IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 2, 2020, accepted April 20, 2020, date of publication May 6, 2020, date of current version May 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992451

The Deep Convolutional Neural Network for NOx Emission Prediction of a Coal-Fired Boiler

NAN LI[1] AND YONG HU[2]

[1]School of Information and Electrical Engineering, Lu Dong University, Yantai 264025, China
[2]State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing 102206, China

Corresponding author: Nan Li (nan_li_ldu@outlook.com)

This work was supported in part by the Shandong Provincial Natural Science Foundation of China under Project ZR2018PF003, and in part by the National Natural Science Foundation of China under Grant 51806063.

ABSTRACT This paper presents a methodology for predicting NOx emissions of a coal-fired boiler by using real operation data, coal properties and CNN (Convolutional Neural Network). Two building blocks are carefully designed following the practical guidelines for the light weight CNN architecture design. Furthermore, the building blocks are used to develop the deep CNN-based model for NOx prediction. A comprehensive comparison among different prediction models based on DL (Deep Learning) shows that the proposed deep CNN-based prediction model outperforms other prediction models in terms of RMSE (Root Mean Square Error) criteria. The results indicate that the developed deep CNN-based prediction model has more excellent accuracy and better numerical stability. Besides, the architecture design of the DL-based prediction model has a significant impact on the performance of the prediction model.

INDEX TERMS Coal-fired boiler, convolutional neural network, deep learning, NOx emission prediction.

I. INTRODUCTION

Coal is the primary fuel used in power plants to generate electricity. However, NOx emissions during coal combustion are responsible for human health and environmental pollution [1]. The control of NOx emissions from coal combustion is still a concerned issue in many countries. Therefore, the clean and efficient utilization of coal in power plants has become one of today's main objectives in coal combustion researches.

The combustion optimization approach can effectively reduce NOx emissions by adjusting the operational parameters [2], [3]. For this approach, it is crucial to develop an accurate prediction model for NOx emissions at the furnace exit. However, because of complex combustion dynamics, fluid mechanics and nitrogen conversion chemistry, it is difficult to build such a prediction model based on the overall dynamics of the boiler. Alternatively, the advanced machine learning algorithms can be used to build the relationship between the related operational parameters and NOx emissions at the furnace exit.

Some studies have been conducted on applying shallow learning algorithms (such as shallow neural network, support vector machine, extreme learning machine, and their variants) for modeling NOx emissions in coal combustion processes [4]–[7]. The above studies have achieved some success in the prediction of NOx emissions; however, there are some weaknesses among these shallow learning algorithms. First, the operational variables for modeling NOx Emissions contain complex data information in part reflecting the complex dynamics of the boiler and the peak load regulation. These algorithms are considered difficult to learn such complex nonlinear functions [8]. Second, these algorithms have some restrictions on the size of the training data set. They are prone to overfitting when using a large data set [9]. Overfitting can severely degrade algorithm performance. Third, these algorithms are of the inability to learn distributed and hierarchical feature representations from the data. Thus, much of the actual effort in deploying these algorithms goes into the design of preprocessing pipelines [10].

Recent studies have focused on introducing deep learning (DL) algorithms for modeling NOx emissions during coal combustion. Due to high-performance computing systems, DL algorithms can model complex non-linear relationships and learn internal representation for a large amount of data [8]–[10]. Also, some techniques are being proposed to alleviate the overfitting problem. Wang *et al.* developed the DL-based NOx prediction model based on the deep belief network [11]. However, the feature representation process is completely independent of the NOx prediction process in

The associate editor coordinating the review of this manuscript and approving it for publication was Venkateshkumar M.

this model. This design approach degrades the model's performance. Tan *et al*. used the recurrent neural network with Long Short-Term Memory (which we will concisely refer to as LSTM) to model NOx emissions [12]. Yang *et al*. used two LSTMs to build the DL-based NOx prediction model [13]. Xie *et al*. used the LSTM variant called bidirectional LSTM as the building block to build encoder-decoder architecture to predict NOx emissions [14]. LSTM can capture long-term temporal dependencies from data by storing the history information, which leads to increased storage cost and computing cost. Thus, training LSTM or its variants is difficult and time-consuming [15].

Compared with LSTM, the convolutional neural network (CNN) is usually at a considerably cheaper computational cost on certain sequence-processing problems. CNN is a type of feed-forward artificial neural network and uses convolution operation in place of general matrix multiplication to reduce the computational burden [16]. The representations learned from data are translation invariant, which means the representations do not change even though the input of CNN is translated by a small amount. Thus, CNN can use fewer training samples to learn representations having better generalization power. Deep CNN has become the master algorithm in computer vision since AlexNet won the ImageNet Challenge [17]–[21]. However, the application of CNN is very limited for modeling NOx emissions in coal-fired power plants. In the present work, we proposed a deep CNN-based model for predicting NOx emissions of a coal-fired boiler, aiming to develop an accurate prediction model for NOx emissions at the furnace exit for more effective emissions reduction. Two building blocks are designed to learning richer data representations with less parameter. The building blocks are used to build a light-weight model to predict the NOx emissions at the furnace exit of a 330MW pulverized coal-fired utility boiler. The data samples from the distributed control system (DCS) are employed to train and test the proposed NOx emission prediction model. Furthermore, comparisons with the other DL-based NOx prediction models are conducted. The remainder of this paper is organized as follows. Section 2 describes the work in developing the building block and deep CNN-based prediction model. Section 3 describes the detailed application of NOx emissions prediction and model comparisons. Section 4 closes with a summary and conclusion.

II. CONSTRUCTION OF THE DEEP CNN-BASED PREDICTION MODEL ARCHITECTURE

A. BRIEF DESCRIPTIONS OF THE BOILER AND DATA PREPARATION

The studied boiler is 330MW subcritical tangential pulverized coal-fired utility boiler manufactured by Shanghai Boiler Co. Ltd. The boiler belongs to one unit of Dong Sheng power plant in Inner Mongolia, China. A schematic diagram of the furnace is shown in Fig.1. Five layers of primary air (A, B, C, D, and E) and eight layers of input air (AA, AB, BC, CC, DD, DE, EF, and FF) are distributed alternately in a vertical direction. Five medium-speed coal pulverizers are put into operation to supply with fuel for combustion. Coal-air mixtures are fed to the burners on A-E levels. Four layers over fire air (OFA) are fixed over the upper nozzles to replenish the air in the combustion anaphase for better combustion efficiency.

