IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 18, 2020, accepted March 24, 2020, date of publication April 3, 2020, date of current version April 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985657

Deep Learning From Spatio-Temporal Data Using Orthogonal Regularizaion Residual CNN for Air Prediction

LEI ZHANG[1,2], (Member, IEEE), DONG LI[1], AND QUANSHENG GUO[1]

[1]School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
[2]Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing 100044, China

Corresponding author: Lei Zhang (lei.zhang@bucea.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871020, in part by the Key Science and Technology Plan Project of Beijing Municipal Education Commission of China under Grant KZ201810016019, in part by the Ministry of Housing and Urban Construction Science and Technology Project of China under Grant 2017-R2-018, in part by the High Level Innovation Team Construction Project of Beijing Municipal Universities under Grant IDHT20190506, and in part by the Practical Teaching Research Foundation of BUCEA under Grant J1712.

ABSTRACT Air pollution is harmful to human health and restricts economic development, so predicting when and where air pollution will occur is a challenging and important issue, especially in fields of urban planning, factory production and human activities. In this paper, we propose a deep Spatio-Temporal Orthogonal Regularization Residual CNN (ST-OR-ResNet) for air prediction. Deep Convolutional Neural Network (CNN) is presented to capture the complex spatio-temporal relation of the dynamic biased meteorological data. Residual learning is designed to avoid unpredictable oscillations when training the network and verifying errors. For the issue of characteristic statistical migration and saddle point proliferation in deep network, the orthogonality regularizations are designed to stabilize the back-propagation errors, utilizing various advanced analytical tools such as restricted isometry property without extra hassle. We then benchmark their effects on public real-world datasets to demonstrate that ST-OR-ResNet has better predictive performance than the state-of-the-art methods.

INDEX TERMS Deep learning, urban computing, spatio-temporal big data, orthogonal regularization, residual network.

I. INTRODUCTION

Nine out of ten people around the world breathe the polluted air, which kills seven million people a year, according to a report by WHO (World Health Organization) in 2018 [1]. Urban expansion exacerbates air pollution and triggers a series of imperative research [2]. Most studies on air pollution prediction are limited to meteorological models, with complex modeling and poor flexibility. Therefore, the research on using data mining and intelligent prediction to improve the urban environment and urban planning, has become an important hot scientific topic.

Environmental authorities face some challenges of establishing ground stations to monitor the air pollution, which are expensive (about $150,000 a site) [3]:

1) Due to the limited number and uneven distribution of monitoring stations, the meteorological data are biased. Therefore, reasonable allocation of resources, data sparsity and data noise should be considered;

2) Meteorological data have dynamic spatio-temporal properties. In terms of space, it is reflected in spatial distance and dimension, as well as different spatial granularity and urban structure. In terms of time, it is reflected in periodicity, tendency and proximity;

3) Heterogeneous meteorological data. The spatial distribution and temporal pattern of pollution sources are different, which are affected by local emission, regional traffic, meteorological conditions and other external factors.

In this paper, we propose an accurate and comprehensive data mining model and prediction method for these dynamic biased spatio-temporal big data to optimize the urban fine management.

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

There are many complicated relationships between measurable and immeasurable spatio-temporal data, which need to consider spatio-temporal reasoning process [4]. The existing machine learning and cloud computing cannot meet the requirements of accurate processing, while the deep neural network is better applied in urban computing, with its strong expression ability of network model and analysis of dynamic biased spatio-temporal big data. However, due to the lack of sophisticated theoretical support for deep learning, there are still many difficulties to be solved, for example how to conduct abstract modeling of various situations and concepts; how to solve the problem of gradient explosion and gradient disappearance; how to speed up training process to avoid falling into local optima.

Researchers have proposed a number of deep neural networks with desirable results. Wang *et al.* [5] designed a neural network which has two branches for attention box prediction (ABP) and aesthetics assessment (AA) on photo cropping. These two sub-networks were designed to share the same holographic convolutional feature map, and obtained better computational efficiency. Lai *et al.* [6] proposed a residual attentive learning network to predict dynamic eye-fixation maps. A composite attention module was integrated for enhancing the spatio-temporal saliency representation with multi-scale information. The composite attention mechanism learned local attentions as well as global attention priors for emphasizing the informative saliency features and filtering out the useless information, thus improving the spatio-temporal saliency representation efficiently.

On the basis of previous studies, we devise a deep *Spatio-Temporal Orthogonal Regularization Residual CNN (ST-OR-ResNet)* for air prediction, which is based on deep learning algorithm by combining the Convolutional Neural Network (CNN) with strong feature expression ability and the Recursive Neural Network (RNN) with strong long-time memory ability. Our approach makes several important new contributions:

- ➢ A Deep CNN is designed to capture the complex correlation of spatio-temporal data and the edge effect in spatial distribution based on deep learning algorithm.
- ➢ Three residual CNN subnets are integrated to couple the mapping relationship between the time dimension and spatial dimension, which is called *ST-ResNet*. These three subnets represent periodicity, tendency and proximity respectively.
- ➢ An *Orthogonal Regularization* (OR) algorithm is proposed to avoid characteristic statistical migration and saddle point proliferation without altering the original framework and achieve more accurate prediction.

The rest of this paper is organized as follows. Section II discusses the literature. Section III elaborates on our proposed network model and implementation details of ST-OR-ResNet. In Section IV, we compare and analyze the experiment results. Finally, we conclude the paper and outlook of the future works in Section V.

II. RELATED WORK

A. METHODS OF AIR PREDICTION

Generally, the dominant models and methods of air prediction are roughly divided into the following four types:

1) Mechanism models. Gibson *et al.* [7] applied Gaussian plume diffusion model for air prediction based on the point source and line source. Wu *et al.* [8] used Numerical Weather Prediction (NWP) model to estimate the Air Quality Index (AQI) in 2005-2006.

2) Remote Sensing and Geographic Information System (GIS) method. Kloog *et al.* [9] adopted a local regression model to predict the spatio-temporal distribution of PM2.5 based on Aerosol Optical Data (AOD) of the mid-Atlantic region in 2000-2008. Fang *et al.* [10] proposed a satellite-based real-time adaptive method to predict the concentrations of PM2.5, and the reliability of the proposed algorithm was tested by combining meteorological factors, land utilization and other multi-source auxiliary data. He and Huang [11] studied the spatio-temporal geographic weighted regression model to estimate AQI using AOD within a spatial resolution of 3km.

