



# Multi-agent collaborative control parameter prediction for intelligent precision loading

Zihua Chen<sup>1</sup> · Chuanli Wang<sup>1</sup> · Jingzhao Li<sup>2</sup> · Shunxiang Zhang<sup>2</sup> · Qichun Ouyang<sup>3</sup>

Accepted: 25 January 2022 / Published online: 19 March 2022  
© The Author(s) 2022

## Abstract

Due to the low adjustment accuracy of manual prediction, conventional programmable logic controller systems can easily lead to inaccurate and unpredictable load problems. The existing multi-agent systems based on various deep learning models has weak ability for advanced multi-parameter prediction while mainly focusing on the underlying communication consensus. To solve this problem, we propose a hybrid model based on a temporal convolutional network with the feature crossover method and light gradient boosting decision trees (called TCN-LightGBDT). First, we select the initial dataset according to the loading parameters' tolerance range and supply supplementing method for the deviated data. Second, we use the temporal convolutional network to extract the hidden data features in virtual loading areas. Further, a two-dimensional feature matrix is reconstructed through the feature crossover method. Third, we combine these features with basic historical features as the input of the light gradient boosting decision trees to predict the adjustment values of different combinations. Finally, we compare the proposed model with other related deep learning models, and the experimental results show that our model can accurately predict parameter values.

**Keywords** Industrial loading · Dilated convolution · Gradient boosting decision tree · Parameters prediction

## 1 Introduction

Industrial loading is a field to achieve accurately quantitative loading for materials, which is widely applied in agriculture, mining, etc., and the conventional loading process based on this system is shown in Fig. 1, which includes a rough loading process and a precise loading process. The existing works for accomplishing the goal mainly utilize the conventional manual programmable logic controller system [1, 2]. The loading capacity is judged by the accumulation height in the rough loading process. When both the front and rear wheels of the truck are on the track scale, the precise loading process uses on the indicating weights of the track scale to

achieve the loading. The above loading process needs to stop midway and repeatedly reload to achieve the target quantity. However, inaccurate and dangerous unbalanced loading problems will often occur in actual scenes because of fuzzy and artificial experience prediction for multi-adjustment values. Thus, how to obtain accurate adjustment parameter values with actual applications has become the key issue for exploration in the paper.

Nowadays, the collaborative control process based on multi-agent systems (MAS) has been studied in many fields. Some authors integrate the multi-agent system with machine learning models [3]. However, the existing prediction models mainly solve the communication consensus problem [4–6], and they cannot be well fitted for the advanced multi-parameter prediction. In a word, it is a challenge that what collaborative control multi-parameter prediction model in MAS should be provided and the relevant loading parameters' standards can cater to the accurate prediction model well.

In recent years, we have found that the hybrid machine learning model can dynamically simulate artificial experience and achieve accurate target prediction [7, 8]. Deep

✉ Chuanli Wang  
chuanliwangaust@163.com

<sup>1</sup> School of Mechanical Engineering, Anhui University of Science and Technology, Huainan 232001, China

<sup>2</sup> School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China

<sup>3</sup> Huaibei Mining Co., Ltd, Huai'bei 235000, China

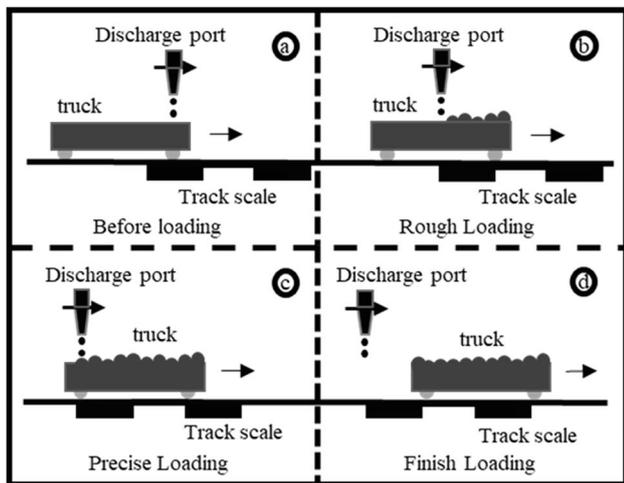


Fig. 1 The dynamic loading process of the conventional method

learning, particularly convolutional neural networks, provided a new perceptive field for feature extraction/learning by parallel convolution of multi-layer filters [9–12]. Further, the latest temporal convolutional network (TCN) that as an alternative model for sequence modeling is extensively used in many fields, such as pattern recognition [13] and signal prediction [14]. The TCN integrates both the feature convolution processing of the convolutional neural networks [9] and the time-series information mining capability of the recurrent neural network [15]. It is suitable for parallel and distributed computing for massive amounts of highly non-linear dynamic process data, making it very popular for data feature extraction. In addition, the gradient boosting decision tree (GBDT) has received a lot of attention, which is adopted as the prediction layer in hybrid learning models. This algorithm is an optimized form of gradient boosting, which has the characteristics of high accuracy, fast convergence, and easy cache optimization. Also, because of the time-series threshold, the GBDT can conduct a nonlinear relationship model for the multi-output prediction [16]. It is suitable for

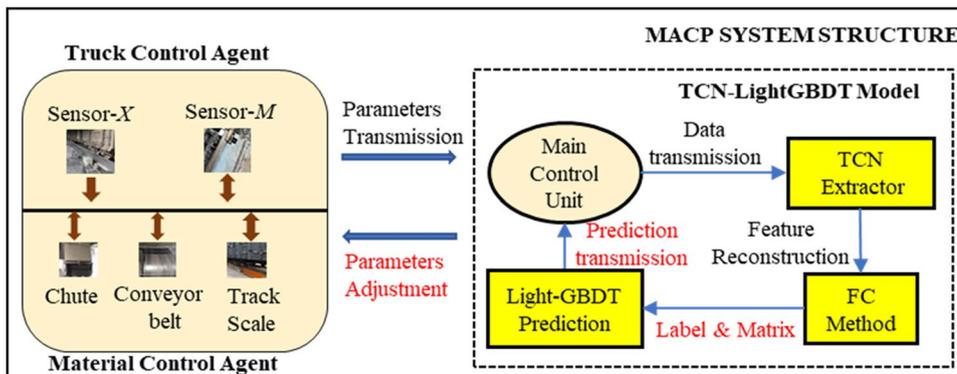
the adjustment values prediction of the multi-parameter in an industrial loading process.

In order to achieve collaborative control multi-parameter prediction, this paper proposes a parallel TCN-LightGBDT model applied in a collaborative control parameters' prediction MAS (MACP), which is shown in Fig. 2. The novelty of this work is that the proposed TCN-LightGBDT model integrates a wide receptive field and cross-layer information transmission for the TCN and negative-gradient learner for the GBDT. We also propose a new theoretical parameter supplementing method and parameter selected deviations for dataset construction. The experimental results show that the proposed model achieves a significant and reasonable improvement compared to the baseline models. The main contributions of this paper are as follows:

- (1) The parameter-selected deviations are formulated to solve the low-precision prediction problem of the multi-parameter in the irregular loading. We propose a theoretical parameter supplementing method to complete the deviated data in the processed dataset and improve the extraction capability for features' fluctuation trends.
- (2) We adopt the TCN to extract the deep time-domain features parallelly. With the help of the feature crossover (FC) method, the two-dimensional feature matrix of parameters is reconstructed, which will be used as a key input in the Light-GBDT model. Further, the Light-GBDT model [17, 18] is applied to the prediction of multi-parameter adjustment values.

The rest of this paper is organized as follows. Section 2 introduces the relative work of multi-agent systems and deep learning models for dynamic target prediction. Section 3 proposes the principle of the TCN-LightGBDT and relevant data processing methods. Section 4 gives the experimental results and the theoretical analysis. Finally, the conclusion and future work are given in Section 5.

Fig. 2 The structure of the MACP System (i.e., The system is used to transmit the parameter data and adjust agents according to the predicted adjustment values in industrial loading. The agents in the MACP system conclude two parts: the truck control agent and the material control agent.)



## 2 Related work

We review the related research work in three main areas in this paper, including: (1) The MAS control systems in relative industrial fields; (2) The target predictive models using neural networks; (3) The optimization methods using the expansive decision tree algorithm.

### 2.1 The MAS systems in relative industrial fields

The existing MASs mainly focus on the effective state consensus of agents [19, 20], the low multi-layer continuous communication costs [21, 22], and the adaptive collaborative control method [12, 23, 24]. For example, Z. Xu, et al. [19] propose the edge event triggering technique to eliminate the Zeon behavior and reduce the burden on the event detector. Y. Han, et al. [20] design an encoding–decoding impulsive protocol to achieve energy constraints in MAS. L. Lindemann, et al. [21] provide a hybrid feedback control strategy based on the time-varying control barrier function. F. Lian, et al. [22] provide sparsity-constrained distributed social optimization and non-cooperative game algorithms to save the cost of the underlying communication network. Y. Qian, et al. [23] adopt distributed event-triggered adaptive output feedback control strategy to solve the control problem of linear multi-agent systems. S. Luo, et al. [12] propose a distributed event-triggered adaptive feedback control strategy to process the consensus problem of external disturbances in MAS. H. Tan, et al. [24] solve the coordination of cloud-based model-free multi-agent systems with communication constraints by the distributed predictive control method.

### 2.2 The target predictive model using neural networks

The target predictive model based on neural networks has proved its success in the target parameters tracking process [7, 8, 25, 26] and nonlinear mechanical collaborative system control [27–30]. A Agga, et al. [7] and T. Bao, et al. [8] present the convolutional neural network and long short-term memory model to predict time-series data. A visual object tracking collaborative architecture based on the convolutional neural network is provided by W. Tian, et al. [25]. Additionally, J. Song, et al., [26] propose a heat load prediction model based on a temporal convolutional neural network. W. He, et al. [27] propose a disturbance observer-based radial basis function neural network control scheme. Z. Wang, et al. [28] proposed a radial basis function neural network control scheme based on disturbance observer.. S. Gehrman, et al. [29] provide a framework of the visual interface collaborative semantic inference for the decision

processes. H. Wang, et al. [30] present an intelligent coordinated control system for the dynamic monitoring of the heavy scraper conveyor.