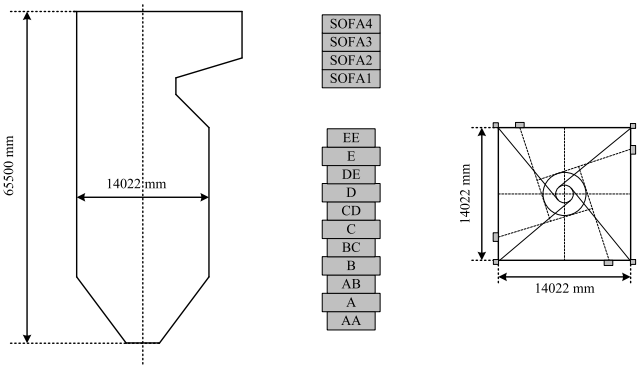


FIGURE 1. Schematic diagram of the furnace.

TABLE 1. Industrial analysis of the combusted coal.

Time	Volatile compounds (%)	Ash(%)	Moisture (%)	Sulfur (%)	Quantity of produced heat (MJ/Kg)
Day1	40.69	16.44	26.0	1.18	3.886
Day2	40.93	16.34	26.1	1.21	3.871
Day3	40.40	17.68	26.4	0.97	3.724
Day4	39.31	15.77	25.2	1.16	3.953
Day5	39.30	15.51	26.2	1.02	3.919
Day6	39.29	14.62	26.1	1.02	4.01
Day7	38.80	14.77	26.8	0.96	3.954
Day8	38.25	15.4	25.8	0.91	3.944
Day9	37.48	14.63	26.50	0.90	3.955
Day10	37.44	12.64	27.10	0.92	4.077

The data for modeling NOx emissions consists of three parts. First, the coal burned in the boiler is an important factor responsible for the NOx formation. The coal properties are given by industrial analysis and the analysis results are listed in Table1. There are no real-time data about coal properties due to the lack of an on-line coal analyzer in the power plant. Thus, these coal properties are introduced to build a NOx prediction model. Second, fifty-five operational variables, including boiler load (one), main steam pressure (one), total fuel flow (one), total air flow (one), coal-feeder rate (five), primary air flow (five), primary air temperature (five), main steam temperature (one), total secondary air flow(two), secondary air temperature (two), secondary air flow (twenty-four), main steam flow (one), OFA air flow (four), Oxygen concentration before the selective catalytic reduction inlet (two), have been selected based on the engineers' advice and the knowledge of the tangentially coal-fired boiler. The data points covering ten days are obtained from DCS with a time resolution of 1 second. Third, NOx emissions at side A and side B of the furnace exit (two) have also been considered.

To construct the dataset for modeling, three steps are adopted in succession on the raw data. First, extreme outliers

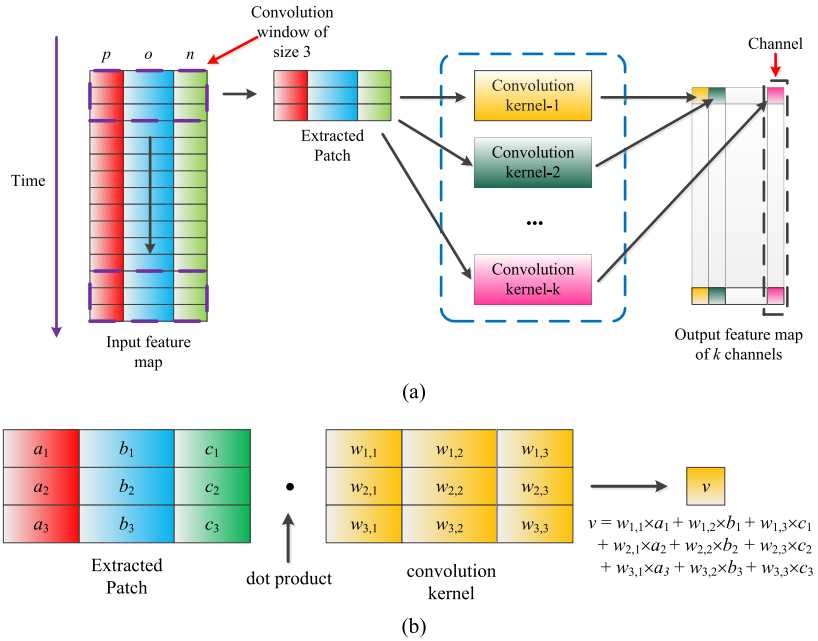


FIGURE 2. The overview of 1D CNN layer. (a) How 1D CNN layer works on the multivariate time series, (b) the dot product on the patch and convolution kernel.

are removed to improve data quality. Second, the data should be normalized to make learning easier for our prediction model. All the data should be standardized by removing the mean and scaling to unit variance as follow:

$$z = \frac{x - \mu}{s} \tag{1}$$

where x and z are the operational variable or NOx emissions before and after scaling, μ is the sample mean, and s is the standard deviation. Third, the data samples in the dataset used for modeling NOx emissions should have the form as follow:

$$(x(t), y(t)) \tag{2}$$

where

$$x(t) = \begin{bmatrix} p & o(t-K_1) & n(t-K_1) \\ p & o(t-K_1-1) & n(t-K_1-1) \\ \vdots & \vdots & \vdots \\ p & o(t-1) & n(t-1) \\ p & o(t) & n(t) \end{bmatrix} \tag{3}$$

$$y(t) = \frac{1}{K_2} \sum_{i=1}^{K_2} n(t+i). \tag{4}$$

In the above matrix, p denotes the row vector containing the coal properties, $o(t)$ denotes the row vector containing selected operational variables at time t, $n(t)$ denotes the row vector containing NOx emissions at time t, K_1 and K_2 are nonnegative integers. Thus, these data samples belong to the multivariate time series. In this study, both of K_1 and K_2 are equal to 60. Modeling NOx emissions based on this dataset means that using the data samples in the first 60 seconds to predict the mean of NOx emissions in the next 60 seconds. There are two motivations. First, there is a stronger correlation between the adjacent data samples. Second, the data samples with a larger value of K_1 contain more information.