3) Machine learning model. Lyu *et al.* [12] used Bayesian hierarchical model to fit the statistical relationship between AOD and spatio-temporal data for the air pollution prediction. Hou *et al.* [13] applied the random forest algorithm on Spark cluster to realize the real-time prediction of a single monitoring site.

4) Deep learning model. Ong *et al.* [14] presented a deep neural network prediction model using the time series. Li *et al.* [15] combined geographic correlation with deep learning to prediction, using the satellite remote sensing data and the ground monitoring data.

To sum up, all kinds of air prediction models are effective. However, mechanism models need to take account of the pollution source diffusion, meteorological conditions and other factors, so the modeling is relatively complex. Remote sensing and GIS method, which use satellite remote sensing image data, is more stringent dependent on data, and the partial distribution of GIS data is unstable, lack of flexibility. Machine learning model is the widest approach; however, it has many restrictions on the quality and quantity of samples, and it needs to spend much time on data preprocessing and feature extraction. Deep learning models are commonly used in Natural Language Processing (NLP). However, deep learning mostly selects only one model for prediction, without simultaneously considering the feature abstraction and the contextual correlation along the time axis, so the performance is unstable.

B. DEEP LEARNING FOR SPATIO-TEMPORAL PREDICTION

Spatio-temporal data processing method for urban computing has been paid much attention in recent years, but there are still many difficulties. Support Vector Machine (SVM) cannot be applied on big data sets due to its computational density, robustness and other reasons [16]. The memory consumption of Decision Tree (DT) for big data processing is too large [17]. The learning speed of Feedforward Neural Network (FNN) is slow, etc. [18], [19].

Compared with these shallow neural networks, DNN provides modeling for the complex nonlinear system, and the extra layers have higher abstraction and learning capacity.

As a representative of DNN, CNN has been outstandingly applied in many fields, especially in computer vision, and it is often used to capture the spatial correlation of images [20], [21]. Ma *et al.* [22] adopted CNN to predict transportation speed. Li *et al.* [23] proposed deep residual learning on images and avoided vanishing/exploding gradients. Yao *et al.* [24] presented a deep multi-view network based on the combination of CNN and Long Short-Term Memory network (LSTM). LSTM can not only process single data (such as image), but also the entire data sequences. Zhang *et al.* [25] put forward spatio-temporal residual algorithm to predict the traffic flow, considering the time proximity, period and trend. While these methods do predict the historical timestamps, they do not explicitly model the chronological order dependencies.

C. FEATURE MAPPING FUNCTION OF CNN

In present urban computing methods, CNN is widely used to capture spatial similarity due to its performance on extracting high-latitude features among pixels. Yi *et al.* [26] proposed a method of space transform component and depth distributed convergence network, considering the spatial correlation of pollutants, heterogeneous data and weather conditions, etc. Jia *et al.* [27] modified CNN structure, introduced a wavelet transforms to replace the sub-sampling layer and redistributed the weight matrix adaptively to improve the oscillation phenomenon.

Outside the scope of urban computing, many scholars began to pursue the potential spatio-temporal attributes in multi-source data and learned new mapping relationships. Wu *et al.* [28] discussed that location-based services could influence people's trajectory. Considering the limitation of geographic space and the correlation of time series, a differential privacy location mechanism was proposed to improve the accuracy and effectiveness of trajectory privacy protection. Fan *et al.* [29] adopted a novel framework for human motion recognition with the local spatio-temporal feature behaviors. A local semantic Siamese framework was proposed to extract more robust features for high-speed visual object tracking, which was realized by adding a classification and residual channel attention block into the Siamese framework during the offline training [30].

Through the above works, CNN is widely applied in capturing the spatial features due to its performance on high dimension mapping correlations in pixels.

III. PROPOSED METHODOLOGY

Although some interesting deep neural networks were studies to examine the potential connections among the training instances, aiming to achieve a more powerful representation [31], [32], our focus is to predict exactly the future trends of dynamic spatio-temporal big data.

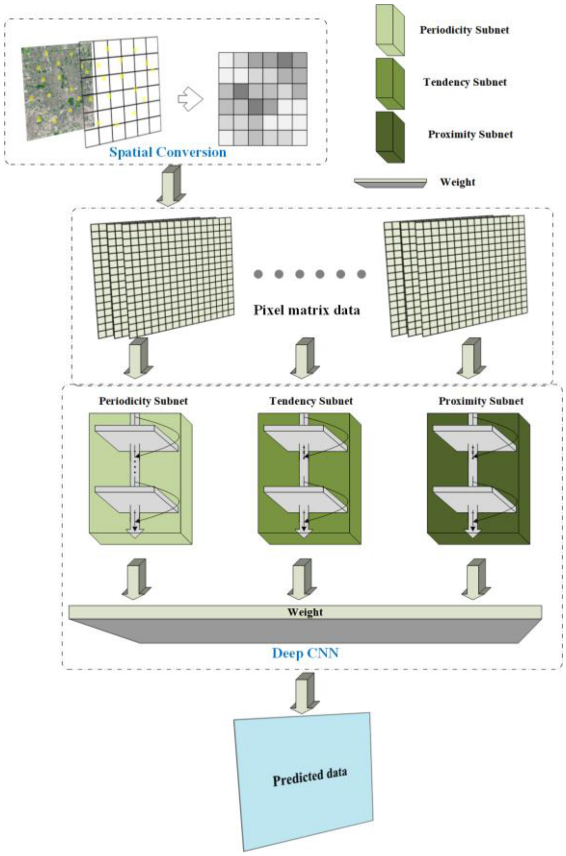


FIGURE 1. Overview of ST-OR-ResNet.

In this section, we elaborate the architecture of ST-OR-ResNet, shown in Fig.1. It contains two modules: Spatial Conversion and Deep CNN. The spatial conversion module uses "transform, fill and interpolation" to convert the real geographical data into sparse pixel matrix. Then, the pixel matrix is fed into Deep CNN to obtain the predicted result.

The detailed architecture and the proposed methodology are described below.

A. OVERALL STATEMENT

Firstly, we transform the raw data to low dimensions for capturing the temporal correlation and the internal dynamics. Then, the timeline is divided into three segments, representing latest time, near history and far history separately. Further, we use three CNN subnets to simulate three temporal characteristics (periodicity, tendency and proximity), which have the same structure, followed by a residual sequence. The whole architecture can capture the spatial dependence of the nearby area and the remote. Finally, we combine other context factors to present the impact on the predictive results, that is, the outputs of these three subnets are weighted together to capture deeper mapping relations by the activated function tanh() to [−1,1]. So that's our Deep CNN.