### 2.3 The optimization methods using expansive decision tree algorithm

The decision tree algorithm and its expansion belong to the machine learning methods widely applied in data classification, regression, and prediction. T. Wang, et al. [31] and L. Wang, et al. [32] integrate the random forest to achieve accurate prediction or classification problems. R. Sun et al. [33] propose a GBDT-based method to predict the pseudo-range errors by considering the relevant signal strength and satellite elevation angle. D. Thai, et al. [34] propose an approach based on a gradient boosting machine to predict the local damage data of reinforced concrete panels under impact loading. L. Lu, et al. [35] propose an LSTM-Light Gradient Boosting Machine model that can predict latency quickly and accurately based on the collected dataset. Y. Dan, et al. [36] combine the deep CNN model with GBDT for the superconductor's critical temperature accurate prediction. H. Kong, et al. [37] propose a risk prediction model based on the combination of Logistic and GBDT. J. Bi, et al. [38] propose a new hybrid prediction method, which combines the capabilities of the temporal convolutional neural network and the LSTM to predict network traffic.

To the best of our knowledge, the existing work on multi-parameter parallel prediction have been less studied in industrial loading fields. Thus, this paper explores the hybrid model to achieve accurate multi-parameter prediction. The detailed structure of the proposed TCN-LightGBDT model is introduced in Section 3.

## 3 The structure of the TCN-LightGBDT prediction model

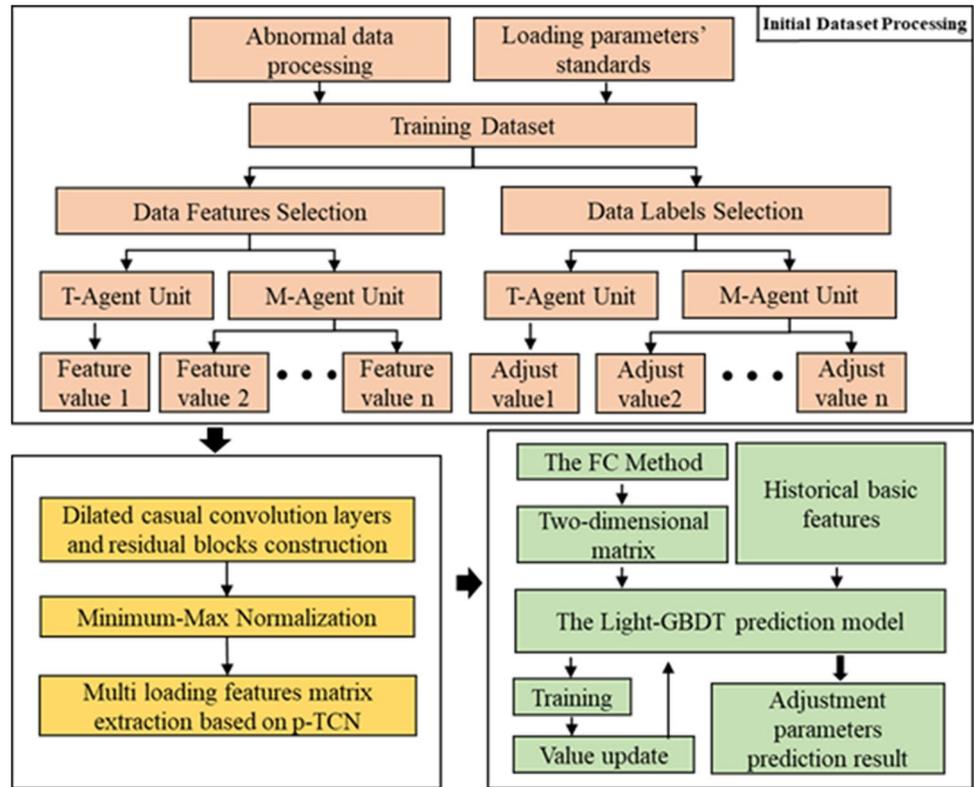
The TCN-LightGBDT prediction model consists of two parts: the multi-parameter's feature extraction based on the TCN and the Light-GBDT optimized prediction. Each framework of the proposed model is designed in Fig. 3.

### 3.1 Multi-parameter's feature extraction based on the TCN

The feature extraction model based on the TCN includes exception data processing, theoretical parameters preparation, and multi-parameter matrix extraction.

- (1) Exception data processing  
Because of the accuracy error and manual operation interference, the raw dataset usually has many low-

**Fig. 3** The framework of the TCN-LightGBDT model



quality data (i.e., missing and over-precision data). The processing methods (as well as their acquisition accuracy) that are used to deal with this problem are described in Table 1. The labels in Table 1 conclude three parts: the speed adjustment value, the flow adjustment value, and the inclination adjustment value.

(2) Theoretical parameter specification

In this section, we propose the method to calculate the theoretical parameter value, and the details are as follows:

a) Let  $L_T, L_B, M_T$ , and  $m_0$  represent the truck length, the truck wheelbase, standard load, and the empty truck weight, respectively.  $n$  presents the loading area number of the truck, and  $r$  represents the loading areas within the horizontal distance from the truck rear baffle to the front wheel. Additionally,  $V = \{v_1, v_2, \dots, v_r, \dots, v_n\}$  denotes the truck target speed of each virtual loading area.  $Q = \{q_1, q_2, \dots, q_r, \dots, q_n\}$  is the belt conveyor flow in each loading area.  $C = \{c_1, c_2, \dots, c_r, \dots, c_n\}$ ,  $c \in (0, 90]$  is the chute inclination angle in each vir-

**Table 1** The low-quality data processing methods

Type	Name	Symbol	Processing Precision	Processing Method
Feature1	Time	$T$	2 decimal places	using theoretical parameter supplementing method
	Displacement	$D$	3 decimal places	
	Truck Speed	$V$	4 decimal places,	
Feature2	Actual Load	$M_A$	3 decimal places	using theoretical parameter supplementing method
	Target Load	$M_T$	3 decimal places	
	Belt Conveyor Flow	$Q$	4 decimal places	
	Chute Inclination	$C$	2 decimal places	
	Height Error	$H$	3 decimal places	
Labels	Adjustment_Speed	$\Delta v$	4 decimal places	using theoretical parameter supplementing method or continuous period variance instead
	Adjustment_Inclination	$\Delta c$	2 decimal places	
	Adjustment_Flow	$\Delta q$	4 decimal places	



where  $H_i$  is the actual loading height of each loading point.  $H_{target}$  is the target loading height.  $\Delta v_i$ ,  $\Delta q_i$ ,  $\Delta c_i$  respectively represent the  $i$ -th speed adjustment, belt conveyor flow adjustment, and the chute inclination adjustment value.

(3) Multi-parameter matrix extraction based on TCN

We denote the pre-input of the parallel TCN neural network as  $\tilde{X} = [V, Q, C, D, T, M_T, M_A, H]$ . Then, the max-min range normalization method is described as follows.

$$MaxRange = |\tilde{X}_{max} - \tilde{X}_{min}| \tag{9}$$

$$X_{ii} = (\tilde{X}_{ii} - \tilde{X}_{min}) / MaxRange, X_{ii} \in X \tag{10}$$

where  $X = [V', Q', C', D', T', M'_A, H']$  is the standard dataset of the input layer,  $X_{ii}$  is an element of the standard dataset.  $\tilde{X}_{ii}$  is an element of the dataset  $\tilde{X}$ ,  $\tilde{X}_{min}$  is the minimum value and  $\tilde{X}_{max}$  is the maximum value in the dataset  $\tilde{X}$ .

The dilated causal convolution of the TCN can perform convolution expansion on the input and solve the problem of limited receptive fields, which is shown in Fig. 5. For the one-dimensional features  $X = (x'_0, x'_1, x'_2, \dots, x'_t, \dots, x'_T)$  and the filters  $D_f = \{f_1, f_2, \dots, f_D\}$ , the dilated convolution operation  $F(\bullet)$  of each element  $B$  is defined in Formula (10).

$$F(T) = (X *_d f_{ii})(T) = \sum_{i=0}^{n-1} f(i) * x_{B-d \bullet i} \tag{11}$$

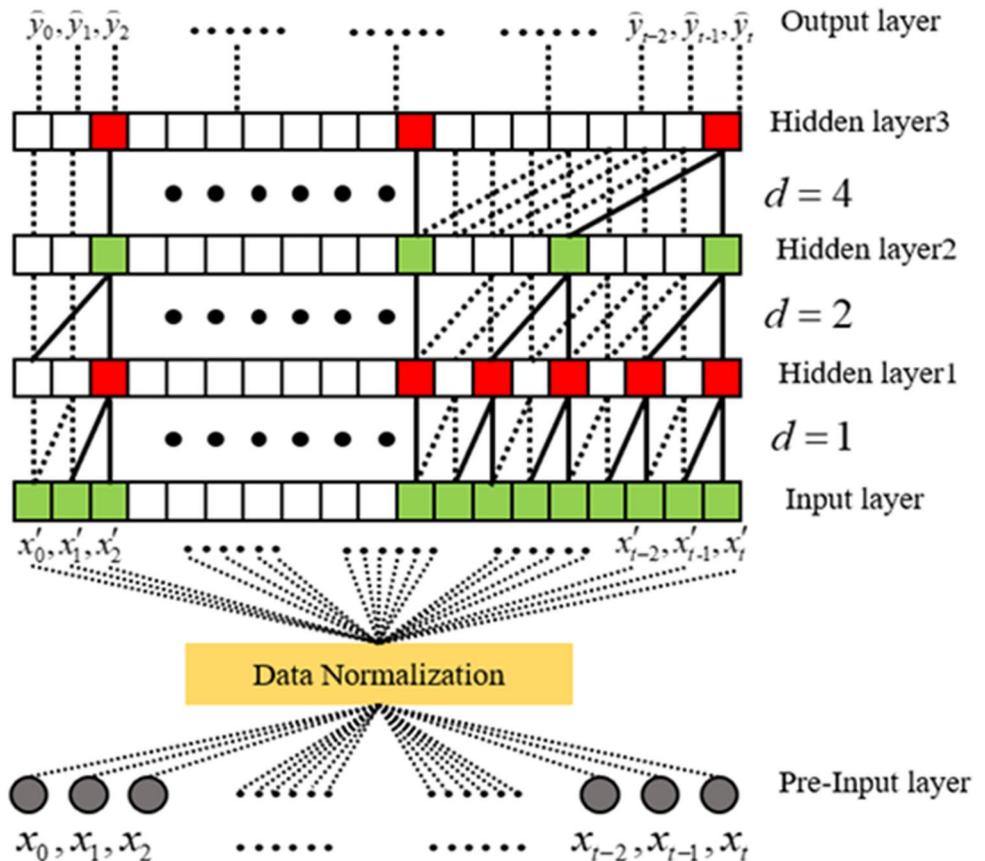
$$\omega = 1 + \sum_{i=0}^{m-1} (k - 1) * d^i = 1 + (k - 1) * \frac{d^m - 1}{d - 1} \tag{12}$$

where  $n$  denotes the filter size,  $d$  represents the dilated factor,  $B-d \bullet i$  is the direction of the past,  $\omega$  indicates the width of the receptive field,  $k$  is the kernel size, and  $m$  is the number of the network layers.