B. BRIEF INTRODUCTION TO CNN

Recently, there has been lots of progress in designing a small and computation-efficient deep CNN architecture for mobile and embedded vision applications, such as ShuffleNetV1 [20] and ShuffleNetV2 [21]. These architectures are suitable for the applications which need to be carried out in a timely fashion. CNNs used in these architectures are designed for processing image data. This type of CNN is referred as 2D CNN layer. However, the multivariate time series data for modeling NOx emissions is completely different from these image data. Thus, we consider using a type of CNN for processing the multivariate time series data which is referred as 1D CNN layer.

The data operated by CNN is called feature map and the column vector in feature map is called a channel. The computation procedure of 1D CNN layer is shown in Fig. 2 (a). Firstly, we should determine the size of the convolution window. The convolution window is used to extract the patches from the multivariate time series along the time axis. The extracted patches are essentially the numerical matrixes having the same dimensions as the convolution window. Secondly, the extracted patches are sent to a group of the convolution kernels. Any convolution kernel is a weight matrix that is not predefined but is learned during the training process of a 1D CNN layer. The scalar value is obtained by taking dot products on the patch and the convolution kernel. Such an instance is shown in Fig. 2 (b). The definition of the dot

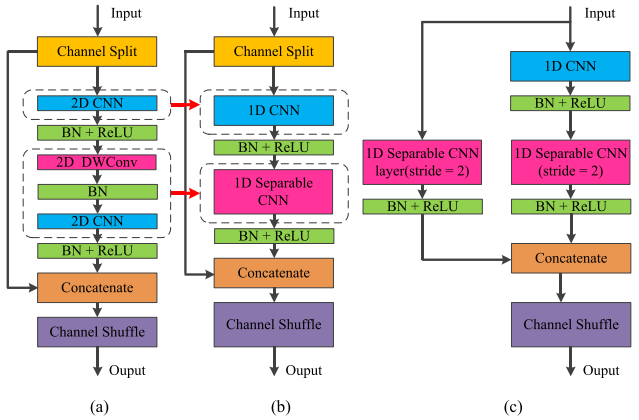


FIGURE 3. Building block of ShuffleNetV2 and our basic building blocks. DWConv stands for depthwise convolution. (a) Building block of ShuffleNetV2, (b) our basic building block, (c) our basic building block with stride 2.

product on two matrixes is as follows:

$$A \cdot B = \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} b_{ij} \tag{5}$$

where $A = \left[a_{ij}\right]_{m \times n}$ and $B = \left[b_{ij}\right]_{m \times n}$.

C. ARCHITECTURE DESIGN OF THE BASIC BUILDING BLOCKS

In [21], there are some practical guidelines, proposed for light weight CNN architecture design. Based on these guidelines for ShuffleNetV2, two basic building blocks in this study are designed. However, the original architecture of ShuffleNetV2, which is designed to process the tasks in computer vision, can't process the multivariate time series data for modeling NOx emissions. In order to process the multivariate time series data feasibly and efficiently, two important modifications are made in our basic building blocks. Firstly, the first 2D CNN layer in the first dashed box in Fig. 3 (a) must be replaced by 1D CNN layer in the first dashed box in Fig. 3 (b). Secondly, the components in the second dashed box in Fig. 3 (a) are replaced by a 1D separable CNN layer in the second dashed box in Fig. 3 (b).

As shown in Fig. 3 (b), there is a channel split operator at the beginning of the basic building block. The input having k channels is split into two branches with k_1 and k_2 channels, respectively. The left branch is a shortcut connection introduced in ResNet [18]. It can be considered as an identity map and all information is always passed through. The right branch consists of four components. The first component is a 1D CNN layer. The size of convolution window of this CNN layer must be fixed to 1. It can be considered as bottleneck layer to reduce the number of input feature maps, and thus to improve computational efficiency. Next, the second component contains batch normalization [22] and a rectified linear unit [23]. Batch normalization maintains an exponential moving average of the batch-wise mean and variance of the data during training. It has been proved to accelerate the training process of the CNN layer. The rectified linear unit (ReLU) is defined by the activation function $f(x) = max\{0, x\}$. It is considered as the most important factor in improving the performance of CNNs [24]. The third component is a 1D separable CNN layer. The separable CNN layer, which is referred as depth-wise separable convolution, consists of the depth-wise convolution and the pointwise convolution [25]. First, the depth-wise convolution performs independently a convolution operation on each channel of its input. Second, the pointwise convolution creates a linear combination of the output channel of the depth-wise convolution. Some studies have demonstrated that the separable CNN layer can efficiently reduce the computation cost and learn better representations using fewer data [19]. 1D separable CNN layer is the version of separable CNN layer which can process the multivariate time series. The fourth component is the same effect as the second component. The results of the two branches are concatenated to keep the number of channels same as the input. At the end of the basic building block, the channel shuffle operation is used to reshape the order of channels of output to enable information communication work between different channels.

Fig. 3 (c) shows the architecture of the basic building block with stride 2. The stride is a parameter defined by the distance between two successive convolution windows. Using stride equal to 2 means the row rank of the input feature map is down sampled by a factor of 2 to reduce the computational cost and the number of parameters. In addition, the risk of overfitting is limited. The basic building block with stride 2 is different from the basic building block. Firstly, the channel split operator is removed. Secondly, in the right branch, a 1D separable CNN layer is replaced by a 1D separable CNN layer with stride 2. Thirdly, in the left branch, a 1D separable CNN layer is added to keep the size of the input the same as the output of the right branch. BN and ReLU achieve the same effect as in the basic building block.

D. DEEP CNN-BASED MODEL FOR NOx PREDICTION

The block diagram as shown in Fig. 4 is a deep CNN-based model for NOx emissions prediction. The model is a streamlined architecture based on the basic building blocks in Fig. 3 (b) and (c). The design of CNNs used in the model has twofold: (1) they are used to gradually increase the number of the channels of the output feature map; (2) they are used to gradually reduce the row rank of the output feature map. This guideline will make the deep CNN-based model wider and deeper, which has been proven to increase the performance of the model [19]–[21].

