Deep-Learning-Based Spatial–Temporal Channel Prediction for Smart High-Speed Railway Communication Networks

Tao Zhou, *Member, IEEE*, Haitong Zhang, Bo Ai, *Fellow, IEEE*, Chen Xue, and Liu Liu, *Member, IEEE*

***Abstract*— Intelligent channel prediction plays a key role in artificial intelligence (AI)-optimized or AI-native communication networks for smart high-speed railways (HSRs). This paper investigates the spatial-temporal prediction of channel state information (CSI) and channel statistical characteristics (CSCs) based on deep-learning (DL) for the future smart HSR communication network. A propagation-graph simulation method is used to generate datasets of CSI and CSCs for massive multiple-input multiple-output (mMIMO) channels in a HSR cutting scenario, and realistic channel measurements are used to validate the datasets. Then, single-step ahead and multi-step ahead prediction problems are formulated with the consideration of both spatial and temporal information hidden in the datasets. By exploiting the temporal and spatial correlations of the HSR mMIMO channel, a novel spatial-temporal channel prediction model that combines the convolutional neural network (CNN) and convolutional long short-term memory (CLSTM) is proposed and called as Conv-CLSTM. Moreover, the hyper-parameters of the Conv-CLSTM model are determined by autocorrelation and similarity analysis and cross-validation. Finally, the performance of the Conv-CLSTM model is evaluated in terms of prediction accuracy and space and time computational complexity, and is compared with classical DL models. The evaluation results show that the proposed model has high prediction accuracy but acceptable computational complexity.**

***Index Terms*— Channel prediction, smart railways, deep-learning, AI-native networks, channel state information.**

I. Introduction

SMART high-speed railways (HSRs) have become a significant direction of future HSR development, which aims to improve the railway operation safety, economical efficiency, convenience, comfortableness, and environmental friendliness. The smart HSRs will fully utilize new generation information technologies such as big data, artificial intelligence (AI), Internet of Things (IoT), next-generation mobile communications, and cloud computing [1]. The smart HSR communication network is one of essential parts in the entire smart HSR system, which will be used for satisfying higher and higher communication requirements of railway operation and passenger experience in terms of data rate, capacity, latency, and reliability. Undoubtedly, traditional dedicated railway networks are unable to undertake this task. Fifth-generation (5G) technologies such as massive multiple-input multiple-output (mMIMO), millimeter wave (mmWave), and ultrareliable low latency communications (URLLC) have been recommended for the smart HSRs and an innovative 5G for railway (5G-R) network was discussed [2]–[4]. The 5G-R network can be possible to meet the requirements of high throughput, high reliability, and low latency for smart HSR communications. Therefore, it has been regarded as a candidate for the future smart HSR communication network.

Recently, AI has become popular in the computer field due to its great success in computer vision, natural language processing, automatic speech recognition, and wireless communications [5]. The AI technologies will play an important role in the smart HSR communication network, which can be mainly reflected in two aspects. On the one hand, the AI can be used as an optimization tool to improve the performance of 5G-R network. On the other hand, the AI can be deeply integrated in the smart HSR communication network, which means the AI technologies will natively exist for the network [6]. With its learning, prediction, and decision-making capabilities, the AI will significantly improve the efficiency, reliability, real-time performance, and security of the smart HSR communication network [7].

Either in the AI-optimized 5G-R network or in the AI-native HSR network, intelligent channel prediction is one of most important and indispensable technologies. The channel prediction aims to forecast future channel state information (CSI) and

Manuscript received June 22, 2021; revised October 10, 2021; accepted December 19, 2021. Date of publication January 10, 2022; date of current version July 12, 2022. This work was supported in part by the Beijing Natural Science Foundation under Grant L212030 and Grant 4212006; in part by the National Natural Science Foundation of China under Grant 62071031, Grant 6196113039, Grant 61725101, and Grant U1834210; in part by the State Key Laboratory of Rail Traffic Control and Safety under Grant RCS2020ZT010 and Grant RCS2019ZZ007; in part by the Open Research Fund through the National Mobile Communications Research Laboratory, Southeast University, under Grant 2021D01; and in part by the National Key R&D Program of China under Grant 2020YFB1804901. The associate editor coordinating the review of this article and approving it for publication was C. Huang. *(Corresponding author: Tao Zhou.)*

Tao Zhou is with the Institute of Broadband Wireless Mobile Communications, Beijing Jiaotong University, Beijing 100044, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: taozhou@bjtu.edu.cn).