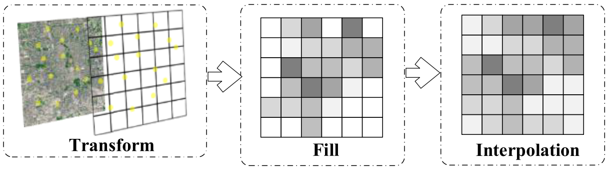


FIGURE 2. Procedure of spatial conversion.

B. SPATIAL CONVERSION

The purpose of spatial conversion is two-fold:

1) Converting the real meteorological data into pixel matrix, then put them into the CNN subnets;

2) Transforming the real-world physical space to an abstract space.

Firstly, space is divided into 32 × 32 grids according to latitude and longitude. Each grid is assigned a pixel value (that is, the monitored data at its location), and the grid outside the space is assigned a value of zero. Then, the spatio-temporal data is transformed to a pixel matrix. As shown in Fig.2, the colors represent different air quality levels.

Due to the limited number of sites, the pixel matrix is sparse and biased, which cannot reflect the global spatial distribution. Moreover, the geographical environment is not only determined by the local monitoring site but also affected by the adjacent area. Therefore, we develop a simple common spatial interpolation method, that is Inverse Distance Weighted Interpolation (IDWI) [33], to carry out data pretreatment. IDWI uses the data of the known grid to interpolate the data of the unknown grid.

$$\hat{y}(S_0) = \sum_{i=1}^{n} \lambda_i y(S_i) \tag{1}$$

Here, the value $\hat{y}(S_0)$ of the unknown grid S_0 is calculated by the weighted sum of the known value S_i with the weight λ_i.

Considering the known data of adjacent regions, IDWI assigns weights to the available readings of adjacent grids in each space according to the distance from the target, weights these weights to average, and then accumulates the interpolation of the vacant grids to finally obtain the average meteorological values in this region.

C. CNN SUBNET

With the in-depth study of spatio-temporal data, more and more scholars use deep CNN to deal with the spatio-temporal events due to its excellent feature representation ability.

Our proposed deep CNN can capture the mapping relationship between the mutual influence of air quality in further space and the impact of human activities, which shows CNN's powerful extract ability.

The deep CNN contains three CNN subnets (shown in Fig.3) to capture the time periodicity, tendency and proximity, respectively.

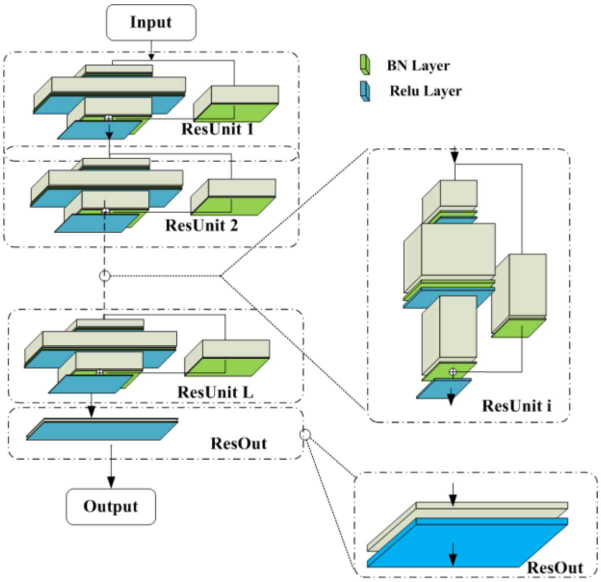


FIGURE 3. Architecture of CNN Subnet.

The previous pixel matrix is a 32×32 single-channel image as the input of convolution. The size of the convolutional kernel is $m \times m$, so m^2 values, which come from the $m \times m$ lower layer, generate a feature of higher layer through convolution, that means enough convolution can capture the spatial dependence of the global region.

In order to make the output size of subnet consistent with the ground truth, we design a residual output layer (ResOut) with only a convolutional kernel and no pooling layer. So, the output channel of the subnet is adjusted to 1.

Increasing the depth of CNN can improve the predictive performance; however, vanishing/exploding gradients are caused by multiple layers. The residual network effectively addresses these issues, and we also propose the novel normalization and orthogonality methodologies during the training processes.

1) RESNET

As is known to all, with the network depth increases, the accuracy reaches saturation and then declines rapidly, not caused by overfitting. Adding more layers to the appropriate depth model will cause higher training errors. With this in mind, we introduce the *residual CNN (ResNet)*.

The core of ResNet is that if a deep network can be trained into a shallow network and a set of invariant mapping networks during the training process, the obtained deep network will have no training errors. In other words, the residual network is essentially a shallow exponential set that avoids the disappearance/explosion of gradients.

Unlike [25], we improve the structure of ResNet:

- ➢ Delete the convolutional layer 1 and layer 2, the pixel matrix is input directly to the residual network;
- ➢ ResOut layer is designed to reduce the dimension and the calculation;

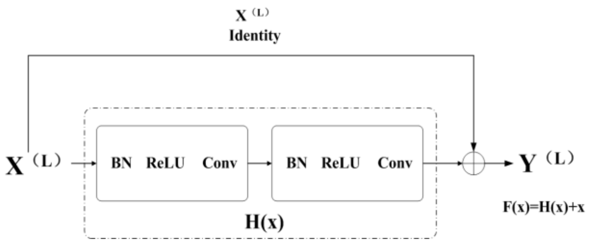


FIGURE 4. Schematic diagram of residual learning.

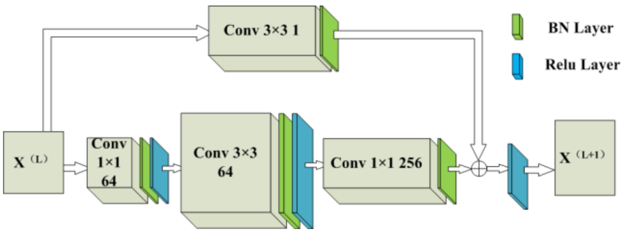


FIGURE 5. Structure of ResNet block.

➢ Add a convolution kernel of size 1 × 1 in shortcut connections to stabilize the training process.

ResNet becomes easier to optimize by adding shortcut connections, and a couple of layers with shortcut connections are called *residual blocks*.