The first component is a 1D CNN layer with stride 2. The size of convolution window of this CNN layer is set to 3. The second component consisting of BN and ReLU has the same effect as the components in the basic building block. The following three components have the same structure but with different parameters. In each stage, the basic building block with stride 2 is set at the beginning, and the basic building

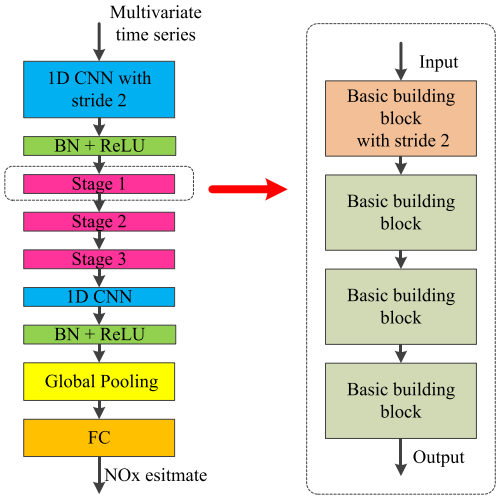


FIGURE 4. The architecture of NOx prediction model.

block is repeated three times. The data representations are further refined after these three stages. Next, the sixth component is also a 1D CNN layer. The size of convolution window is set to 1. The seventh component is the same as the second component. After that, the data representations will have two dimensions but can't directly be used for prediction. Consequently, the global average pooling layer is introduced to reduce the dimensions of the data representations and result in one-dimension vectors. These vectors go through the final component which is a regular fully-connected layer (FC layer). This FC layer has two outputs for NOx emissions at side A and side B. The first eight components are used to extract the data representations from the multivariate time series, and the final component is used to predict NOx emissions.

To evaluate our prediction model, the dataset should be splitted into three sets. The training set consists of 60% data samples; the validation set consists of 30% data samples; and the test set consists of 10% data samples. The division of the data depends on the size of the dataset which covers different operation conditions. It is stresses that the training set and the validation set containing the enough data can improve the generalization error of the prediction model. Root mean square error (RMSE) is introduced to evaluate the performance of the NOx prediction model. It is true that the root mean square error (RMSE) is a widely used performance measure for regression problems. It gives an idea of how much error the model typically makes in its predictions, with a higher weight for large errors. It is defined as,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \tag{6}$$

where N denotes the number of the data samples, $\hat{y}_i$ denotes the measured value and $\hat{y}_i$ denotes the corresponding predicted value.

TABLE 2. The summary statistics of the prediction results of our prediction model on the test set.

		Mean RMSE(mg/Nm^3)	Standard deviation of RMSE(mg/Nm^3)
Our model	Side A	1.11	0.68
	Side B	1.06	0.59

III. RESULTS AND DISCUSSION

A. NOx PREDICTION RESULTS

We implemented our model using the open-source deep learning library Keras with the TensorFlow back-end [26]. A single NVIDA GeForce GTX 1080 is used. The convolution library is CUDNN 10.0 [27]. The optimization configuration is used for our model: the Optimizer is Adam [28]; the initial learning rate is 0.001 and the decay of rate is 0.95 every 5 epochs. To avoid the overfitting problem, the early stopping strategy is applied to the validation set. Thus, a model checkpoint procedure should be performed either on the training set or validation set to keep the best model during the training process. The model follows most of the hyper-parameters used in [21].

There are 30 runs of our model to evaluate model reliability. The summary statistics are shown in Table 2. The mean RMSEs of the test set at side A and side B are 1.11 mg/Nm^3 and 1.06 mg/Nm^3 respectively. The lowest mean RMSEs show that our model has high prediction accuracy on the testing set. The standard deviations of RMSEs at side A and side B are 0.68 mg/Nm^3 and 0.59 mg/Nm^3, respectively. The lowest standard deviations of RMSEs demonstrate a good stability of our proposed model.

For the 3rd run, RMSEs at side A and side B are 0.94 mg/Nm^3 and 1.07 mg/Nm^3, which are very close to the average RMSEs at side A and side B. Fig. 5 shows the predicted values at the 3rd run on the test set. The predicted values are in good agreement with the reference values. Fig. 6 shows the relative errors at the 3rd run on the test set. The maximum relative error is 1.55% at side A, and the maximum relative error is -1.6% at side B. The good prediction performance on test set exhibits a satisfactory capability of the deep CNN-based prediction model in this study.

B. MODEL COMPARISONS AND DISCUSSIONS

In this section, we survey a variety of DL-based prediction models based on the leading building blocks and make comparisons with our proposed model. For fair comparison, we do not use any data preprocessing methods except the methods in building the dataset, and all prediction models for a comparison have the same training environment. DL-based prediction models for comparison are as follows:

(1) VGGNet is a deep CNN architecture consisting of multiple 2D CNN layers. Its building block is a single 2D CNN layer. Following the design principle of VGGNet, the VGG-like prediction model was developed based on the

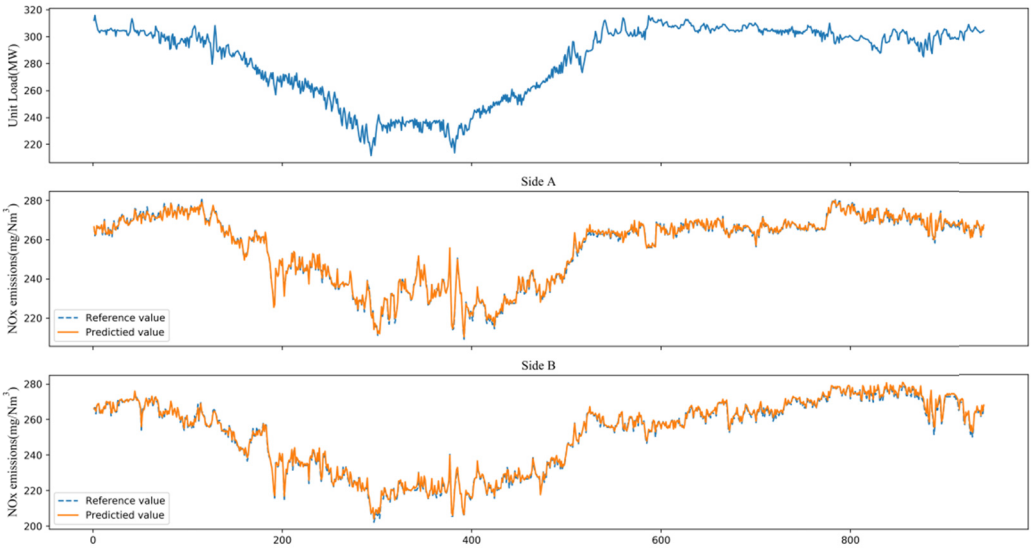


FIGURE 5. Variations of unit load and estimation of NOx emissions.

