithm, and Adam algorithm trains network parameters according to the gradient;
Recording Actor network losses and Critic network losses;
End while

4.3. Online optimizing simulation

4.3.1. Promotion of boiler performance

Fig. 14 shows thermal efficiency and NO_x increasement for all tested

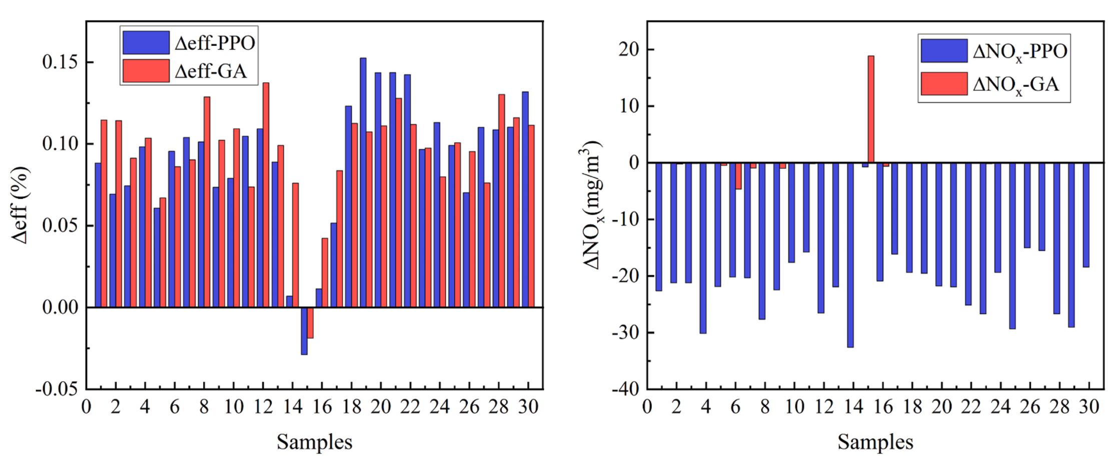

Fig. 17. Thermal efficiency and NO_x changes after optimization by strategy and GA.

conditions, and effective optimization percentage of thermal efficiency improvement and NO_x reduction ($\Delta eff > 0, \Delta NO_x < 0$) is 63.5 %. After the single-step combustion adjustment, thermal efficiency increasement ranges of 0–0.61 % and NO_x reduction ranges of 0–65 mg/m^3, and certain optimization can be achieved under different load conditions. It shows that the optimization of thermal efficiency and NO_x is better when the combustion optimization strategy is applied to new boiler operating conditions.

Fig. 15 shows the variation of the flue gas oxygen content and the adjustment magnitude of the 12 secondary damper openings corresponding to all the successfully optimized combustion conditions out of the 5619 test conditions. It can be seen that the magnitude of most of the adjustment amplitudes varies with different test cases, and only a few of the adjustment amplitudes reach the boundary, such as the secondary damper openings (layer D #3) and secondary damper openings (layer B #2), and almost remain the same in each test case. Since the sample of test conditions is large, it can be considered that the strategy is more effective in combustion optimization in a small range, and the proposed combustion adjustment scheme is not easy to be affected by the artificially set adjustment boundary.

In addition, individual combustion tuning scenarios are analyzed. For example, for the case with the greatest optimization of thermal efficiency, the combustion adjustment scheme is shown in Table 6b. It can be seen that$\Delta a_1 > 0$, indicating a slight increase in the total air volume; Δa_2 to$\Delta a_5 < 0$, indicating a decrease in the secondary air volume of layer D (#1-#4); layer C (#1-#4) secondary air volume is also reduced, but less compared to layer D; and the secondary air volume of layer B (#1-#4) is almost unchanged. At this time, the secondary air volume in the graded combustion area is reduced, and the oxygen concentration is reduced to inhibit the generation of NOx. At the same time, the proportion of secondary air volume in the lower level is still large, and the air distribution is still in a positive tower shape, thus favoring the combustion of coal dust. The increase of total air volume and the decrease of secondary air volume imply that the degree of pulverized coal combustion can be further improved by increasing the combustion air volume, thus improving the thermal efficiency. Therefore, to a certain extent, the combustion adjustment program derived from the algorithm is more in line with the actual combustion adjustment rule.

4.3.2. Comparison with GA

To compare the combustion optimization strategy based on the PPO algorithm with GA randomly selects 30 states from the test state set and perform combustion optimization for these conditions using GA, and couples the composite model, the results are as follows.

Fig. 16 shows that the combustion tuning schemes derived from the strategy and GA successfully optimized most of the 30 test conditions, and reward values obtained by combustion tuning schemes is not different, so it is difficult to distinguish between good and bad optimization effects.