The schematic diagram of residual learning is shown in Fig.4. The traditional input and output of the neural network are x and $F(x)$, and the training objective is $F(x) = H(x)$.

In our proposed ResNet, the input of the neural network is x, the output is $H(x) + x$, and the training target is $H(x) = F(x) - x$. $H(x)$ is called residual. It's easier to train $H(x)$ than $F(x)$.

Fig.5 shows one ResNet block, and a CNN subnet contains twelve blocks. In this ResNet unit, assuming the input pixel matrix size is 32 × 32.

In the first layer, the size of the convolutional kernel is 1×1, and the number of convolutional kernels is 64, which is followed by Batch Normalization (BN) and activation function ReLU()

In the second layer, the size of the convolutional kernel is 3 × 3, the number of convolutional kernels is also 64, and the following step is the same as the first layer.

In the third layer, the size of the convolutional kernel is 1 × 1, and the number of convolutional kernels is 256. The shortcut connection has only one convolutional kernel with the size of 3 × 3. After the third convolution, the residual convolution and BN are performed.

Finally, the output of ResNet is obtained through a fully connection layer, and its size is 32 × 32 × 256.

2) NORMALIZATION

Normalization approaches are indispensable components in deep learning, and they are often stacked after each convolutional layer or fully connection layer to improve the generalization ability [34].

The essence of deep learning process is the distribution of learning data; meanwhile, the data distribution affects the training speed and the generalization, that is why we add *Batch Normalization (BN)* into network structure.

Almost all data preprocessing uses the normalization, and it applies Gaussian standardization to subtract the mean of the representation, and then to divide the centered representation by the standard deviation.

BN is characterized by refactoring and introducing two learnable variables, and its implementation is shown in the following equations:

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \tag{3}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{4}$$

$$y_i \leftarrow \lambda \hat{x}_i + \beta \equiv \mathbf{BN}_{\lambda,\beta}(x_i) \tag{5}$$

where m denotes mini-batch's size, the mean and variance are calculated by (2) and (3) separately. Equation (4) defines the sample data normalization process. The transformation and scaling process are based on (5), which contains the learning parameters λ and β, allowing a new variable to have any mean and standard deviation.

The chain rule for back-propagation is shown below:

$$\frac{\partial \ell}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \lambda \tag{6}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}\right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\mathcal{B}})}{m} \tag{7}$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2}\left(\sigma_{\mathcal{B}}^2 + \epsilon\right)^{-3/2} \tag{8}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m} \tag{9}$$

$$\frac{\partial \ell}{\partial \lambda} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \cdot \hat{x}_i \tag{10}$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \tag{11}$$

Back-propagation calculates the gradient error, where is the i-th sample in mini-batch, is input error, so the gradient of mean and variance are calculated by (7) and (8). Equation (9) denotes the output error, and the scale parameters of BN are expressed in (10) and (11).

Due to the complex dynamics of nonlinear deep learning, even a proven mathematical theory cannot guarantee that multiple signals remain equidistant in practical applications. The depth causes the ''butterfly effect'' of exponential diffusion, and the nonlinearity leads to the uncertainty and randomness.

The state-of-the-art approaches [35]–[37] tried to stabilize the signal strength in one direction (forward /backward), so some studies found an alternative method for controlling signals in bio-directions. BN simplifies the problem by no longer controlling bio-direction signals but instead focusing on forwarding signals to reduce the internal covariant shifts in a hierarchical way.

3) ORTHOGONAL REGULARIZATION

We use ResNet to improve the training performance of Deep CNN, but Veit *et al.* [38] believed that ResNet was a relatively shallow exponential set, which could avoid vanishing/exploding gradients, instead of solving it directly.

Essentially, the performance gains of a network depend on its diversity, not its depth. As mentioned above, BN ensures the stability of the forward propagation, and the key to improve learning effectiveness on its reverse channel of error propagation. However, BN cannot guarantee the stable error rate in the back propagation. Therefore, we study the *Orthogonal Regularization (OR)* algorithm to replace the traditional weight decay regularization and use orthogonal rules to stabilize the back-propagation error [39].

Orthogonality is imposed on linear transformations in hidden layers. It reserves the energy and guarantees that the activation energy will not be amplified [40]. Therefore, it stabilizes the distribution of activations among layers [41], [42] and improves the generalization ability.

OR is ''plug-and-play'' added to CNNs, without any other modification. We derive and discuss several orthogonality regularizers.

A vector x maps to a vector y using linear transformation W, that is $y = W^T x$, W^T is the transposed matrix of W. If $||\mathrm{x}|| = ||\mathrm{y}||$, this transformation is norm-preserving, as well as the linear transformation matrix W is orthogonality, so the formula shows:

$$||\mathbf{y}|| = \sqrt{\mathbf{y}^T\mathbf{y}} = \sqrt{\mathbf{x}^T\mathbf{W}\mathbf{W}^T\mathbf{x}} = \sqrt{\mathbf{x}^T\mathbf{x}} = ||\mathbf{x}||\mathbf{iff}.\ \mathbf{W}\mathbf{W}^T = \mathbf{I} \tag{12}$$

For a convolutional layer $C \in S \times H \times C \times M$, where S, H, C, M are filter width, filter height, input channel number and output channel number, respectively.

Firstly, we reshape C into a matrix form $W^0 \in m^0 \times n^0$, where $m^0 = S \times H \times C$ and $n^0 = M$. The setting for regularizing convolutional layers follows [43], [44] to enforce orthogonality among filter layers.

Then, we discuss two novel orthogonality regularization, *Soft Orthogonality (SO)*Regularization and *Spectral Restricted Isometry Property (SRIP)* Regularization.

According to the previous studies [44]–[46], which suggested to adopt the Gram matrix, we define the SO regularization:

$$(\mathrm{SO}) \quad \lambda \left\| W^T W - I \right\|_F^2 \tag{13}$$

where λ is the regularization coefficient (the same below).

SO is a direct relaxation of the ''hard orthogonality'' assumption [47]–[49] under the standard Frobenius norm and a different weight attenuation term.I is the identity matrix, $W \in R^{m \times n}(m \times n$ shows the real spatial matrix) where $m = W \times H \times C$, $n = M$. $|| \cdot ||_F$ represents the Frobenius norm.

SO limits the orthogonality among filter layers and minimizes the correlation of learning functions, thus reducing redundancy and enhancing the diversity of filters, especially from the bottom filter [50].