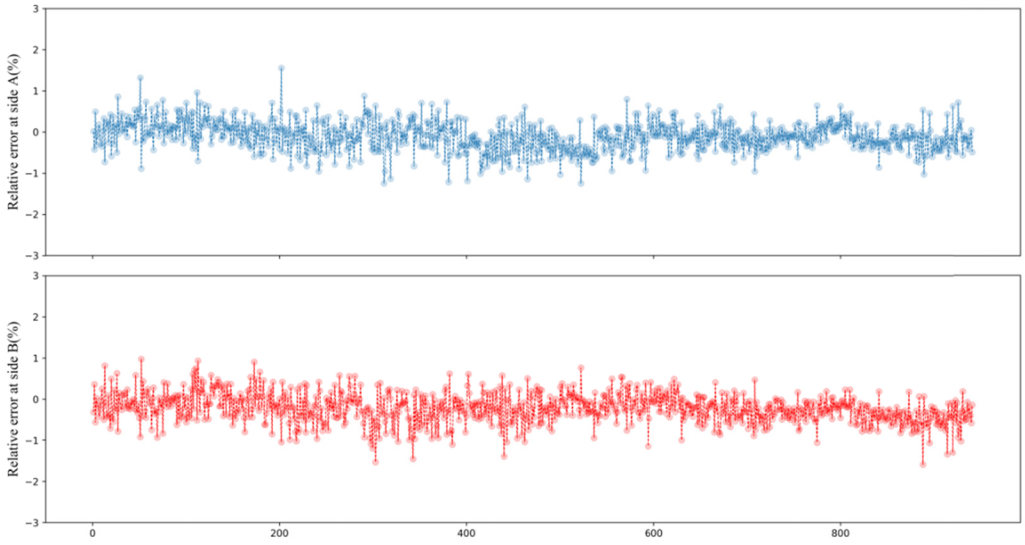


FIGURE 6. Variations of relative error for NOx emission prediction.

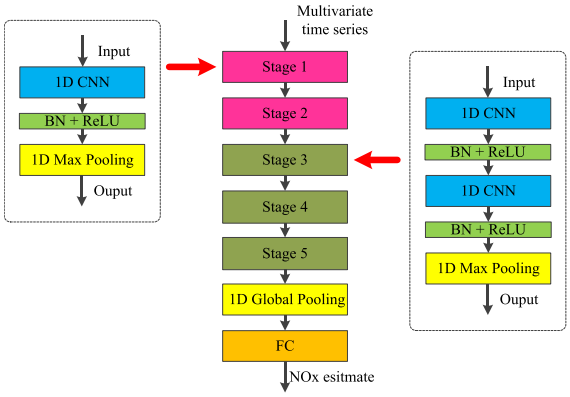


FIGURE 7. The VGG-like prediction model.

1D CNN layers. Different from VGGNet, BN and ReLU are added after each of the 1D CNN layers. The overall architecture of the VGG-like prediction model is shown in Fig. 7. The first two stages have the same structure, and the next three stages have the same structure. This model follows most of the hyper-parameters used in [17].

(2) ResNet is a deep CNN architecture introducing shortcut connections which can improve the training efficiency. Its building block consists of multiple 2D CNN layers and a shortcut connection as shown in Fig. 8 (a). We use 1D CNN layers to replace 2D CNN layers. The modified building blocks of ResNet-18 are shown in Fig. 8 (b) and (c). The overall architecture of the ResNet-like prediction model is shown in Fig. 8 (d). In the ResNet-like prediction model, the stage consists of a modified building block of ResNet with stride 2 and a modified building block of ResNet. This model follows most of the hyper-parameters used in [18].

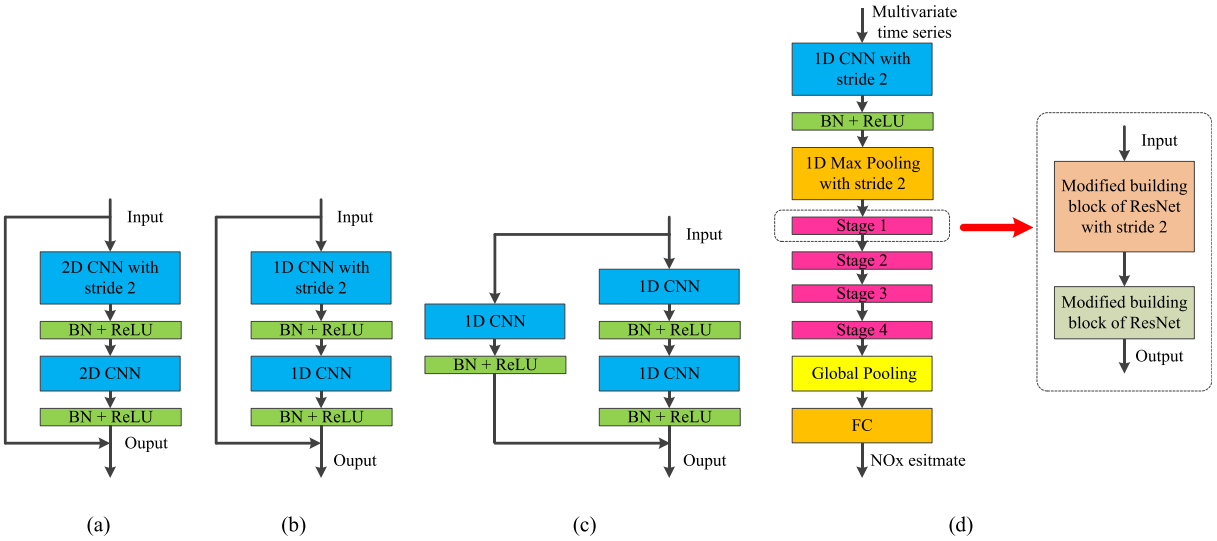


FIGURE 8. The overview of the ResNet-like prediction model. (a) Original building block of ResNet, (b) modified building block of ResNet, (c) modified building block of ResNet with stride 2, (d) the ResNet-like prediction model.