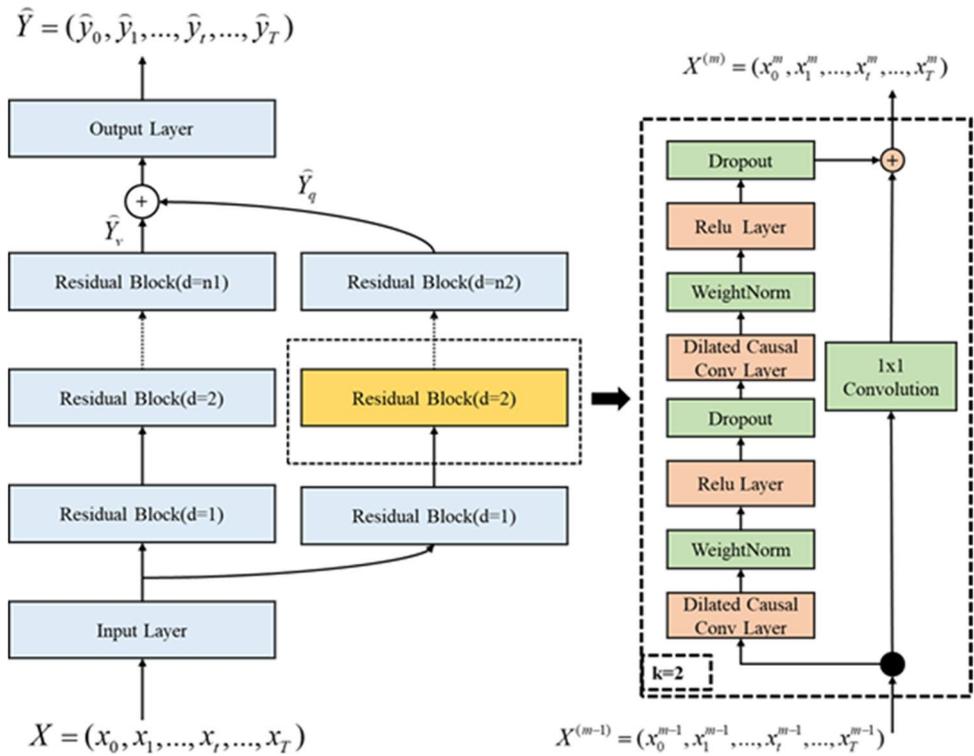
In addition, the increasing number of hidden layers will affect the deep network stability and complexity. We use the multi-residual blocks connection [39] with different dilated factors to solve the problem. The detailed structure of the multi-residual blocks is shown in Fig. 6, and the output  $X^m$  is denoted in Formula (13). When the residual connection operations are completed, we can get a two-dimensional matrix as the convolutional feature output  $(\hat{Y}_v, \hat{Y}_q / \hat{Y}_c)$  in Formula (14). The related multiplication factors are denoted in Formula (15).

$$X^m = \psi_{Relu}(F(X^{m-1}) + X^{m-1}) \tag{13}$$

Fig. 5 The dilated causal convolution stack diagram



**Fig. 6** The detailed structure of the multi residual blocks



$$\begin{aligned} \widehat{Y} &= (\widehat{y}_{0,T}, \widehat{y}_{1,T}, \widehat{y}_{2,T}, \dots, \widehat{y}_{M,T}) \\ &= \widehat{Y}_{q/c} + \widehat{Y}_v \\ &= \delta_1 \cdot (\widehat{y}_{0,T}^v, \widehat{y}_{1,T}^v, \widehat{y}_{2,T}^v, \dots, \widehat{y}_{M,T}^v) + \delta_2 \cdot (\widehat{y}_{0,T}^{q/c}, \widehat{y}_{1,T}^{q/c}, \widehat{y}_{2,T}^{q/c}, \dots, \widehat{y}_{M,T}^{q/c}) \end{aligned} \tag{14}$$

$$\delta_1 = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}^T, \delta_2 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}^T \tag{15}$$

where  $\psi_{Relu}(\cdot)$  is an activation operation,  $X^{m-1}$  is (m-1)-th input of residual block connection,  $R$  is the number of filters, both  $\delta_1$  and  $\delta_2$  represent matrix multiplication factors.

Notably, the FC method is adopted to explore further and synthesize the extracted features of the convolutional output matrix  $\widehat{y}$ . The FC method can average the hidden relationship and quantify the characteristics among different parameters, which is shown in Fig. 7. First, the FC method swaps the elements corresponding to the same subscript of the column vectors in the two-dimensional matrix. Second, we calculate the average values of each new column vector as the relative elements and reshape as a restructuring multi-parameter extraction matrix ( $\widehat{y}'$ ). The process is described in Formula (16). Finally, the output matrix  $O = (O_{0,T}, O_{1,T}, \dots, O_{R,T})$  shown in Formula (17) will be used as the Light-GBDT model's input to predict the suitable parameters' adjustment value.

$$\widehat{y}_{i,T}^v \Leftrightarrow \widehat{y}_{i,T}^{q/c} \Rightarrow \begin{cases} \widehat{y}_{i,T}^{v'} = \text{average}(\widehat{Y}_v(\widehat{y}_{i,T}^{q/c})) \\ \widehat{y}_{i,T}^{q'/c'} = \text{average}(\widehat{Y}_{q/c}(\widehat{y}_{i,T}^v)) \end{cases}, i \in [1, M] \tag{16}$$

$$O = (\widehat{Y}'_{v,q/c}) = \begin{bmatrix} \widehat{y}_{0,T}^{v'} & \widehat{y}_{1,T}^{v'} & \dots & \widehat{y}_{M,T}^{v'} \\ \widehat{y}_{0,T}^{q'/c'} & \widehat{y}_{1,T}^{q'/c'} & \dots & \widehat{y}_{M,T}^{q'/c'} \\ \widehat{y}_{0,T} & \widehat{y}_{1,T} & \dots & \widehat{y}_{M,T} \end{bmatrix} \tag{17}$$

where  $\widehat{y}_{i,T}^{v'}$  is the speed element of the output matrix  $O$ ,  $\widehat{y}_{i,T}^{q'/c'}$  is the flow or inclination element of the output matrix  $O$ , and  $\text{average}(\bullet)$  denotes the average value function.

### 3.2 The Light-GBDT optimized prediction

The GBDT is a gradient boosting framework based on a regression decision cart tree, which is applied to the feature regression by selecting the best split point. Normally, the final output in the GBDT is the sum of the results of all regression decision trees. The detailed structure of the GBDT is designed in Fig. 8. This paper utilizes the Light-GBDT to effectively deal with nonlinear and low-dimensional features to regressively predict adjustment values of the multi-parameter.

The multi-parameter extraction matrix and basic historical features of industrial loading are combined as the input dataset ( $Input_{gbd}$ ), which is described in Formula (18).

$$Input_{gbd} = X_g = [T', V', Q', C', M'_A, O] \tag{18}$$

Fig. 7 The FC Method Schematic Diagram

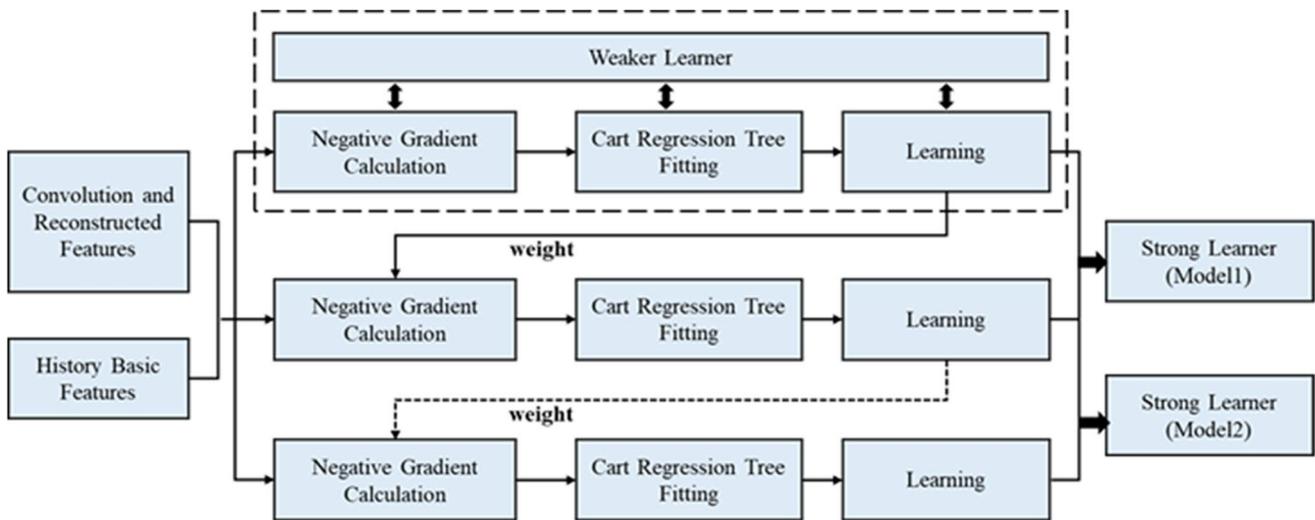
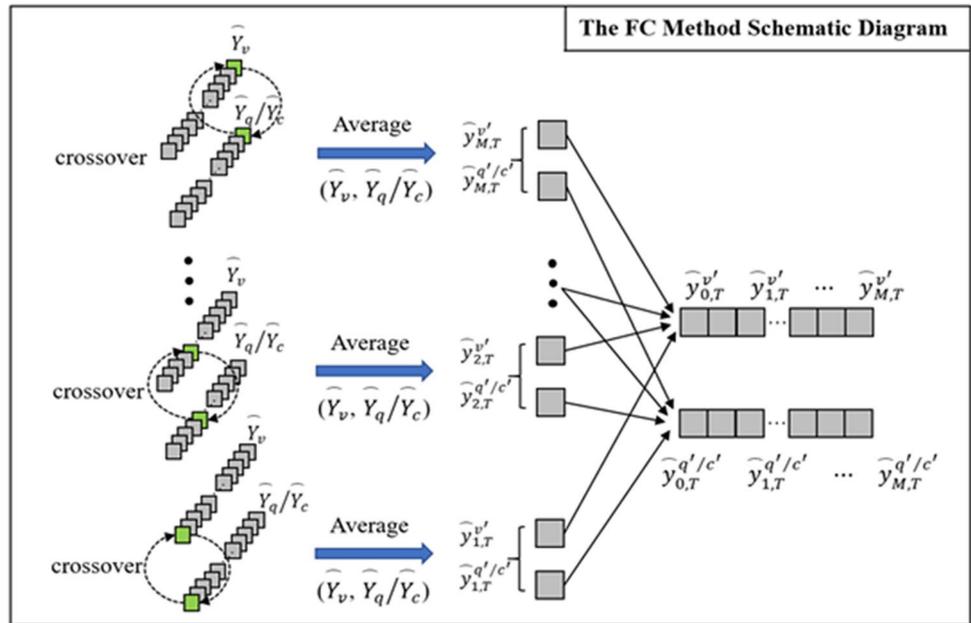