Reviewing the Restricted Isometry Property (RIP) [51]–[53]:

$$(1-\delta)||\boldsymbol{\theta}||_2{}^2 \le ||A\boldsymbol{\theta}||_2{}^2 \le (1+\delta)||\boldsymbol{\theta}||_2{}^2 \tag{14}$$

Here, $||A\boldsymbol{\theta}||_2{}^2$ is the energy of the output signal, and $||\boldsymbol{\theta}||_2{}^2$ is the energy of the input signal. We rewrite the special RIP condition with $k = n$ in the form below:

$$\left| \frac{||Wz||^2}{||z||^2} - 1 \right| \le \delta_W, \quad \forall z \in \mathbb{R}^n \tag{15}$$

In order to increase orthogonality of W, it may minimize RIP constant δ_W in a special case $k = n$, which should be chosen as:

$$\sup_{z \in \mathbb{R}^n, z \neq 0} \left| \frac{||Wz||^2}{||z||^2} - 1 \right|$$

Therefore, we ultimately minimize the spectral norm of $W^T W - I$:

$$(\mathrm{SRIP}) \quad \lambda \cdot \sigma \left(W^T W - I \right) \tag{16}$$

Equation (16) is called *Spectrally Restricted Isometry Property (SRIP)*regularization [54], [55]. We iterate the following procedure a few times (such as two times) [39]:

$$u \leftarrow \left(W^T W - I\right) v,\ v \leftarrow \left(W^T W - I\right) u,\ \sigma\left(W^T W - I\right) \leftarrow \frac{||v||}{||u||} \tag{17}$$

The experiment is proceeded later, and we could observe consistent performance gains after applying SRIP, both the final accuracies and the convergences. SRIP reduces the computational cost from $O(n^3)$ to $O(mn^2)$ and is practically much faster for implementation.

D. CONVERGENCE METHODOLOGY

In this section, we use the parametric matrix-based convergence method to merge the three time-attributes of the trend, period and proximity in Fig.6 (a∼c), respectively.

As shown in Fig.6(a), the x-axis is twelve months in a year, and the y-axis is the monthly average AQI. It illustrates that

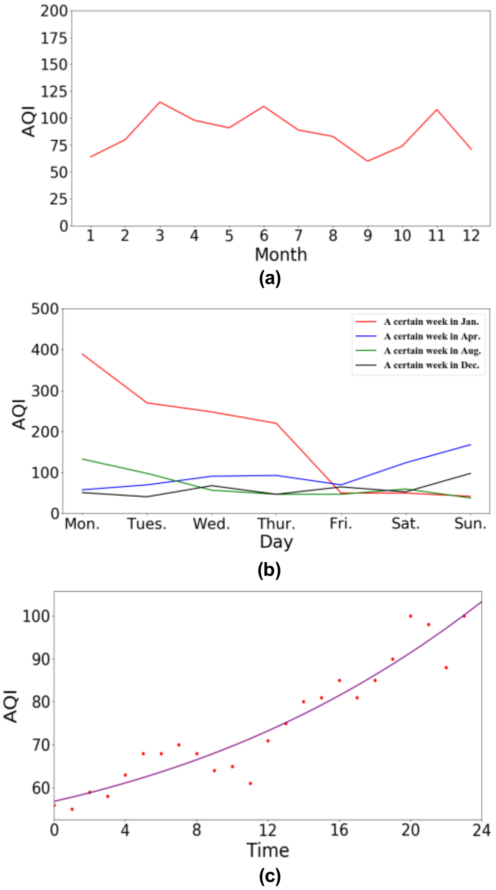
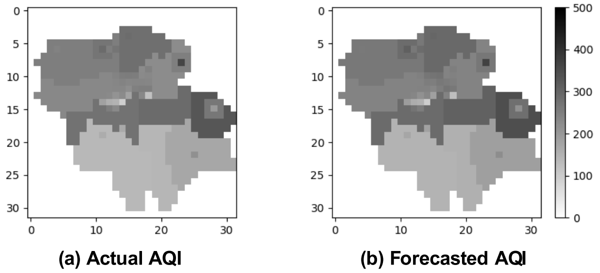


FIGURE 6. Monthly average AQI in one year (trend); (b). Daily AQI in a given week in different months (period); (c). Changing AQI at different times in a day (proximity).



(a) Actual AQI

(b) Forecasted AQI

FIGURE 7. a) Actual AQI (b) Forecasted AQI.

the change of air quality has a certain trend, that is, AQI in spring (February) and winter (December) is lower than which in summer (June).

The variation of air quality in some given week on different months is shown in Fig.6(b). The x-axis is the days of a certain week, and the y-axis is the daily average AQI. The weekly AQI has a certain period, but the curve is different corresponding to different months.

Fig.6(c) depicts the changing AQI at different times in a day. The curve is relatively smooth, indicating that AQI is not an instantaneous sudden change; in other words, the change of AQI has time proximity.

In summary, for the time series, AQI is affected by the trend, period and proximity, and the weight of the influence is different. Therefore, these three attributes are assigned different weights to connection:

$$X_{Res} = W_c * X_c^L + W_p * X_p^L + W_q * X_q^L \tag{18}$$

where X_c^L is periodic input of L layer, W_c is its weight matrix;X_p^L is proximity input,W_p is its weight matrix;X_q^L is trend and its weight matrix is W_q.

The final prediction at time t is:

$$\mathbf{X}_t = \tanh\left(\mathbf{X}_{Res}\right) \tag{19}$$

The optimal method adopts Adam algorithm [56] during the training process. The loss function is the Mean Square Error (MSE) in (20).

$$\mathcal{L}(\theta) = \|\mathbf{X}_t - \mathbf{X}_t\|_2^2 \tag{20}$$

IV. EXPERIMENT AND PERFORMANCE ANALYSIS

In this section, we use the real data sets from different regions in Beijing to evaluate the predictive performance of our proposed model. In the first part of the experiments, we compare our proposed *ST-OR-ResNet* with some baseline models. In the second part, we make two OR-model evaluations.

A. DATA AND ENVIRONMENT

The experiments are implemented on GPU server and Keras programming environment (TensorFlow).

We use real data sets collected by 42 official monitoring stations in Beijing from 2012 to 2018. Each record contains five pollutants: PM2.5, PM10, O_3, CO and SO_2. Then we calculate the integrated AQI according to the Chinese meteorological standard. The monitoring interval is one time per hour, and data is collected 24 times per day.