(3) Xception architecture can be considered as a linear stack of 2D separable CNN layers. The original building block of Xception is shown in Fig. 9 (a). We use the 1D separable CNN layers to replace the 2D separable CNN layers in the original building block of Xception. The modified building block of Xception is shown in Fig. 9 (b). The overall architecture of the Xception-like prediction model is shown in Fig. 9 (c). In the prediction model, the stage consists of a single modified building block of Xception. This model follows most of the hyper-parameters used in [19].

(4) ShuffleNetV1 is a light weight CNN architecture which also introduces the channel shuffle operator. Its building block is designed based on the depth-wise CNN layers and group CNN layers as shown in Fig. 10 (a). The modified building blocks of ShuffleNetV1 are shown in Fig. 10 (b) and (c). Fig. 10 (d) shows the overall architecture of the ShuffleNetV1-like prediction model. In the prediction model, the stage consists of a modified building block of ShuffleNetV1 with stride 2 and three modified building blocks of ShuffleNetV1. This model follows most of the hyper-parameters used in [20].

(5) The LSTM layer has been used to build the prediction model in [12] and [13]. The detailed architecture of the LSTM layer can be found in [29]. As shown in Fig. 11 (a), the LSTM-based prediction model consists of a LSTM layer and a FC layer. Because the number of units is an important hyper-parameter of the LSTM layer, we test the LSTM-based prediction models with a different number of units. There are 10 runs for each number of units. The detailed summary statistics of the prediction results are shown in Table 3. For example, we use LSTM-100 to denote an LSTM layer with 100 units. The average RMSE and the standard deviation of RMSE rise with the increase of the number of units. Among the three settings, LSTM-10 achieves the best results. Thus, we prefer to LSTM-10 for comparison.

TABLE 3. The summary statistics of the prediction results of the LSTM-based prediction models on the test set.

		Mean RMSE(mg/Nm^3)	Standard deviation of RMSE(mg/Nm^3)
LSTM-10	Side A	13.38	12.93
	Side B	9.81	5.87
LSTM-50	Side A	57.8	91.28
	Side B	47.75	77.89
LSTM-100	Side A	40.11	39.9
	Side B	48.82	60.51

(6) The bidirectional LSTM (BLSTM) layer, which is a variant of the LSTM layer, consists of two LSTM layers, one processing the input sequence forwards and the other one backward. The detailed architecture of the BLSTM layer can be found in [29]. As shown in Fig. 11 (b), the BLSTM-based prediction model consists of a BLSTM layer and a FC layer. Also, there are 10 runs for each number of units. The detailed summary statistics of the prediction results are shown in Table 4. It is clear that BLSTM-10 has the best results. Thus, we prefer to BLSTM-10 for comparison.

(7) Stacking multiple LSTM layers (or BLSTM layers) is a way to form a deeper model [30]. Based on the results in Table 3 and Table 4, we have 10 runs for some settings and the summary statistics of prediction results is shown in Table 5. It is clear that the prediction results are not improved by adding more layers. Thus, we do not use this class of models for comparison.

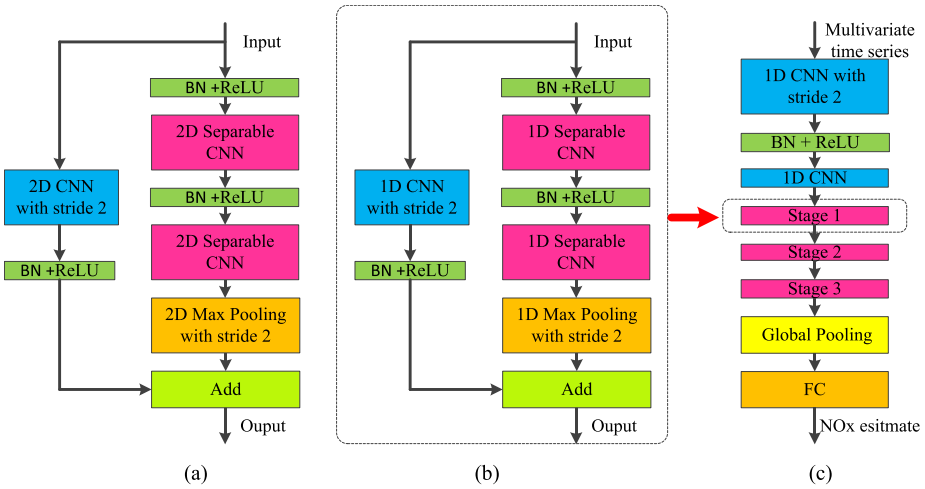


FIGURE 9. The overview of the Xception-like prediction model. (a) Original building block of Xception, (b) modified building block of Xception, (c) the Xception-like prediction model.

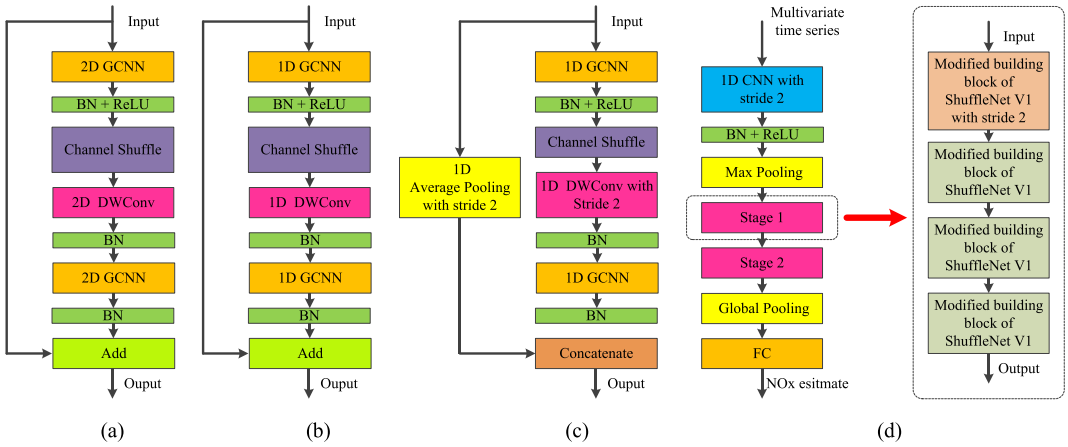


FIGURE 10. The overview of the ShuffleNetV1-like prediction model. (a) Original building block of ShuffleNetV1, (b) modified building block of ShuffleNetV1, (c) modified building block of ShuffleNetV1 with stride 2, (d) the ShuffleNetV1-like prediction model.