Haitong Zhang, Chen Xue, and Liu Liu are with the Institute of Broadband Wireless Mobile Communications, Beijing Jiaotong University, Beijing 100044, China (e-mail: 19120166@bjtu.edu.cn; 18120152@bjtu.edu.cn; liuliu@bjtu.edu.cn).

Bo Ai is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China, also with the Henan Joint International Research Laboratory of Intelligent Networking and Data Analysis, Zhengzhou University, Zhengzhou 450001, China, also with the Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518066, China, and also with the Frontiers Science Center for Smart High-Speed Railway System, Beijing 100044, China (e-mail: boai@bjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3139384.

Digital Object Identifier 10.1109/TWC.2021.3139384

channel statistical characteristics (CSCs) in advance without spending extra radio resources [8]. This is different with the traditional channel estimation which could yield the outdated CSI and CSCs due to the rapid channel variation in HSR networks. With the assist of predicted CSI and CSCs, adaptive transmission technologies, such as transmission power control, pre-equalization, adaptive modulation and coding, and intelligent decisions, e.g., intelligent resource allocation and scheduling can be efficiently realized. Thus, the performance of the network will be significantly improved. The channel prediction approaches can be generally divided into two categories, statistical prediction and intelligent prediction (or AI based prediction). The statistical prediction was usually based on the auto-regressive (AR) model and the parametric model [9]–[11]. However, this traditional channel prediction method is fossilized, leading to a gap between these models and real channels. The AI based prediction as a data-driven approach is more flexible and accurate, which has attracted more and more attentions [12].

Machine-learning (ML) has been widely applied to the prediction of CSI and CSCs. Authors in [13] employed an artificial neural network (ANN) combined with chirp Z-transform to achieve the CSI prediction. A support vector machine (SVM) was used to forecast the CSI under double-Rayleigh fading channels in vehicle to vehicle (V2V) communication systems [14]. In addition to the CSI prediction, the ANN was also utilized for the path loss (PL) prediction in rural and railway scenarios [15], [16]. Two ML algorithms, random forest and k-nearest neighbor (KNN), were exploited to build the PL prediction model for unmanned aerial vehicle (UAV) scenarios [17]. In [18], an ANN based channel model was proposed to predict the CSCs such as received power, root mean square (RMS) delay spread (DS), and RMS angle spread (AS) according to the input parameters including transmitter (Tx) and receiver (Rx) coordinates, Tx-Rx distance, and carrier frequency. The ANN based channel prediction model was validated by mmWave channel measurement data in HSR station scenarios in terms of large-scale and small-scale channel parameters [19].

Deep-learning (DL) that possesses more powerful learning capacity has been increasingly used for the channel prediction. By leveraging the strong time series prediction capability of recurrent neural network (RNN), authors in [20] applied the RNN to build a frequency-domain channel predictor for wide-band communications. It has been shown that this predictor was effective to combat the outdated CSI with reasonable computational complexity and intrinsically enabled multi-step prediction. In [21], a MIMO channel predictor built on the deep RNN that incorporates long short-term memory (LSTM) or gated recurrent unit (GRU) was proposed to forecast the CSI. Convolutional neural network (CNN) was employed to predict the CSCs of any sub-channels for mmWave mMIMO systems in indoor scenarios [22]. As for the used CNN, the input parameters were the location information of Tx and Rx antennas. Compared with the ML algorithms, the DL networks such as CNN, RNN, and LSTM have the ability to dig the spatial or temporal information hidden in the data, which are more suitable for the spatial or temporal channel prediction. As for the time-varying mMIMO channel in the smart HSR communication network, both spatial and temporal channel prediction should be considered [23]. In fact, the joint spatial and temporal prediction depending on DL has been investigated in some forecasting fields such as rainfall intensity prediction [24], crowd flow prediction [25], and demand prediction in cellular networks [26]. However, to the best of our knowledge, the spatial-temporal channel prediction based on DL in HSR scenarios is still missing.

To fill the aforementioned research gaps, this paper aims to investigate the DL based spatial-temporal channel prediction for the future smart HSR communication network. The major contributions and novelties of this paper are as follows.

1) Datasets related to the CSI and CSCs of mMIMO channels in a HSR cutting scenario are generated based on a propagation-graph simulation method and validated according to field channel measurement results. Moreover, time-varying mMIMO channel prediction problems considering both spatial and temporal information are formulated for single-step ahead and multi-step ahead predictions.