According to Section III.B, we divide 32 × 32 grids, and obtain the pixel matrix by (1). We predict the air quality over the next two days using our ST-OR-ResNet. Fig.7 shows the actual AQI and our forecasted result; that is, our model has a reliable measure.

Then, we compare our methods with baseline models, using *Accuracy*and *Rooted Mean Square Error (RMSE)*.

$$Acc = 1 - \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2}$$

where $y^{(i)}$ represents the real data, and $h\left(x^{(i)}\right)$ represents the predictive results; m is the number of grids in the matrix.

Keras uses the default parameters to initialize the learning parameters under a uniform distribution [57]. As depicted in section III.C, the residual blocks contain 64 convolutional kernels of size 1 × 1, 64 convolutional kernels of size 3 × 3, 256 convolutional kernels of size 1 × 1, and the ResOut layer has only one convolutional kernel with the size of 3 × 3.

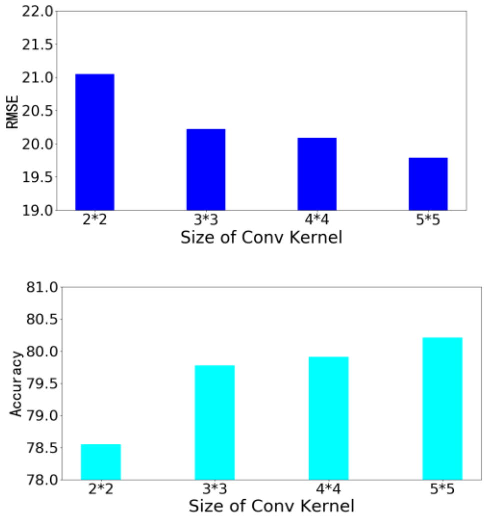


FIGURE 8. Evaluation on the changing size of conv kernel.

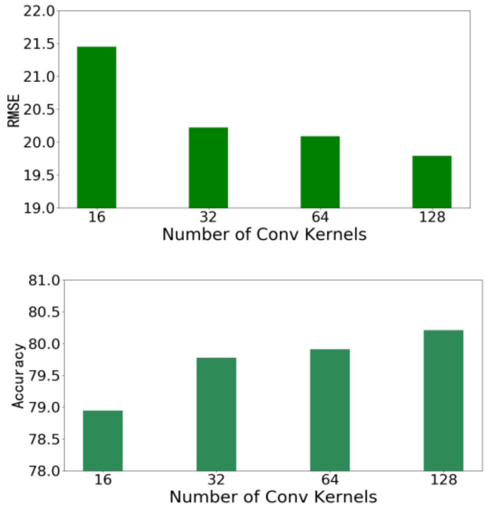


FIGURE 9. Evaluation on the changing number of conv kernels.

12-depth ST-ResNet includes twelve residual blocks and one ResOut layer. Adam algorithm is applied for optimization, the batch size is 100, and the initial learning rate is 0.001. The decay of the learning rate uses the function LearningRateScheduler ().

c (peroid) and q (trend) are constant for one day and one week. We set $l_c \in\{1,2,3,4,5\}$, $l_p \in\{1,2,3,4\}$, $l_q \in\{1,2,3,4\}$ and a 5-1 training-validating split data.

B. PERFORMANCE ANALYSIS

1) COMPARISON WITH BASELINE MODELS

In this part, we compare our proposed methods with some baseline models.

Ours: ST-ResNet without orthogonal regularization; ST-OR-ResNet_SO using *Soft Orthogonal* regularization; ST-OR-ResNet_SRIP using *Spectrally Restricted Isometry Property* regularization.

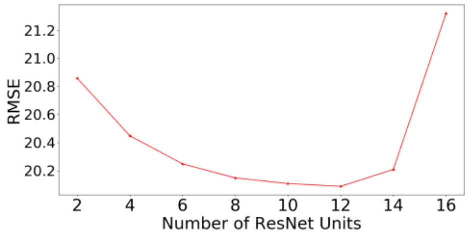


FIGURE 10. Different number of ResNet units impact on RMSE.

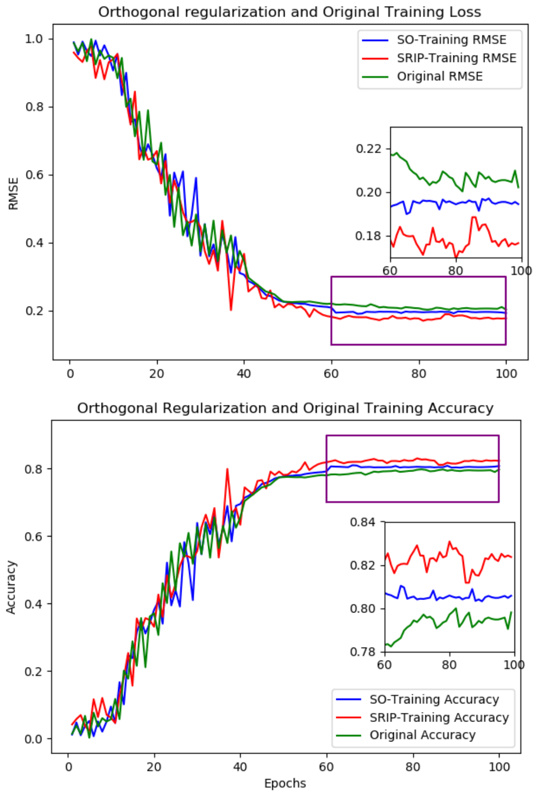


FIGURE 11. OR-model evaluations.

Baselines: LSTMs with 3, 6, 12, 24, and 48 layers are constructed respectively, Autoregressive Integrated Moving Average (ARIMA) [58] and Seasonal ARIMA (SARIMA) [59].

The results of RMSE and accuracy are shown in Table 1. We can see that the RMSE of LSTM is 26∼27% with a depth of 3∼48. ARIMA and SARIMA have the worst RMSE. Our methods perform better than these baseline models, especially ST-OR-ResNet_SRIP.

Further, we change the structure of ST-ResNet in two ways: the size of convolutional kernels and the number of convolutional kernels. The comparisons are shown in Fig.8 and Fig.9.

Fig.8 depicts the relationship between the size of the convolutional kernel and the error rate; that is, the error rate decreases as the size of the convolutional kernel increases.