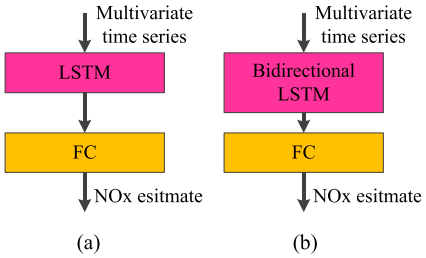


FIGURE 11. The overview of the prediction model based on LSTM or BLSTM. (a) The LSTM-based prediction model, (b) the BLSTM-based prediction model.

TABLE 4. The summary statistics of prediction results of the BLSTM-based prediction models on the test set.

		Mean RMSE(mg/Nm^3)	Standard deviation of RMSE(mg/Nm^3)
BLSTM-10	Side A	12.11	7.62
	Side B	13.27	7.05
BLSTM-50	Side A	21.18	18.63
	Side B	18.07	13.89
BLSTM-100	Side A	77.65	82.33
	Side B	76.04	66.92

(8) The DL-based model has also been constructed to estimate NO_X emission of coal-fired power plants in [11]. This model has exhibited satisfactory performance in the prediction accuracy and the most details can be found in [11].

The variations of RMSEs of the eight models among 30 runs are shown in Fig. 12. The minimum RMSE at side A is 0.25 mg/Nm^3 and achieved by our prediction model at the 7th run, and the minimum RMSE at side B is 0.28 mg/Nm^3 and achieved by our prediction model at the 5th run. For

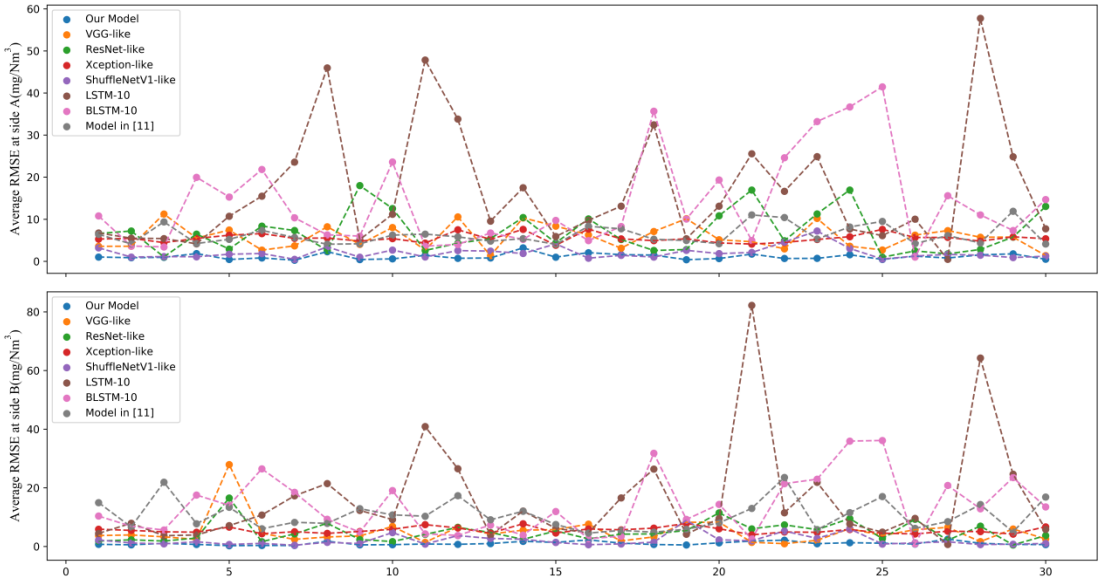


FIGURE 12. Comparison of RMSEs among different models.

TABLE 5. The summary statistics of prediction results of LSTM-10-10 and BLSTM-10-10 on the test set.

		Mean RMSE(mg/Nm^3)	Standard deviation of RMSE(mg/Nm^3)
LSTM-10-10	Side A	18.16	7.61
	Side B	20.58	8.41
BLSTM-10-10	Side A	17.66	8.44
	Side B	18.69	9.99

our prediction model, a smooth trend of RMSEs is observed, which can empirically demonstrate the good performance of our prediction models. The similar smooth trends of RMSEs can be observed for the Xception-like model and the ShuffleNetV1-like model. There exist some fluctuations among the trends of RMSEs for the VGG-like model and the ResNet-like model. For LSTM-10 and BLSTM-10, the significant fluctuations can be observed on RMSEs. The maximum RMSE at side A is 57.73 mg/Nm^3 and achieved by LSTM-10 at the 28th run, and the maximum RMSE at side B is 82.21 mg/Nm^3 and achieved by LSTM-10 at the 21th run. These significant outliers mean that LSTM-10 and BLSTM-10 sometimes fail to successfully learn effective data representations from the multivariate time series which contains more information. In other words, it is difficult to obtain acceptable results from the prediction model based on a single LSTM layer or a single BLSTM. Combined with the results in Table 5, the prediction performance can't improve by simply stacking more LSTM layers (or BLSTM layers). This is mainly due to the lack of practical guidelines of architecture design for organizing multiple LSTM layers or its variants. Although the model in [11] has smooth trend of RMSEs, the performance of this model is weaker than our model.

TABLE 6. The summary statistics of the prediction results of all prediction models on the test set.

		Mean RMSE(mg/Nm^3)	Standard deviation of RMSE(mg/Nm^3)
Our model	Side A	1.11	0.68
	Side B	1.06	0.59
VGG-like model	Side A	5.77	2.89
	Side B	5.05	4.81
ResNet-like model	Side A	6.94	4.91
	Side B	4.95	3.46
Xception-like model	Side A	5.49	1.02
	Side B	5.5	1.04
ShuffleNetV1-like model	Side A	2.03	1.42
	Side B	2.09	1.91
LSTM-10	Side A	16.79	14.36
	Side B	16.07	18.11
BLSTM-10	Side A	13.97	11.15
	Side B	13.87	10.05
Model in [11]	Side A	6.28	2.28
	Side B	10.57	5.08

Based on the above results, it is obvious that all deep CNN-based prediction models have better results than the LSTM-based prediction models. There are two reasons: (1) the building blocks in the CNN-based prediction models are carefully designed following the practical guidelines;

(2) during the training process of the deep CNN-based prediction model, multiple down-sampling processes are used to reduce the complexity of the data representations learned from the multivariate time series.