2) A novel spatial-temporal channel prediction model is proposed by combining CNN and convolutional LSTM (CLSTM), called as Conv-CLSTM. The Conv-CLSTM model is able to capture the spatial-temporal information hidden in the HSR mMIMO channel datasets, including spatial correlation at both Tx and Rx sides and temporal correlation.

3) Theoretical analysis involving autocorrelation analysis and similarity analysis are used to determine two important parameters such as time window length and convolution kernel size in the Conv-CLSTM model. In addition, cross-validation is performed to verify the results of theoretical analysis and further decide the other optimal hyper-parameters.

4) The performance of the Conv-CLSTM model is evaluated in terms of prediction accuracy and computational complexity from both space and time aspects, and is compared with existing ML and DL models such as SVM, CNN, RNN, LSTM, etc. It is shown that the proposed model has high prediction accuracy but acceptable computational complexity.

The remainder of this paper is outlined as follows. Section II describes the dataset generation and prediction problem formulation for HSR mMIMO channels. In Section III, the novel deep-learning based spatial-temporal prediction model for HSR mMIMO channels is proposed. Then, the hyper-parameters determination of the proposed model is presented in Section IV. In addition, the performance of the proposed model is evaluated in Section V. Finally, conclusions are drawn in Section VI.

II. Datasets and Prediction Problems for HSR mMIMO Channels

A. Propagation-Graph-Based HSR mMIMO Channel Simulation

Due to the strict measurement limitation on HSR, e.g., the coordination of high-speed test train and the usage of train-mounted antennas, it is quite difficult to conduct field measurements for HSR channels, especially for HSR MIMO channels [27]. To generate sufficient channel data in different

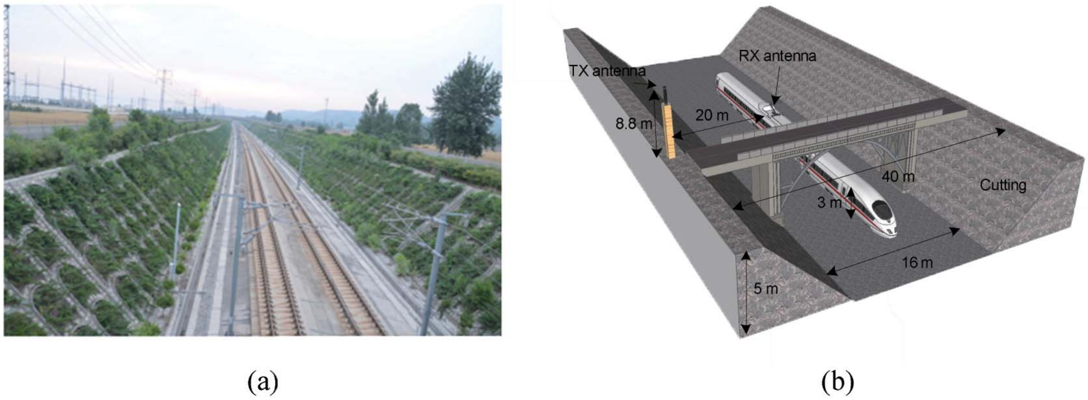

Fig. 1. The measured cutting scenario on Zhengzhou-Xi'an HSR in China.

configuration, we have to resort to channel simulation. There have been three popular methods for HSR channel simulation, including ray-tracing [28], [29], geometry-based stochastic model (GBSM) [30]–[32], and propagation-graph [33], [34]. The ray-tracing method is more suitable to simulate the static channel, whereas the GBSM and propagation-graph approaches can support the time-varying channel simulation. Compared with the GBSM, the propagation-graph method is more effective to describe actual propagation environments and has lower simulation complexity. Thus, the propagation-graph based channel simulation is employed here.

In the propagation-graph, vertices are used to represent Tx antennas, Rx antennas, and scatterers, while edges indicate propagation conditions between vertices, involving line of sight (LOS) and non-line of sight (NLOS) conditions. There are four types of edges, such as Tx-Rx edge, Tx-scatterer edge, scatterer-Rx edge, and scatterer-scatterer edge. A signal propagates between these edges depending on relevant propagation mechanisms, which will experience the space-time-frequency channel fading. When the propagation mechanisms are assumed to be linear, this channel effect can be simulated by the multiplication with transfer functions. Applying the propagation-graph based channel simulation, channel frequency responses (CFRs) can be generated. A detailed simulation procedure has been introduced in [34].