The difference in accuracy is small between 3×3 and 4×4 convolutional kernel, and 5×5 convolutional kernel has the lowest error rate. However, the expansion of the convolutional

TABLE 1. Comparison with different baselines.

Model	Type	RMSE	Accuracy
LSTM	LSTM-3	27.33	72.67
	LSTM-6	27.94	72.06
	LSTM-12	26.50	73.50
	LSTM-24	27.10	72.90
	LSTM-48	26.79	73.21
ARIMA		31.29	68.71
SARIMA		28.88	71.12
Ours	ST-ResNet	20.20	79.80
	ST-OR-ResNet_SO	19.20	80.80
	ST-OR-ResNet_SRIP	17.60	82.40

kernel to 5 × 5 increases the calculation, running time, and the number of hyperparameters, as well as decreases the optimization.

Analogously, Fig.9 shows the relationship between the number of convolutional kernels and the error rate.

Fig.10 shows that the RMSE of ST-ResNet decreases with the increasing depth, however, when the depth reaches a certain level, the inflection point occurs, and the RMSE suddenly rises.

2) OR-MODEL EVALUATIONS

In this part, we apply two orthogonal regularization methods on ST-ResNet and employ model configurations on real datasets. The convolutional kernel regularization learning rates are set to 0.01 and 0.0001, respectively.

From the experiments, we observe that the complete replacement of the l^2 weight attenuation with the orthogonal regularizer can accelerate and stabilize the training process at the beginning.

The regularization coefficient is λ which is expressed in (16), and it plays an important effect on the training process. For λ, we start with 10^{-8}, increase to 5×10^{-4} after 20 iterations in SO model.

To SRIP regularizer, we maintain the initial λ throughout the whole training process. The reason is that SRIP has a stronger effect on forcing $W^T W$ close to I, so it is insensitive to λ.

As shown in Fig.11, applying orthogonal regularization can improve the optimization, and it has a strong positive impact at the early training stage (not just initialization). However, when the training process approaches the end, the effect is weaker.

From the experimental results, SRIP is the best practice choice, and it consistently performs in achieving the highest precision and acceleration/stability training curves.

V. CONCLUSION AND FUTURE WORK

In this paper, we studied the spatio-temporal data predictive mechanism for air pollution and proposed a deep learning model based on the convolution residual approach and orthogonality regularization algorithm. Our solution, *ST-OR-ResNet*, was used to optimize the prediction accuracy and accelerate the training efficiency. Our method not only considered the coupling of time and space but also combined with the complex mapping relationship at high latitude. In almost all times, the novel SRIP regularizer exceeded all else consistently and remarkably. Using real data sets, we demonstrated that our proposed methodology outperforms the existing baselines especially in prediction accuracy and generalization.

In the future, we will study how to enrich this research by taking the multi-source heterogeneous spatio-temporal big data and various external factors into account, e.g., points of interests and emergency. In addition, we will expand our research into other relevant fields of urban computing.

REFERENCES

[1] "Air pollution overview." Accessed: 2019. [Online]. Available: https://www.who.int/news-room/

[2] H. Akimoto, "Global air quality and pollution," *Science*, vol. 302, no. 5651, pp. 1716–1719, 2003.

[3] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2013, pp. 1436–1444.

[4] L. Zhao, Q. Sun, J. Ye, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Multi-task learning for spatio-temporal event forecasting," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2015, pp. 1503–1512.

[5] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1531–1544, Jul. 2019.

[6] Q. Lai, W. Wang, H. Sun, and J. Shen, "Video saliency prediction using spatiotemporal residual attentive networks," *IEEE Trans. Image Process.*, vol. 29, no. 8, pp. 1113–1126, Aug. 2020.

[7] M. D. Gibson, S. Kundu, and M. Satish, "Dispersion model evaluation of PM2.5, NO_x and SO_2 from point and major line sources in nova scotia, canada using AERMOD Gaussian plume air dispersion model," *Atmos. Pollut. Res.*, vol. 4, no. 2, pp. 157–167, Apr. 2013.

[8] L. Cordero, Y. Wu, B. M. Gross, and F. Moshary, "Assessing satellite AOD based and WRF/CMAQ output PM2.5 estimators," *Proc. SPIE*, vol. 8723, May 2013, Art. no. 872319.

[9] I. Kloog, F. Nordio, and B. A. Coull, "Incorporating local land use regression and satellite aerosol optical depth in a hybrid model of spatiotemporal PM2.5 exposures in the mid-Atlantic states," *Air Pollut. Sci. Technol.*, vol. 46, no. 21, pp. 11913–11921, 2012.

[10] X. Fang, B. Zou, X. Liu, T. Sternberg, and L. Zhai, "Satellite-based ground PM2.5 estimation using timely structure adaptive modeling," *Remote Sens. Environ.*, vol. 186, pp. 152–163, Dec. 2016.

[11] Q. He and B. Huang, "Satellite-based mapping of daily high-resolution ground PM2.5 in China via space-time regression modeling," *Remote Sens. Environ.*, vol. 206, pp. 72–83, Mar. 2018.

[12] B. Lv, Y. Hu, H. H. Chang, A. G. Russell, and Y. Bai, "Improving the accuracy of daily PM2.5 distributions derived from the fusion of ground-level measurements with aerosol optical depth observations, a case study in north China," *Environ. Sci. Technol.*, vol. 50, no. 9, pp. 4752–4759, May 2016.

[13] J. X. Hou, Q. Li, and Y. J. Zhu, "Real-time PM2.5 prediction system based on random forest," *Sci. Surveying Mapping*, vol. 42, no. 1, pp. 1–6, 2017.

[14] B. T. Ong, K. Sugiura, and K. Zettsu, "Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1553–1566, Aug. 2016.

[15] T. Li, H. Shen, Q. Yuan, X. Zhang, and L. Zhang, "Estimating ground-level PM2.5by fusing satellite and station observations: A geo-intelligent deep learning approach," *Geophys. Res. Lett.*, vol. 44, no. 23, pp. 11985–11993, Dec. 2017.

[16] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[17] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, Sep. 1997.

[18] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Netw.*, vol. 2, no. 6, pp. 459–473, Jan. 1989.

[19] L. Bertinetto, F. João Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 523–531.

[20] A. Krizhevsky and I. G. E. Sutskever Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[21] D. B. Bert, X. Jia, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 667–675.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818–827, 2017.