IV. CONCLUSION

In this study, a novel deep CNN-based model architecture has been developed for predicting NOx emissions from a 330MW tangentially coal-fired power plant boiler. The collected raw data are translated to the multivariate time series and the dataset for modeling NOx emissions is built. In order to efficiently process the multivariate time series samples, two basic building blocks are carefully designed based on the combination of the 1D CNN layer, the 1D separable CNN layer, the channel split operator and the channel shuffle operation. The overall prediction model architecture is developed mainly based on these two basic building blocks. The comparisons among the different prediction models have suggested that our proposed model has the best performance. In particular, the minimum RMSE of the test set at side A is 0.25 mg/Nm^3 and the minimum RMSE of the test set at side B is 0.28 mg/Nm^3. It also demonstrates that architecture design is important to build an accurate prediction model. There are two reasons that affect the accuracy of the prediction model: (1) the developed deep CNN-based prediction model depends on the sufficient data covering different operation conditions; (2) Recent advances in modern network architectures, which are also crucial components for other state-of-the-art networks, are adopted in our prediction model. The proposed model architecture has good potential to predict NOx emissions on similar pulverized coal-fired utility boilers with adequate data.

ACKNOWLEDGMENT

The authors thank Dr. Ren Li for his grammar review on this paper.

REFERENCES

[1] T. Boningari and P. G. Smirniotis, "Impact of nitrogen oxides on the environment and human health: Mn-based materials for the NOx abatement," *Current Opinion Chem. Eng.*, vol. 13, pp. 133–141, Aug. 2016.

[2] L.-G. Zheng, H. Zhou, K.-F. Cen, and C.-L. Wang, "A comparative study of optimization algorithms for low NOx combustion modification at a coal-fired utility boiler," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2780–2793, Mar. 2009.

[3] L. Ma, Q. Fang, P. Tan, C. Zhang, G. Chen, D. Lv, X. Duan, and Y. Chen, "Effect of the separated overfire air location on the combustion optimization and NOx reduction of a 600 MWe FW down-fired utility boiler with a novel combustion system," *Appl. Energy*, vol. 180, pp. 104–115, Oct. 2016.

[4] H. Zhou, K. Cen, and J. Fan, "Modeling and optimization of the NOx emission characteristics of a tangentially fired boiler with artificial neural networks," *Energy*, vol. 29, no. 1, pp. 167–183, Jan. 2004.

[5] Y. Lv, J. Liu, T. Yang, and D. Zeng, "A novel least squares support vector machine ensemble model for NOx emission prediction of a coal-fired boiler," *Energy*, vol. 55, pp. 319–329, Jun. 2013.

[6] P. Tan, J. Xia, C. Zhang, Q. Fang, and G. Chen, "Modeling and reduction of NOX emissions for a 700 MW coal-fired boiler with the advanced machine learning method," *Energy*, vol. 94, pp. 672–679, Jan. 2016.

[7] Y. Lv, J. Liu, and T. Yang, "Nonlinear PLS integrated with error-based LSSVM and its application to NOx modeling," *Ind. Eng. Chem. Res.*, vol. 51, no. 49, pp. 16092–16100, Dec. 2012.

[8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[9] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[11] F. Wang, S. Ma, H. Wang, Y. Li, and J. Zhang, "Prediction of NOX emission for coal-fired boilers based on deep belief network," *Control Eng. Pract.*, vol. 80, pp. 26–35, Nov. 2018.

[12] P. Tan, B. He, C. Zhang, D. Rao, S. Li, Q. Fang, and G. Chen, "Dynamic modeling of NOX emission in a 660 MW coal-fired boiler with long short-term memory," *Energy*, vol. 176, pp. 429–436, Jun. 2019.

[13] G. T. Yang, Y. N. Wang, and X. L. Li, "Prediction of the NOx emissions from thermal power plant using long-short term memory neural network," *Energy*, vol. 192, Feb. 2020, Art. no. 116597, doi: 10.1016/j.energy.2019.116597.

[14] P. Xie, M. Gao, H. Zhang, Y. Niu, and X. Wang, "Dynamic modeling for NOx emission sequence prediction of SCR system outlet based on sequence to sequence long short-term memory network," *Energy*, vol. 190, Jan. 2020, Art. no. 116482, doi: 10.1016/j.energy.2019.116482.

[15] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap, "Compressive transformers for long-range sequence modelling," 2019, *arXiv:1911.05507*. [Online]. Available: http://arxiv.org/abs/1911.05507

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[20] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[21] N. N. Ma, X. Y. Zhang, H. T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 116–131.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, Madison, WA, USA, Jun. 2010, pp. 807–814.

[24] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.

[25] L. Sifre, "Rigid-motion scattering for image classification," Ph.D. dissertation, CMAP, Ecole Polytechnique, Palaiseau, France, 2014.

[26] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467

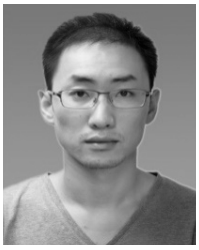
[27] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "CuDNN: Efficient primitives for deep learning," 2014, *arXiv:1410.0759*. [Online]. Available: http://arxiv.org/abs/1410.0759

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[29] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

[30] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks," 2017, *arXiv:1707.06799*. [Online]. Available: http://arxiv.org/abs/1707.06799

NAN LI received the B.Eng. degree in automation from the Nanjing University of Science and Technology, Nanjing, China, in 2006, and the M.Sc. degree in systems engineering and the Ph.D. degree in control theory and control engineering from North China Electric Power University, Beijing, China, in 2011 and 2017, respectively. He is currently a Lecturer with the School of Information and Electrical Engineering, Lu Dong University. His current research interests include machine learning and digital signal/image processing.



YONG HU received the Ph.D. degree in control theory and control engineering from North China Electric Power University, Beijing, China, in 2015. He is currently holding a postdoctoral position in energy engineering with the Mechanical Engineering College, North China Electric Power University. He has been engaged in the research of intelligent power generation operation control systems, modeling and optimal control of thermal power plant for a long time.