Fig. 8 The structure of the Light-GBDT model

Suppose the sample data input sequence is  $[X_g, O_g] = [(x_{g1}, o_{g1}), (x_{g2}, o_{g2}), \dots, (x_{gN}, o_{gN})]$ , where  $N$  is the number of the dataset collected samples, and  $o_{gi}$  ( $i = 1, 2, \dots, N$ ) denotes the actual value of the adjustment elements in data samples. The initial weak learner  $f_0(x_g)$  in Formula (19) is used to minimize the initial loss function  $L(y_i, c)$ . We choose the information gain as an index to evaluate the split candidate-point from all feature values. In addition, the gradient descent method is adopted to approximate the calculation because the greedy algorithm cannot be accurate in selecting the optimal basis function. For the training sample

$i$  of the  $m$ -th iteration, the negative gradient  $\gamma_{m,i}$  is calculated by Formula (20), and the gains after splitting each leaf node is described in Formula (21).

$$f_0(x_g) = \arg \min_c \sum_{i=1}^N L(o_{gi}, c) \tag{19}$$

$$\gamma_{m,j} = -\left[ \frac{\partial L(o_{gi}, f(x_{gi}))}{\partial f(x_{gi})} \right]_{f(x)=f_{m-1}(x)} \tag{20}$$

$$Gain = \frac{1}{2} \left[ \left( \frac{G_L^2}{H_L + \lambda} \right) + \left( \frac{G_R^2}{H_R + \lambda} \right) - \left( \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) \right] - \gamma \tag{21}$$

when we adopt the square variance function, the loss expression  $L(o_{gi}, f(x))$  is  $(o_{gi}, f(x))^2/2$ . If the absolute loss function is, the loss expression  $L(o_{gi}, f(x))$  is  $|o_{gi}, f(x)|$ , where  $m = (1, 2, \dots)$  denotes the number of iterations.  $G_{L,R} = \sum_{i \in leaf_{ij}} q_i$  denotes the first derivative of the loss function in the  $i$ -th sample of the  $j$ -th leaf node.  $H_{L,R} = \sum_{i \in leaf_{ij}} (q_i)^{-1}$  denotes the second derivative sum of the loss function.  $\gamma$  represents the penalties for the increased complexity of trees.

Furthermore, by fitting the residual value with the regression tree, the leaf node area of the  $m$ -th decision tree can be represented as  $\mathfrak{R}_{m,j}, j = 1, 2, \dots, J$ . The minimal residual loss value of a leaf node  $c_{m,j}$  is calculated for the  $j = 1, 2, \dots, J$  is described in Formula (22). The value of the whole decision tree is shown in Formula (23), and the calculation formula of update learner is shown in Formula (24).

$$c_{m,j} = \arg \min_c \sum_{x_i \in \mathfrak{R}_{m,j}} L(o_{gi}, f_{m-1}(x_{gi}) + c) \tag{22}$$

$$h_m(x) = \sum_{i=1}^{|leaf|_m=J} c_{m,j} I(x \in leaf_{m,j}) \tag{23}$$

$$f_m(x_g) = f_{m-1}(x_g) + v \cdot \sum_{j=1}^J c_{m,j} I(x_g \in \mathfrak{R}_{m,j}) \tag{24}$$

where  $h_m(\cdot)$  denotes the value of the  $m$ -th decision tree and we have  $x \in leaf_{m,j}, I = 1$ ; else  $I = 0$ . In addition,  $f_m(\cdot)$  denotes the updated learner's value of the  $m$ -th decision tree,  $v$  represents the scaling factor, and  $x_g$  is a vector element of the input dataset  $X_g$ .

The Light-GBDT prediction can be expressed as a combination of multi-decision trees, and the final output functions are shown in Formula (25) and (26).

$$F_M(x_g) = F_0 + v_1 F_1(x_g) + v_2 F_2(x_g) + \dots + v_T F_T(x_g) \tag{25}$$

$$\Rightarrow \widehat{F}_{output}(O_g) = f_0(x_g) + \sum_{m=1}^T [v_m \cdot \sum_{j=1}^J c_{m,j} I(x_g \in \mathfrak{R}_{m,j})] \tag{26}$$

where  $F_M(\cdot)$  denotes a combined output of all decision trees.  $v_1, v_2, \dots, v_T$  is the weight of each tree,  $T$  is the number of the trees.  $F_i(\cdot)$  denotes the weighted sum of optimal basis  $f_m(\cdot)$ .  $\widehat{F}_{output}(O_g)$  is the final output of the Light-GBDT.

Here, the time complexity of the process is  $O(KB)$ ,  $K$  is the neural network training epoch,  $B$  is the number of data in the training dataset. If the initial sample of the Light-GBDT is  $N$ , the time complexity of the training parameters dataset

is  $O(NJM)$ . In a word, the time complexity of the proposed model is  $O(KB + NJM)$ .

## 4 Experiments

### 4.1 The experimental settings and performance metrics

This paper collects a real loading dataset from the coal mine in Anhui Province, China. In the process of collaborative control parameter prediction for industrial coal loading, we have taken the whole carriage as a single research object, and the related historical data collected from Mar 1<sup>st</sup>, 2020 to Nov 1<sup>st</sup>, 2020 are applied to carry out experiments. In addition, the dataset selected deviations are presented in Table 2.

The proposed model and other baseline models are studied in this paper. These include two kinds of models, the classical learning models (i.e., the Light-GBDT [33], the Light-GBM [34], the TCN [26]) and the hybrid learning models (i.e., the CNN-LSTM [7], the TCN-LSTM [38], the LSTM-LightGBM [35], the TCN-CNN, the CNN-LightGBDT [36], and the TCN-LightGBDT). The experimental programming environment is Python 3.8, the Intel Core i7-9700 k CPU, and the 16 GB of memory.

The mean absolute error (MAE) represents the average absolute error between actual and prediction values. The gradient of mean square error (MSE) will change with the loss value. The Mean Absolute Percentage Error (MAPE) expresses the prediction percentage accuracy.  $R^2$  represents the coefficient of determination, which represents the interpretation of the independent variable to the dependent variable in the regression analysis, and the value range is (0, 1]. Namely, the larger the coefficient is, the closer the predicted value is to the real value. The evaluation metrics are defined in Formula (26), (27), (28), and (29).

$$RMSE = \sqrt{\frac{1}{N} \sum_{g=1}^N (F_g - \widehat{F}_g)^2} \tag{26}$$

$$FMAE = \frac{1}{N} \sum_{g=1}^N |F_g - \widehat{F}_g| \tag{27}$$

**Table 2** The dataset selected deviations of parameters

Adjustment Parameters	Selected deviations
Time(s)	240 ± 10.00
Load_value(t)	70 ± 1.000
Height_error(m)	± 0.200
Area-load_error(t)	± 0.200
Average_area_loading(t)	0.593 ± 0.100

$$FMAPE = \frac{1}{N} \sum_{g=1}^N \left| \frac{F_g - \hat{F}_g}{F_g} \right| \times 100\% \quad (28)$$

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{\omega=1}^N (\hat{Y}_\omega - Y_\omega)^2}{\sum_{\omega=1}^N (\bar{Y} - Y_\omega)^2} \quad (29)$$

where  $N$  denotes the number of testing instances,  $F_g$  and  $\hat{F}_g$  represent the actual and predicted adjustment value of the parameters in  $g$ -th instance, respectively.

## 4.2 Prediction for loading collaborative control adjustment parameters

In this experiment, the history loading data collected from Mar 1<sup>st</sup>, 2020 to Oct 1<sup>st</sup>, 2020 are employed to train simulation models. To better predict the dynamic collaborative control parameters, we randomly select the completed loading data are adopted as the testing instances. The prediction experiments for the adjustment values of parameters are as follows:

- 1) Experiment 1: Adjustment value prediction of truck speed and belt conveyor flow

Based on the above-listed experimental environment and data splitting rules, we select the continuous 97 front-loading area data to verify the model prediction effect. The parameters of each model are summarized as follows.

- (1) TCN: The temporal convolution network is built by the Keras library. The dilated convolution factors are [1, 2, 4, 8], the filters are 128/64/32/16, and the convolutional kernel size is 3.
- (2) Light-GBDT: The number of trees is 500, the maximum depth is 6, the model learning rate is 0.1, and the split criterion adopts *MSE*. The minimum samples split is 2, and the minimum leaf is 1.

- (3) Light-GBM: The number of trees is 500, the maximum depth is 6, the model learning rate is 0.1, the number of leaves is 40, the split metric is L1\_mse, the minimum samples split is 2, the bagging fraction is 0.45, the feature fraction is 0.6, and the boosting method is GBDT.
- (4) CNN: The convolutional neural network is built by the Keras library. The CNN model is concluded by convolutional layers and the full connection layers. The convolutional layers are 2, the filters of each convolutional layer are 128/64, and the kernel size of filters is 3. The number of the fully connected layers is 2, and the number of neurons is 16/2, respectively.
- (5) LSTM: The number of hidden layers is 4, and the hidden neurons are 128/64/32/16. The fully connected layers are 2, and the number of neurons is 16/2.