[23] D. Li, L. Zhang, M. Z. Guo, and Y. L. Liu, "Air quality prediction based on spatiotemporal convolution residual network," *Comput. Technol. Develop.*, vol. 56, no. 6, pp. 43–51, 2020.

[24] H. Yao, F. Wu, and J. Ke, "Deep multi-view spatial-temporal network for taxi demand prediction," *Statistics*, vol. 2, no. 4, pp. 468–475, 2018.

[25] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[26] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 965–973.

[27] F. Jia and L. Zhang, "Estimation of population density based on improved convolutional neural network," *Comput. Technol. Develop.*, vol. 29, no. 2, pp. 77–80, 2019.

[28] Y. C. Wu, H. Chen, and S. Y. Zhao, "A differential privacy trajectory protection mechanism based on spatiotemporal correlation," *Comput. J.*, vol. 41, no. 2, pp. 309–322, 2018.

[29] X. J. Fan, S. B. Xuan, and F. Tang, "Human motion recognition based on hybrid spatiotemporal feature descriptors," *Comput. Technol. Develop.*, vol. 28, no. 2, pp. 98–118, 2018.

[30] Z. Liang and J. Shen, "Local semantic siamese networks for fast tracking," *IEEE Trans. Image Process.*, vol. 29, no. 12, pp. 3351–3364, Dec. 2019.

[31] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3516–3527, Jul. 2019.

[32] X. Dong and J. Shen, "Triplet loss in Siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 459–474.

[33] G. Y. Lu and D. W. Wong, "An adaptive inverse-distance weighting spatial interpolation technique," *Comput. Geosci.*, vol. 34, no. 9, pp. 1044–1055, Sep. 2008.

[34] P. Luo and Z. Peng, "Differentiable dynamic normalization for learning deep representation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 4203–4211.

[35] D. Arpit, Y. Zhou, B. U. Kota, and V. Govindaraju, "Normalization propagation: A parametric technique for removing internal covariate shift in deep networks," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1168–1176.

[36] D. Sussillo and L. F. Abbott, *Random Walk Initialization for Training Very Deep Feedforward Networks*. Mountain View, CA, USA: Google Inc, 2014, pp. 1–10.

[37] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell, "Data-dependent initializations of convolutional neural networks," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–12. [Online]. Available: http://arxiv.org/abs/1511.06856

[38] A. Veit, M. Wilber, and S. Belongie, "Residual networks are exponential ensembles of relatively shallow networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2016, pp550-558.

[39] N. Bansal, X. Chen, and Z. Wang, "Can we gain more orthogonality regularizations in training deep CNNs?" in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2018, pp4266-4276.

[40] J. Zhou, M. N. Do, and J. Kovacevic, "Special paraunitary matrices, Cayley transform, and multidimensional orthogonal filter banks," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 511–519, Feb. 2006.

[41] P. Rodríguez, J. Gonzalez, G. Cucurull, J. M. Gonfaus, and X. Roca, "Regularizing CNNs with locally constrained decorrelations," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–11. [Online]. Available: http://arxiv.org/abs/1611.01967

[42] G. Desjardins, K. Simonyan, and R. Pascanu, "Natural neural networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 2071–2079.

[43] D. Xie, J. Xiong, and S. Pu, "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6176–6185.

[44] L. Huang, X. Liu, B. Lang, A. W. Yu, and B. Li, "Orthogonal weight normalization: Solution to optimization over multiple dependent Stiefel manifolds in deep neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2017, pp. 3271–3278.

[45] R. Balestriero and R. Baraniuk, "A spline theory of deep learning," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 374–383.

[46] R. Balestriero and R. Baraniuk, "Mad max: Affine spline insights into deep learning," 2018, *arXiv:1805.06576*. [Online]. Available: http://arxiv.org/abs/1805.06576

[47] M. Harandi and B. Fernando, "Generalized BackPropagation, Étude de cas: Orthogonality," 2016, *arXiv:1611.05927*. [Online]. Available: http://arxiv.org/abs/1611.05927

[48] M. Ozay and T. Okatani, "Optimization on submanifolds of convolution kernels in CNNs," 2016, *arXiv:1610.07008*. [Online]. Available: http://arxiv.org/abs/1610.07008

[49] H. Xu, Z. Wang, H. Yang, D. Liu, and J. Liu, "Learning simple thresholded features with sparse support recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 970–982, Apr. 2020.

[50] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 2217–2225.

[51] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[52] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[53] Z. Wang and J. Yang, *Sparse Coding and its Applications in Computer Vision*. Singapore: World Scientific, 2016.

[54] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," 2017, *arXiv:1705.10941*. [Online]. Available: http://arxiv.org/abs/1705.10941

[55] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. 26th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–26. [Online]. Available: http://arxiv.org/abs/1802.05957

[56] S. Andradóttir, "A method for discrete stochastic optimization," *Manage. Sci.*, vol. 41, no. 12, pp. 1946–1961, Dec. 1995.

[57] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

[58] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

[59] S. C. Hillmer and G. C. Tiao, "An ARIMA-model-based approach to seasonal adjustment," *J. Amer. Stat. Assoc.*, vol. 77, no. 377, pp. 63–70, 2012.

LEI ZHANG (Member, IEEE) was born in Jinan, Shandong, China, in 1981. She received the B.S. degree in communication engineering from Shandong University, Shandong, in 2004, and the Ph.D. degree in communication and information system from the Beijing University of Post and Telecommunications, Beijing, China, in 2009.

From 2009 to 2017, she was an Assistant Professor, and since 2017, she has been an Associate Professor with the Department of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. She has held a visiting position at Arizona State University, USA, in 2019. She is currently the Dean of the School of Computer Science and Technology, Beijing University of Civil Engineering and Architecture. Her main research interests include urban computing, machine learning, the intelligent Internet of Things, and data mining.

Prof. Zhang is a member of ACM. She received many flagship honors, including the Program for Beijing Youth Talents from Beijing Municipal Education Commission.

DONG LI was born in Xingtai, Hebei, China, in 1996. He received the B.S. degree in electric engineering from Hebei University, Hebei, in 2018. He is currently pursuing the M.S. degree with the Department of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. His research interests include urban computing, deep learning, and data mining.

QUANSHENG GUO was born in Zhangjiakou, Hebei, China, in 1996. He received the B.S. degree in computer engineering from the Hebei University of Architecture, Hebei, in 2018. He is currently a Graduate with the Department of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. His main research fields contain data mining and machine learning.