As shown in the right of the Fig. 17, combustion optimization strategy based on PPO does not differ much from GA in terms of the degree of thermal efficiency optimization for the working conditions. For nearly half of the working conditions, the optimization effect is better after applying the combustion adjustment scheme proposed by the strategy, and the remaining half of the working conditions are better after applying the combustion adjustment scheme of GA. However, as shown in the right of the Fig. 16, the difference between the two for NO_x reduction is large, and the strategy is significantly stronger than GA for NO_x concentration reduction. With the same fitness function and reward function, GA is almost difficult to optimize for NO_x, and only ensures NO_x no growth.

Furthermore, in terms of decision speed, the combustion optimization strategy based on the PPO algorithm can be considered as an online combustion optimization strategy because it is pre-trained offline without further iterative operations, and the single decision is almost instantaneous with an average time of 0.015 s,but GA takes about 3 min on average for a single optimization, and it is still difficult to achieve online decision making for variable boiler operating conditions compared to the combustion optimization strategy based on the PPO algorithm.

5. Conclusion

In this paper, a combustion optimization study is carried out for a 410 t/h quadrangular cut-round pulverized coal boiler, and the dynamic prediction model of boiler thermal efficiency and NO_x is established by using LSTM and 1D_CNN neural network for the problem that the static model is limited to the optimization of steady-state conditions. To optimize the general dynamic model combined with the optimization algorithm to establish online combustion optimization algorithm, this paper classifies the model input features, couples the dynamic model with BP neural network, and establishes a static-dynamic composite prediction model. To address the problem of slow speed of traditional heuristic optimization algorithm which is difficult to meet the speed of online optimization, this paper couples the composite prediction model with PPO reinforcement learning algorithm to establish an online optimization strategy for boiler combustion. Through the experimental validation of 5619 test cases, the strategy successfully achieves 63.5 % co-optimization of NO_x and thermal efficiency, with thermal efficiency increase ranging from 0-0.61 % and NO_x reduction ranging from 0-65 mg/m^3. Meanwhile, comparing the optimization effect of the PPO algorithm with that of the GA shows that the PPO strategy has a more significant effect on NO_x reduction while keeping the thermal efficiency optimized. Moreover, the online decision-making speed of the PPO strategy is much higher than that of the GA, with an average time consumption of only 0.015 s, while the GA requires about 3 min for a single optimization. The offline trained strategy can be directly applied to real-time boiler operating conditions and output combustion optimization scheme online to guide operators in real time to optimize boiler thermal efficiency and NO_x emission.

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.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (No. 2022ZFJH004), Ningbo Science and Technology innovation 2025 Major Project (No. 2018B10024) and the Innovative Research Groups of the National Natural Science Foundation of China (No. 51621005).

References

[1] J.F. Tuttle, L.D. Blackburn, K. Andersson, et al., A systematic comparison of machine learning methods for modeling of dynamic processes applied to combustion emission rate modeling, Appl. Energy 292 (2021) 116886.
[2] S.A. Kalogirou, Artificial intelligence for the modeling and control of combustion processes: a review, Prog. Energy Combust. Sci. 29 (6) (2003) 515–566.
[3] P. Ilamathi, V. Selladurai, K. Balamurugan, et al., ANN–GA approach for predictive modeling and optimization of NO_x emission in a tangentially fired boiler, Clean Techn. Environ. Policy 15 (1) (2013) 125–131.
[4] Y. Shi, W. Zhong, X. Chen, et al., Combustion optimization of ultra supercritical boiler based on artificial intelligence, Energy 170 (2019) 804–817.
[5] G.-Q. Li, X.-B. Qi, K.C.C. Chan, et al., Deep Bidirectional Learning Machine for Predicting NO_x Emissions and Boiler Efficiency from a Coal-Fired Boiler, Energy Fuel 31 (10) (2017) 11471–11480.