The evaluation results of all models with the testing data are presented in Tables 3 and 4, respectively. Further, the adjustment prediction results of all models with the testing data are presented in Fig. 9a and b, respectively. The related absolute error (ABS\_error) is shown in Fig. 10a and b. For the prediction effect of severe peaks and valleys, the performances of the single learning models are relatively weak. The performance of the proposed TCN-LightGBDT model compared with the other listed models is more optimal.

Figure 11a and b show that the scatter distribution of the proposed TCN-LightGBDT model is more compact than that of other baseline models, which suggests that the predicted value of the proposed model is closer to the actual adjustment value. Further, the fitting curve of the TCN-LightGBDT can fit the real instances well than other models, meaning that the total change in the dependent variable is small. In addition, the  $R^2$  value of all hybrid models indicates that the proposed TCN-LightGBDT model has the highest interpretation for the predicted value. (i.e., TCN-LightGBDT vs LSTM-LightGBM: 1.009, TCN-LightGBDT vs CNN-LightGBDT 1.018, TCN-LightGBDT vs TCN-CNN: 1.025,

**Table 3** The speed-prediction evaluation metrics results of all models

Evaluation Metrics	<i>RMSE-V</i> (10–5 m/s)	<i>FMAE-V</i> (10–4 m/s)	<i>FMAPE-V</i> (%)	$R^2$ (\)	Computational Time (s)
TCN	11.35	6.98	21.66	0.865	2.85
Light-GBDT	7.84	5.32	17.78	0.924	0.12
Light-GBM	7.46	4.82	16.60	0.926	0.34
CNN-LSTM	7.13	4.78	15.88	0.938	3.36
TCN-LSTM	7.01	4.70	15.72	0.941	3.41
TCN-CNN	6.94	4.68	15.65	0.947	3.74
CNN-LightGBDT	5.03	4.01	15.63	0.956	0.65
LSTM-LightGBM	4.20	2.93	11.24	0.961	3.58
Proposed Model	<b>3.84</b>	<b>2.70</b>	<b>8.46</b>	0.968	2.47

**Table 4** The flow-prediction evaluation metrics results of all models

Evaluation Metrics	<i>RMSE-Q</i> (10–4 t/s)	<i>FMAE-Q</i> (10–3 t/s)	<i>FMAPE-Q</i> (%)	<i>R</i> <sup>2</sup> (\)	Computational Time (s)
TCN	4.72	2.71	26.33	0.911	2.17
Light-GBDT	3.96	2.60	19.74	0.924	0.08
Light-GBM	3.89	2.35	18.16	0.929	0.23
CNN-LSTM	3.51	2.09	18.07	0.932	3.27
TCN-LSTM	3.46	2.06	17.76	0.933	3.62
TCN-CNN	3.41	2.03	17.45	0.935	3.86
CNN-LightGBDT	3.39	2.03	16.69	0.938	1.89
LSTM-LightGBM	2.73	1.68	11.55	0.949	3.42
Proposed Model	<b>1.79</b>	<b>1.23</b>	<b>8.82</b>	0.960	3.01

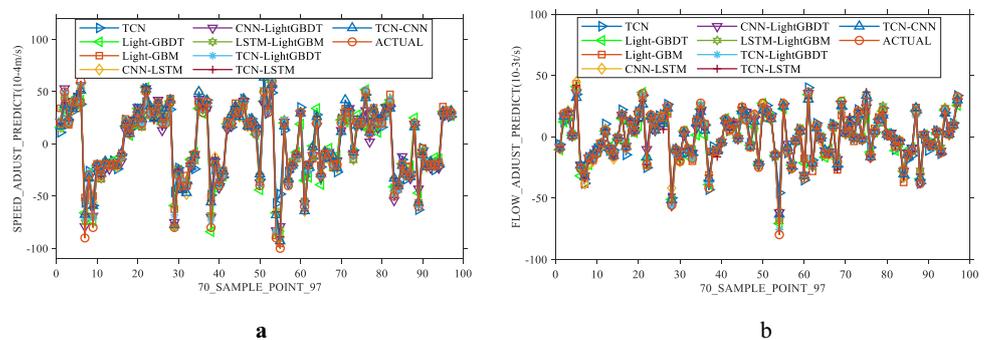
TCN-LightGBDT vs TCN-LSTM: 1.029, TCN-LightGBDT vs CNN-LSTM: 1.031). Namely, the *R*<sup>2</sup> value of the proposed model by linear regression fitting reinforces the fact that the linear correlation between the true and the predicted value is the strongest. In summary, the TCN-LightGBDT model has a better performance than the other models.

2) Experiment 2: Adjustment value prediction of truck speed and chute inclination

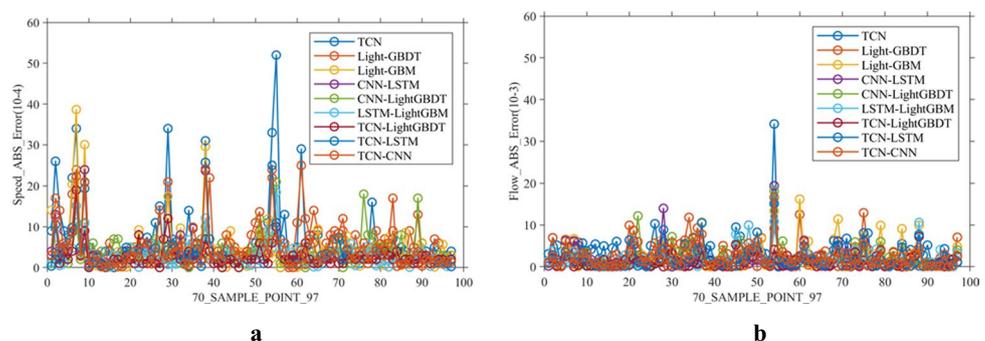
Based on the listed experimental environment and data splitting rules discussed above, we select the data of 20 rear-loading areas to verify the prediction performance of the proposed model. The parameters of each model are summarized as follows.

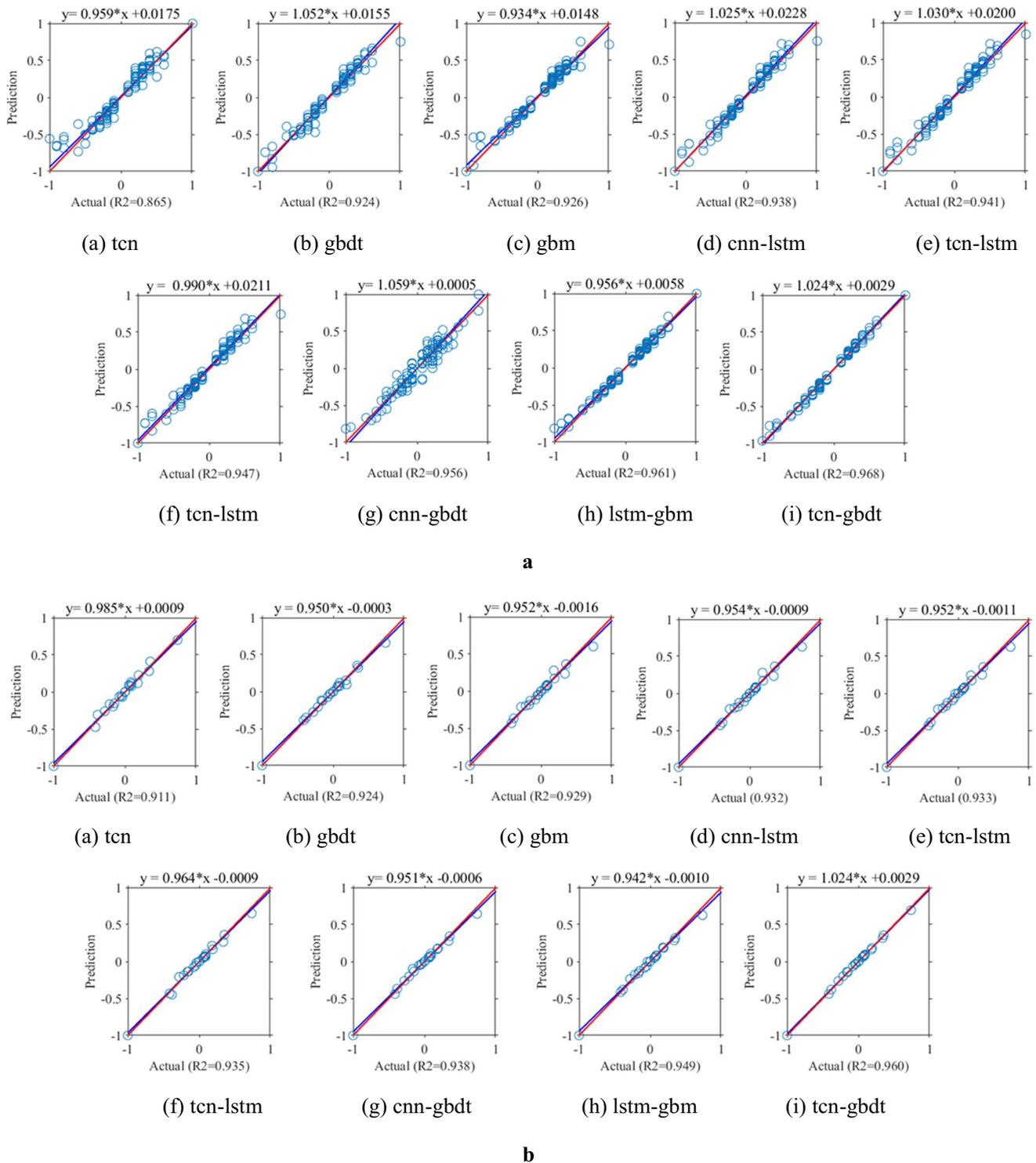
- (1) TCN: The temporal convolution network is built by the Keras library. The dilated convolution factors are [1, 2, 4, 8]. The filters are 64/32/16/16, and the convolutional kernel size is 2.
- (2) Light-GBDT: The number of trees is 500, the maximum depth is 4, the model learning rate is set to 0.1, the minimum sample leaf is 1, and the split criterion adopts *MSE*.
- (3) Light-GBM: The number of trees is 500, the maximum depth is 4, the model learning rate is 0.1, the number of leaves is 40, the split metric is *L1\_mse*, and the minimum samples split is set to 2, the bagging fraction is 0.4, the feature fraction is 0.5, and the boosting method is GBDT.

**Fig. 9** **a** The speed adjustment prediction results. **b** The flow adjustment prediction results



**Fig. 10** **a** The ABS error of speed prediction results. **b** The ABS error of flow prediction results





**Fig. 11** a Scatter diagrams of contrast models for the speed prediction. b. Scatter diagrams of contrast models for the flow prediction

(4) CNN: The convolutional neural network is built by the Keras library. The CNN model is concluded by convolutional layers and the fully connected layers. The convolutional layers are 2, the filters of

each convolutional layer are set to 64/32, and the kernel size of filters is 2. The number of the fully connected layers is 2, and the number of neurons respectively is 16/2.

- (5) LSTM: The number of hidden layers is 3, and the hidden neurons are 64/32/16. The fully connected layers are 2, and the number of neurons is 16/2.

Similar to *Experiment 1*, Fig. 12a and b show the prediction results of the speed and inclination adjustment value. Figure 13a and b show the trend distribution of ABS\_error for each model. It is observed that the prediction performance of the proposed model can precisely match the actual loading adjustment data. In addition, the TCN-LightGBDT model can well capture the continuous stable trend while other hybrids or non-hybrid models show significant fluctuation errors.

To evaluate the effectiveness of our model, we list the evaluation metrics of all models in Table 5 and Table 6. Because of the lower complexity in reconstructed hidden layers of the proposed model, the time cost of speed prediction is slightly reduced. The time cost of the proposed model is no more than 2 s. Figure 14a and b show that the predicted instances using our proposed model can fit the regression curve well. It means that the TCN-LightGBDT model has a better prediction performance for adjustment parameters in the industrial loading than other listed models.

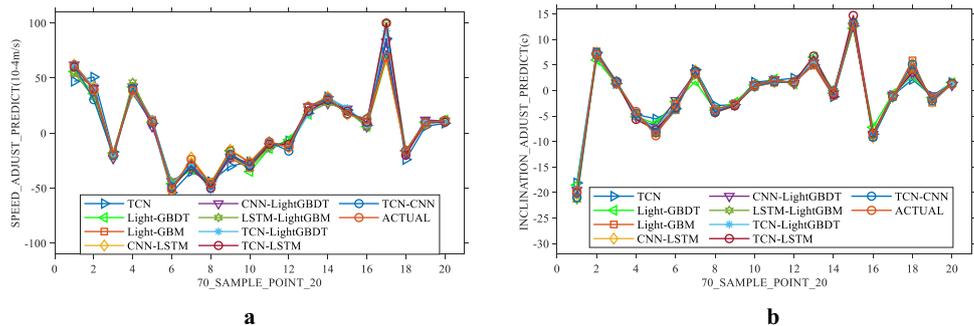
### 4.3 Comparison results between TCN-LightGBDT and TCN-LightGBDT(non-FC)

To verify the effectiveness of the FC method, we compared the TCN-LightGBDT model with the TCN-LightGBDT model (non-FC). The dataset was collected from Jun 1<sup>st</sup>, 2020 to Nov 1<sup>st</sup>, 2020. Further, we normalize the prediction labels to show the difference in performance more clearly, and the complete loading data are randomly adopted to perform the testing results intuitively for the compared models. The detailed experimental settings are as follows.

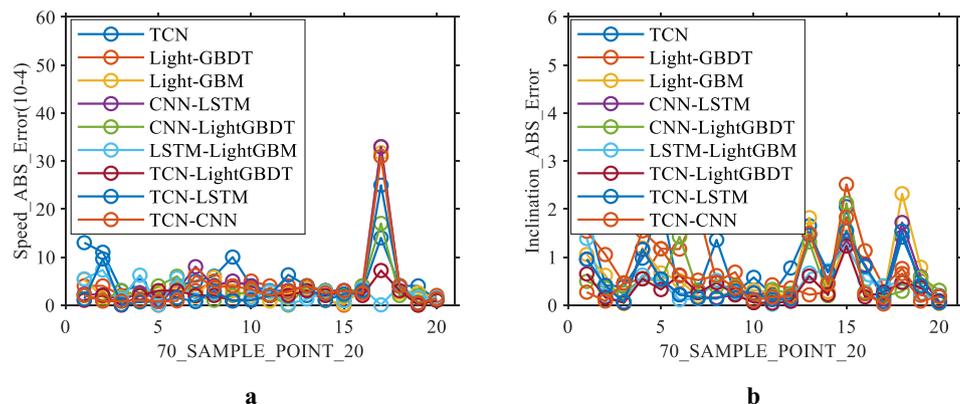
- (1) TCN: The temporal convolution network is built by the Keras library, and the dilated convolution factors are [1, 2, 4], the filters are 64/32/16, and the convolutional kernel size is 2.
- (2) Light-GBDT: The number of trees is 200, the maximum depth is 4, the model learning rate is 0.1, and the split criterion adopts *MSE*. Additionally, the minimum samples split is 1, and the minimum samples leaf is 1.

According to Fig. 15 and Fig. 16, we know that the TCN-LightGBDT without the FC method fluctuates significantly. Based on the FC method, the features convolution and regression prediction can be well connected to improve the

**Fig. 12** a The speed adjustment prediction results. b The inclination adjustment prediction results



**Fig. 13** a The ABS error of speed prediction. b The ABS error of inclination prediction



**Table 5** The speed-prediction evaluation metrics results of all models

Evaluation Metrics	<i>RMSE-V</i> (10–4 m/s)	<i>FMAE-V</i> (10–4 m/s)	<i>FMAPE-V</i> (%)	$R^2$ (\)	Computational Time (s)
TCN	1.41	4.95	19.16	0.917	1.81
Light-GBDT	1.80	4.40	15.39	0.914	0.05
Light_GBM	1.71	3.96	14.62	0.920	0.22
CNN-LSTM	1.39	3.70	13.28	0.926	2.58
TCN-LSTM	1.33	3.43	13.21	0.931	2.86
TCN-CNN	1.26	3.39	13.19	0.945	3.02
CNN-LightGBDT	1.04	3.15	13.17	0.947	0.98
LSTM-LightGBM	0.74	2.52	10.78	0.953	2.94
Proposed Model	<b>0.68</b>	<b>2.24</b>	<b>9.62</b>	0.957	1.87

**Table 6** The inclination-prediction evaluation metrics results of all models

Evaluation Metrics	<i>RMSE-C</i> (°C)	<i>FMAE-C</i> (°C)	<i>FMAPE-C</i> (%)	$R^2$ (\)	Computational Time (s)
TCN	0.23	0.81	21.66	0.921	2.85
Light-GBDT	0.21	0.78	19.87	0.927	0.12
Light-GBM	0.19	0.67	17.35	0.933	0.17
CNN-LSTM	0.17	0.59	16.62	0.938	2.63
TCN-LSTM	0.17	0.56	16.43	0.945	2.97
TCN-CNN	0.16	0.55	14.65	0.947	3.14
CNN-LightGBDT	0.16	0.53	14.73	0.951	1.56
LSTM-LightGBM	0.13	0.41	13.22	0.958	2.37
Proposed Model	<b>0.11</b>	<b>0.35</b>	<b>9.71</b>	0.964	1.63

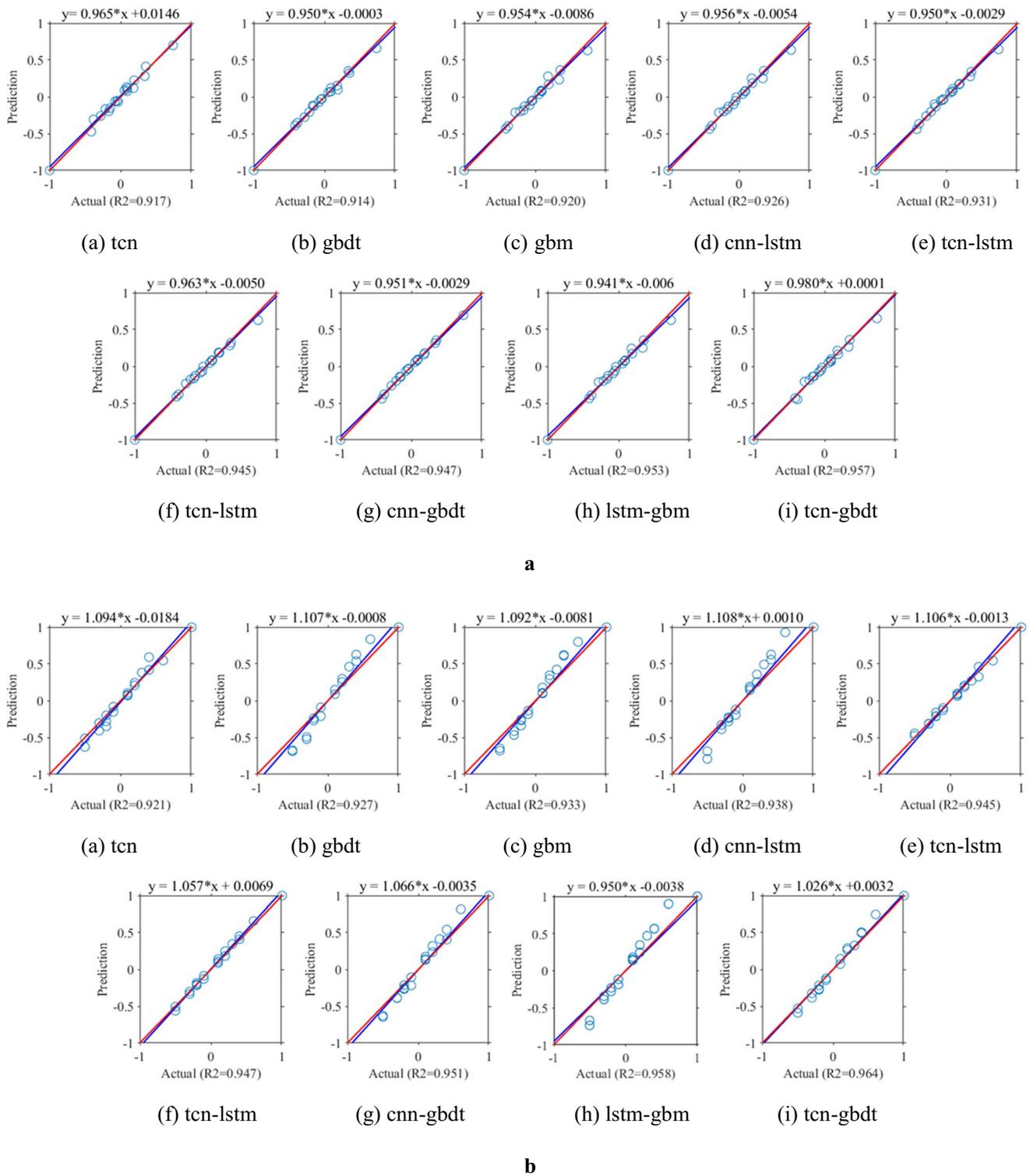
prediction accuracy. Table 7 shows the evaluation metrics results of compared models. It is indicated that the TCN-LightGBDT can have lower errors than the model without the FC method. The  $R^2$  score of the proposed model with the FC method is higher than the TCN-LightGBDT(non-FC) model (TCN-LightGBDT vs. TCN-LightGBDT(non-FC): 1.013/1.015). The computational time of the contrast models is similar and acceptable.