[6] D. Adams, D.-H. Oh, D.-W. Kim, et al., Prediction of SO_x-NO_x emission from a coal-fired CFB power plant with machine learning: Plant data learned by deep neural network and least square support vector machine, J. Clean. Prod. 270 (2020).
[7] L. Zheng, H. Zhou, C. Wang, et al., Combining Support Vector Regression and Ant Colony Optimization to Reduce NO_x Emissions in Coal-Fired Utility Boilers, Energy Fuel 22 (2) (2008) 1034–1040.
[8] L.-G. Zheng, H. Zhou, K.-F. Cen, et al., A comparative study of optimization algorithms for low NO_x combustion modification at a coal-fired utility boiler, Expert Syst. Appl. 36 (2, Part 2) (2009) 2780–2793.
[9] Q. Li, Q. He, Z. Liu, Low NO_x combustion optimization based on partial dimension opposition-based learning particle swarm optimization, Fuel 310 (2022) 122352.
[10] Y. Ma, H. Wang, X. Zhang, et al., Three-objective optimization of boiler combustion process based on multi-objective teaching–learning based optimization algorithm and ameliorated extreme learning machine, Machine Learning with Applications 5 (2021) 100082.
[11] P. Tan, J. Xia, C. Zhang, et al., Modeling and reduction of NO_x emissions for a 700 MW coal-fired boiler with the advanced machine learning method, Energy 94 (2016) 672–679.
[12] J. Song, C.E. Romero, Z. Yao, et al., Improved artificial bee colony-based optimization of boiler combustion considering NOX emissions, heat rate and fly ash recycling for on-line applications, Fuel 172 (2016) 20–28.
[13] N. Li, Y. Hu, The Deep Convolutional Neural Network for NO_x Emission Prediction of a Coal-Fired Boiler, IEEE Access 8 (2020) 85912–85922.
[14] Y. Cheng, Y. Huang, B. Pang, et al., ThermalNet: A deep reinforcement learning-based combustion optimization system for coal-fired boiler, Eng. Appl. Artif. Intel. 74 (2018) 303–311.
[15] B.H. Peng Tan, C. Zhang, D. Rao, S. Li, Q. Fang, G. Chen, Dynamic Modeling of NO_x Emission In a 660 Mw Coal-fired Boiler With Long Short-term Memory, Energy 176 (2019) 429–436.
[16] H. Pan, T. Su, X. Huang, et al., LSTM-based soft sensor design for oxygen content of flue gas in coal-fired power plant, Trans. Inst. Meas. Control 43 (1) (2021) 78–87.
[17] Y.W. Guotian Yang, X. Li, Prediction of the NO_x Emissions From Thermal Power Plant Using Long-short Term Memory Neural Network, Energy (2020) 192.
[18] Zhan X, Xu H, Zhang Y, et al. DeepThermal: Combustion Optimization for Thermal Power Generating Units Using Offline Reinforcement Learning[J], 2021.
[19] Q. Li, G. Yao, Improved coal combustion optimization model based on load balance and coal qualities, Energy 132 (2017) 204–212.
[20] W. Zheng, C. Wang, Y. Yang, et al., Multi-objective combustion optimization based on data-driven hybrid strategy, Energy 191 (2020) 116478.
[21] Y. Niu, J. Kang, F. Li, et al., Case-based reasoning based on grey-relational theory for the optimization of boiler combustion systems, ISA Trans. 103 (2020) 166–176.
[22] D. Adams, D.-H. Oh, D.-W. Kim, et al., Deep reinforcement learning optimization framework for a power generation plant considering performance and environmental issues, J. Clean. Prod. 291 (2021) 125915.
[23] Bao X, Li Y, Zhu C. Deep deterministic policy gradient-based combustion optimization method for coal-fired boiler[C]. 2021 40th Chinese Control Conference (CCC), 2021: 6255-6260.
[24] Zou L, Cheng Y, Zhuang Z, et al. An Optimization Control of Thermal Power Combustion Based on Reinforcement Learning[C]. 2018 37th Chinese Control Conference (CCC), 2018: 3553-3558.
[25] Samsuden M A, Diah N M, Rahman N A. A Review Paper on Implementing Reinforcement Learning Technique in Optimising Games Performance[C]. 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 2019: 258-263.
[26] H. Nguyen, H. La, Review of Deep Reinforcement Learning for Robot Manipulation [C], Third IEEE International Conference on Robotic Computing (IRC) 2019 (2019) 590–595.
[27] D. Cao, W. Hu, J. Zhao, et al., Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review, J. Mod Power Syst. Clean Energy 8 (6) (2020) 1029–1042.
[28] Z. Zhang, D. Zhang, R.C. Qiu, Deep reinforcement learning for power system applications: An overview, CSEE J. Power Energy Syst 6 (1) (2020) 213–225.
[29] M. Wiering, M. Van Otterlo, Reinforcement learning, Adaptation, Learning, and Optimization 12 (2012).
[30] S. Kiranyaz, O. Avci, O. Abdeljaber, et al., 1D convolutional neural networks and applications: A survey, Mech. Syst. Sig. Process. 151 (2021) 107398.
[31] Schulman J, Wolski F, Dhariwal P, et al. Proximal Policy Optimization Algorithms [J], 2017.
[32] Z. Tang, Z. Zhang, The multi-objective optimization of combustion system operations based on deep data-driven models, Energy 182 (2019) 37–47.
[33] J.H. Cheong, Four ways to quantify synchrony between time series data[DB/OL]. Towards, Data Science (2020), https://doi.org/10.17605/OSF.IO/BA3NY.
[34] C. Shen, Analysis of detrended time-lagged cross-correlation between two nonstationary time series, Phys. Lett. A 379 (7) (2015) 680–687.
[35] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.