#### 4.4 Discussion and analysis

In the paper, the proposed TCN-LightGBDT model using the TCN and the Light-GBDT is compared with the above baseline models to illustrate better prediction accuracy. Further, some insightful conclusions and theoretical analysis are presented as follows:

- (1) The TCN is superior to the CNN and LSTM in dynamic features extraction. The receptive field size of each residual layer calculated by Formula (12) is listed in Table 8. Compared to the TCN, we can see that the receptive field size entirely depends on the convolution
- (2) The Light-GBDT and the Light-GBM have better prediction performance than the TCN. Because of the strong learner with the negative-gradient fitting, the models can accurately predict nonlinear and low-

kernel for each convolutional layer of the CNN. The final receptive field size of the TCN is 31(16), which can reduce the unnecessary coverage of time-series and improve feature extraction accuracy. Namely, due to the dilated convolution and residual blocks connection, the TCN can obtain a wider receptive field to capture long-term historical relationships. Thus, the prediction ability of the TCN-CNN outperforms the CNN-LSTM and the TCN-CNN. In addition, the prediction of the CNN-LSTM and CNN-LightGBDT are both worse than that of the LSTM-LightGBM. This is because of the limitations of the extraction object or techniques. Namely, the one-dimensional convolution has a relatively poor ability to capture the long-time-range features. Additionally, the LSTM is relatively short of the ability to solve the time series concurrency problem. It is also the main reason why the prediction results of the CNN-LSTM model and the TCN-LSTM model are not as good.

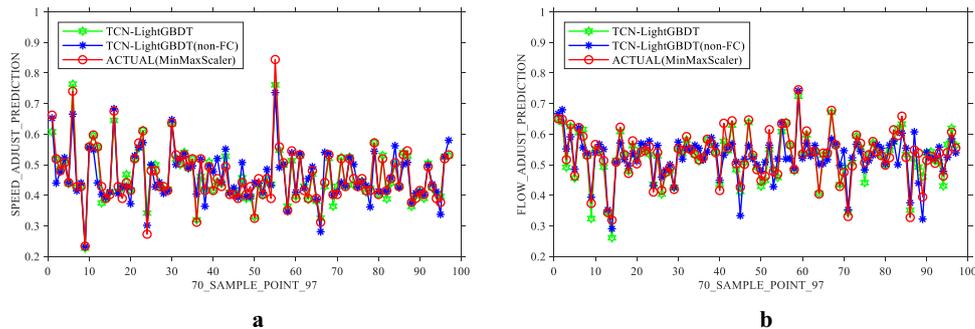


**Fig. 14** **a** Scatter diagrams of contrast models for the speed prediction. **b** Scatter diagrams of contrast models for the flow prediction

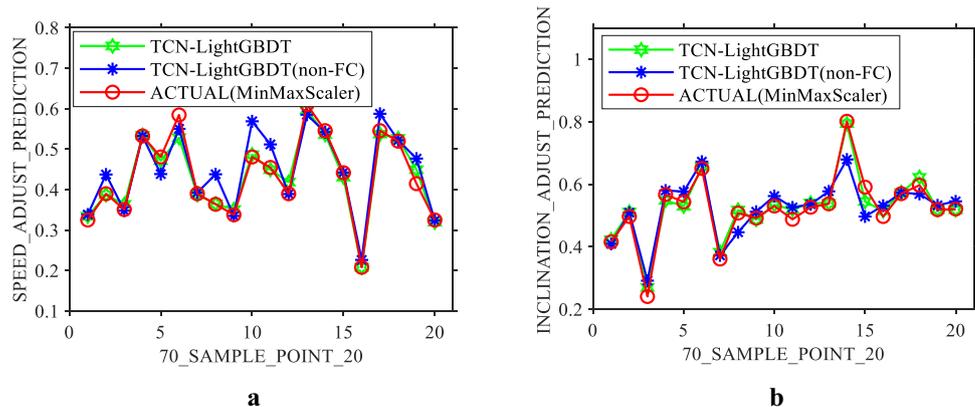
dimensional data. In addition, with the help of the gradient-based one-side sampling and the histogram algorithm, the Light-GBM outperforms the Light-GBDT based on cart regression trees. The decrease of the loss

function along the gradient direction accelerates the function convergence. So, the time consumptions of the Light-GBDT and the Light-GBM are significantly less than that of the TCN.

**Fig. 15** **a** The speed prediction results of 97 points. **b** The flow prediction results of 97 points



**Fig. 16** **a** The speed prediction of 20 points. **b** The inclination prediction of 20 points



(3) In hybrid learning models, the proposed model outperforms the LSTM-LightGBM and the CNN-LightGBDT. Since LSTM relies on historical time series, it will make the predicted error and computational time higher

of the LSTM-LightGBM than the proposed model. In addition, due to the coordinated changes among the adjustment parameters, we provide the FC method to reconstruct the extracted features. The method averages

**Table 7** The evaluation metrics results of compared models

Evaluation Metrics		TCN-LightGBDT (non-FC)	Proposed Model
97 POINTS	<i>RMSE-V</i>	0.04	<b>0.02</b>
	<i>RMSE-Q</i>	0.04	<b>0.02</b>
	<i>FMAE-V</i>	0.03	<b>0.01</b>
	<i>FMAE-Q</i>	0.04	<b>0.02</b>
	<i>FMAPE-V (%)</i>	5.77	<b>2.91</b>
	<i>FMAPE-Q (%)</i>	6.62	<b>3.08</b>
	<i>R<sup>2</sup></i>	0.934	<b>0.946</b>
<i>Computational Time (s)</i>		2.40/2.65	2.43/2.78
20 POINTS	<i>RMSE-V</i>	0.03	<b>0.02</b>
	<i>RMSE-C</i>	0.01	<b>0.02</b>
	<i>FMAE-V</i>	0.02	<b>0.01</b>
	<i>FMAE-C</i>	0.04	<b>0.02</b>
	<i>FMAPE-V (%)</i>	5.56	<b>2.64</b>
	<i>FMAPE-C (%)</i>	6.58	<b>3.01</b>
	<i>R<sup>2</sup></i>	0.939	<b>0.953</b>
<i>Computational Time (s)</i>		1.37/1.46	1.49/1.51

**Table 8** The receptive field size of the TCN and CNN

Layers	TCN	
Residual Layer-1	3	2
Residual Layer-2	7	4
Residual Layer-3	15	8
Residual Layer-4	31	16
Layers	CNN	
Convolutional Layer-1	3	2
Convolutional Layer-2	3	2

feature values to reduce the loss of the abnormal extraction by the TCN, which improves the prediction accuracy of the proposed model. Also, Fig. 17(a) to (d) are the important values of features that are extracted by different models. Among them, Fig. 17(a) and (b) indicate that the CNN-LightGBDT or the TCN-LightGBDT without the FC method excessively depend on the certain feature, and will cause a large error for feature prediction. Figure 17(c) shows that the GBDT prediction relies on too many features extracted by the LSTM, which will decrease the prediction accuracy. In Fig. 17(d), the FC method makes the GBDT prediction process associated with the appropriate features, improving the proposed model's predicted effect.

### 5 Conclusions

In the paper, we propose a TCN-LightGBDT model to achieve the accurate prediction of adjustment values for multi-agent collaborative control parameters in industrial

loading. The loading parameter deviations and theoretical parameter supplement method are used to optimize the dataset, and the FC method is provided for matrix reconstruction of the temporal features extracted by the parallel TCN. Further, we utilize the reconstruction matrix as the feature training set and accurately predict the adjustment parameter values of different combinations using the Light-GBDT in the 117 virtual loading regions. In experiments, we show that the model significantly outperforms other compared models. However, there are still some problems that have not yet been resolved. In the future, we will explore how to accelerate gradient convergence for further reducing time consumption by weight optimization algorithms. Furthermore, we will adopt and apply our proposed model to more related fields (e.g., image target prediction).

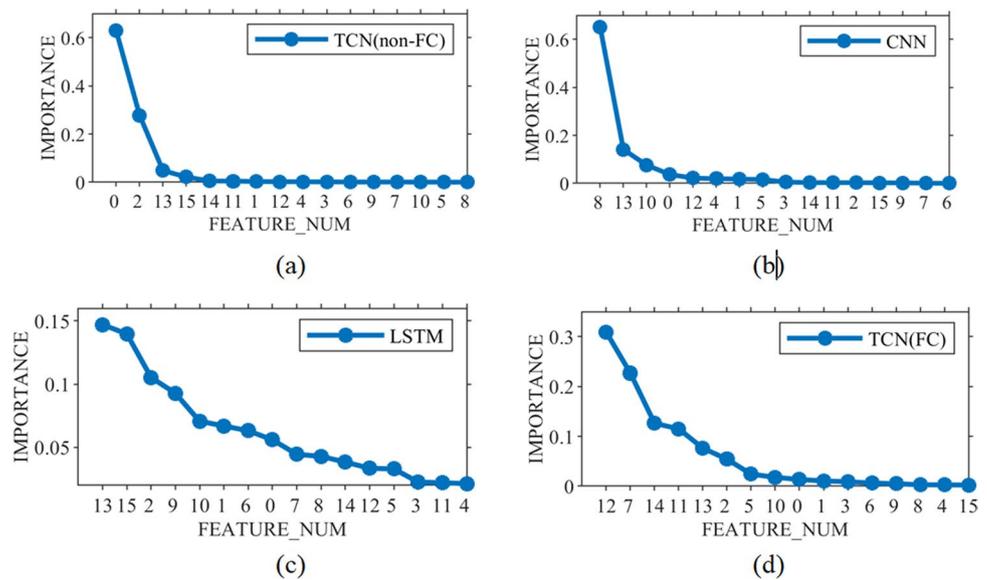
**Author contribution** The designated author has made due contributions to the research work in the following areas, and therefore shares common responsibilities and obligations for the research results.

**Funding** This work was supported by the National Natural Science Foundation of China (Grant No.51874010), the National Natural Science Foundation of China. (Grant No.51675003), the Key Technology Research Innovation Team Project. (Grant No. 201950ZX003) and the Natural Science Research Projects of Colleges and Universities in Anhui Province (KJ2020A0309).

**Data availability** Data sharing and codes are not provided to this article or analyzed during the current research period.

**Code availability** Not available.  
This article is for non-life science journals. And all named authors. Agree to submit this paper to Applied Intelligence.

**Fig. 17** The important values of extracted features



## Declarations

**Conflict of interest** All authors declare there are no other relationships or activities that could appear to have influenced the submitted work. To the best of our knowledge, the named authors have no conflict of interest, financial or otherwise.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Peng Z, Jiang Y et al (2021) Event-triggered dynamic surface control of an underactuated autonomous surface vehicle for target enclosing. *IEEE Trans Industr Electron* 68(4):3402–3412. <https://doi.org/10.1109/TIE.2020.2978713>
- Jani M et al (2020) Performance analysis of a mixed cooperative PLC–VLC system for indoor communication systems. *IEEE Syst J* 14(1):469–476. <https://doi.org/10.1109/JSYST.2019.2911717>
- Yaw CT, Yap KS et al (2020) Enhancement of neural network based multi-agent system for classification and regression in energy system. *IEEE Access* 8:163026–163043. <https://doi.org/10.1109/ACCESS.2020.3012983>
- Aryankia K, Selmic RR (2021) Neuro-adaptive formation control and target tracking for nonlinear multi-agent systems with time-delay. *IEEE Control Syst Lett* 5(3):791–796. <https://doi.org/10.1109/LCSYS.2020.3006187>
- HanQ, Cao R et al (2019) Leader-Following Consensus of Multi-Agent System based on Cellular Neural Networks Framework. 2019 2nd International Conference on Safety Produce Informatization (IICSPI), pp 142–146. <https://doi.org/10.1109/IICSPI.2019.9096018>
- Dong G, Li H et al (2021) Finite-time consensus tracking neural network FTC of multi-agent systems. *IEEE T Neur Net Lear* 32(2):653–662. <https://doi.org/10.1109/TNNLS.2020.2978898>
- Agga A, Abbou A et al (2021) Short-Term Self Consumption PV Plant Power Production Forecasts Based on Hybrid CNN-LSTM, ConvLSTM Models. *Renewable Energy*, available online. <https://doi.org/10.1016/j.renene.2021.05.095>
- Bao T, Zaidi SAR et al (2021) A CNN-LSTM hybrid model for wrist kinematics estimation using surface electromyography. *IEEE Trans Instrum Meas* 70:1–9. <https://doi.org/10.1109/TIM.2020.3036654>
- Roy SK, Krishna G et al (2020) HybridSN: exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci Remote Sens Lett* 17(2):277–281. <https://doi.org/10.1109/LGRS.2019.2918719>
- Kollias D, Zafeiriou S (2021) Exploiting Multi-CNN features in cnn-rnn based dimensional emotion recognition on the OMG in-the-Wild dataset. *IEEE Trans Affect Comput* 12(3):595–606. <https://doi.org/10.1109/TAFFC.2020.3014171>
- Fang F, Li L et al (2020) Combining faster R-CNN and model-driven clustering for elongated object detection. *IEEE Trans Image Process* 29:2052–2065. <https://doi.org/10.1109/TIP.2019.2947792>
- Luo S, Ye D (2019) Adaptive double event-triggered control for linear multi-agent systems with actuator faults. *IEEE Trans Circuits Syst I Regul Pap* 66(12):4829–4839. <https://doi.org/10.1109/TCSI.2019.2932084>
- Xiao G et al (2021) Multimodality sentiment analysis in social internet of things based on hierarchical attentions and CSAT-TCN with MBM network. *IEEE Internet Things J* 8(16):12748–12757. <https://doi.org/10.1109/JIOT.2020.3015381>
- Bethausen JL et al (2020) Stable responsive EMG sequence prediction and adaptive reinforcement with temporal convolutional networks. *IEEE Trans Biomed Eng* 67(6):1707–1717. <https://doi.org/10.1109/TBME.2019.2943309>
- Zhang P, Xue J et al (2020) EleAtt-RNN: adding attentiveness to neurons in recurrent neural networks. *IEEE Trans Image Process* 29:1061–1073. <https://doi.org/10.1109/TIP.2019.2937724>
- Zhang Z, Jung C (2021) GBDT-MO: gradient-boosted decision trees for multiple outputs. *IEEE Trans Neural Netw Learn Syst* 32(7):3156–3167. <https://doi.org/10.1109/TNNLS.2020.3009776>
- Zhang Y, Wang J et al (2020) Efficient selection on spatial modulation antennas: learning or boosting. *IEEE Wireless Commun Lett* 9(8):1249–1252. <https://doi.org/10.1109/LWC.2020.2986974>
- Ma X, Ding CS et al (2017) "Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method. *IEEE Trans Intell Transp Syst* 18(9):2303–2310. <https://doi.org/10.1109/TITS.2016.2635719>
- Xu Z, Li C et al (2020) Impulsive consensus of nonlinear multi-agent systems via edge event-triggered control. *IEEE Trans Neural Netw Learn Syst* 31(6):1995–2004. <https://doi.org/10.1109/TNNLS.2019.2927623>
- Han Y, Li C et al (2020) Impulsive consensus of multiagent systems with limited bandwidth based on encoding–decoding. *IEEE Trans Cybernetics* 50(1):36–47. <https://doi.org/10.1109/TCYB.2018.2863108>
- Lindemann L, Dimarogonas DV (2019) Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks. *IEEE Control Syst Lett* 3(3):757–762. <https://doi.org/10.1109/LCSYS.2019.2917975>
- Lian F, Chakraborty A, Duel-Hallen A (2017) Game-theoretic multi-agent control and network cost allocation under communication constraints. *IEEE J Sel Areas Commun* 35(2):330–340. <https://doi.org/10.1109/JSAC.2017.2659338>
- Qian Y, Liu L et al (2020) Distributed event-triggered adaptive control for consensus of linear multi-agent systems with external disturbances. *IEEE Trans Cybernetics* 50(5):2197–2208. <https://doi.org/10.1109/TCYB.2018.2881484>
- Tan H, Miao Z et al (2020) Data-driven distributed coordinated control for cloud-based model-free multiagent systems with communication constraints. *IEEE Trans Circuits Syst* 67(9):3187–3198. <https://doi.org/10.1109/TCSI.2020.2990411>
- Tian W, Salscheider NO et al (2020) A collaborative visual tracking architecture for correlation filter and convolutional neural network learning. *IEEE Trans Intell Transp Syst* 21(8):3423–3435. <https://doi.org/10.1109/TITS.2019.2928963>
- Song J, Xue G et al (2020) Hourly heat load prediction model based on temporal convolutional neural network. *IEEE Access* 8:16726–16741. <https://doi.org/10.1109/ACCESS.2020.2968536>
- He W, Sun Y et al (2020) Disturbance observer-based neural network control of cooperative multiple manipulators with input saturation. *IEEE Trans Neural Netw Learn Syst* 31(5):1735–1746. <https://doi.org/10.1109/TNNLS.2019.2923241>

28. Wang Z, Li L et al (2018) Handover control in wireless systems via asynchronous multiuser deep reinforcement learning. *IEEE Internet Things J* 5(6):4296–4307. <https://doi.org/10.1109/JIOT.2018.2848295>
29. Wang H, Zhang Q (2017) Dynamic tension test and intelligent coordinated control system of a heavy scraper conveyor. *IET Sci Meas Technol* 11(7):871–877. <https://doi.org/10.1049/iet-smt.2016.0425>
30. Gehrman S, Strobelt H et al (2020) Visual interaction with deep learning models through collaborative semantic inference. *IEEE Trans Visual Comput Graphics* 26(1):884–894. <https://doi.org/10.1109/TVCG.2019.2934595>
31. Wang T, Wang X et al (2020) Random forest-bayesian optimization for product quality prediction with large-scale dimensions in process industrial cyber–physical systems. *IEEE Internet Things J* 7(9):8641–8653. <https://doi.org/10.1109/JIOT.2020.2992811>
32. Wang L, Yang J et al (2020) Impact of backscatter in pol-insar forest height retrieval based on the multimodel random forest algorithm. *IEEE Geosci Remote Sens Lett* 17(2):267–271. <https://doi.org/10.1109/LGRS.2019.2919449>
33. Sun R et al (2021) Improving GPS code phase positioning accuracy in urban environments using machine learning. *IEEE Internet Things J* 8(8):7065–7078. <https://doi.org/10.1109/JIOT.2020.3037074>
34. Thai D, Tu TM, Bui TQ, Bui TT (2021) Gradient tree boosting machine learning on predicting the failure modes of the RC panels under impact loads. *Eng Comput -Germany* 37(1):597–608. <https://doi.org/10.1007/s00366-019-00842-w>
35. Lu L, Bo L (2021) Reducing energy consumption of Neural Architecture Search: an inference latency prediction framework. *Sustain Cities Soc* 67(8):102747. <https://doi.org/10.1016/j.scs.2021.102747>
36. Dan Y et al (2020) Computational prediction of critical temperatures of superconductors based on convolutional gradient boosting decision trees. *IEEE Access* 8:57868–57878. <https://doi.org/10.1109/ACCESS.2020.2981874>
37. Kong H, Lin S et al (2019) The risk prediction of mobile user tricking account overdraft limit based on fusion model of logistic and GBDT. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, pp 1012–1016. <https://doi.org/10.1109/ITNEC.2019.8729173>
38. Bi J, Zhang X, Yuan H, Zhang J, Zhou M (2021) A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM. *IEEE T Autom Sci Eng* 1–11. <https://doi.org/10.1109/TASE.2021.3077537>
39. Zhang Y, Sun Y, Liu S (2021) Deformable and residual convolutional network for image super-resolution. *Appl Intell*. <https://doi.org/10.1007/s10489-021-02246-0>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.