It has been reported that there exist over ten propagation scenarios for HSRs [35]. Cutting belongs to one of the most typical HSR scenarios, which is designed to ensure the smoothness of rail and high speed of the train operation. The cutting has a semi-closed and long and narrow structure, and its material is concrete. The two slopes of the cutting are usually covered with vegetation. Fig. 1 shows a realistic cutting scenario on Zhengzhou-Xi'an HSR in China. We have performed field channel measurements with single-input single-output (SISO) configuration in this scenario [34]. The measured cutting had an upper width of 40 m, a bottom width of 16 m, and a depth of 5 m. The Tx antenna with the height of 9 m was placed on the top of the cutting and was 20 m away from the railway. The Rx used the train-mounted antenna with the height of 3 m. The moving velocity of the high-speed test train was 200 km/h.

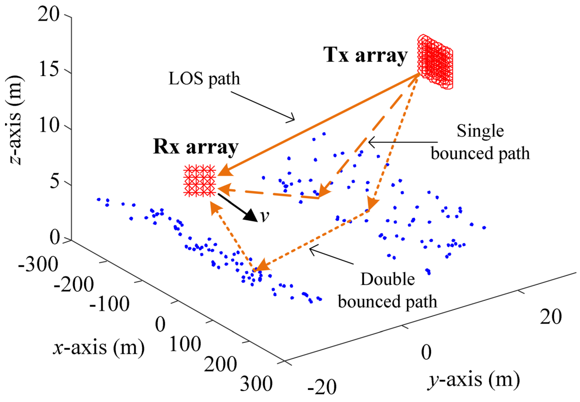

Fig. 2. The described cutting scenario using the propagation-graph.

TABLE I
SIMULATION CONFIGURATION

Parameter	Value
Scenario size	500 m×40 m×20 m
Carrier frequency	2.35 GHz
Bandwidth	50 MHz
Tx power	30 dBm
Channel sampling rate	2 kHz
Tx array size	8×8
Rx array size	4×4
Antenna spacing	0.5 wavelength
Moving speed	200/250/300 km/h

In order to generate the mMIMO channel data in the measured cutting scenario, we first use the propagation-graph to describe the propagation scenario, as shown in Fig. 2. In the three-dimensional (3-D) space (x, y, z), the Tx is fixed at the position of (0, 22 m, 14 m), while the Rx moves along the x-axis with an initial position of (-250 m, 0, 3 m). Planar arrays with 64 and 16 antennas are employed at Tx and Rx sides, respectively. The scatterers marked with blue dot are uniformly distributed on the two slopes of cutting. Note that the other scatterers around the cutting such as trees and cross-bridge are ignored here. The simulation configuration is completely in accordance with the measurement, as listed in Table I. The moving distance of Rx is 500 m and different moving speeds are considered in the simulation, including 200 km/h, 250 km/h, and 300 km/h. Then, according to the propagation-graph based channel simulation method, the CFRs with LOS, single bounced and double bounced components for the cutting scenario are derived. Moreover, the channel impulse responses (CIRs) are obtained by applying inverse discrete Fourier transform (IDFT) to the CFRs. Finally, the CIRs with 17500 snapshots for 1024 sub-channels are simulated.

B. Dataset Generation and Validation

Since the CIRs are the complex-valued data, we further transform them to the real-valued data in order to adapt the input parameters of DL. Here, four typical real-valued channel parameters are considered, involving channel magnitude (CM), K-factor (KF), RMS DS, and RMS AS. These parameters are commonly used to reflect the CSI and CSCs, which can be derived from the CIRs as follows.

1) CM: The CM is a parameter that can describe the CSI of wireless systems. Although the CM only contains the magnitude of CSI, it is already enough for most of adaptive tasks. The CM can be simply calculated as [21]

$$CM_t^{ij} = \left| \int h_{ij}\left(t,\tau\right) d\tau \right| \tag{1}$$

where $h_{ij}\left(t,\tau\right)$ denotes the CIR between i-th Tx antenna and j-th Rx antenna, t and τ indicate the time (snapshot) and delay index, respectively.

2) KF: The KF is a measure of channel fading severity, defined as the power ratio of the LOS component to the NLOS components. The KF can be useful for the system design such as link budget, adaptive Rx design and optimal loading for Tx diversity. A moment-based method is widely used to estimate the KF as follows [34]

$$KF_t^{ij} = \frac{\sqrt{1 - Var\left\{P_{ij}\left(t\right)\right\} / E^2\left\{P_{ij}\left(t\right)\right\}}}{1 - \sqrt{1 - Var\left\{P_{ij}\left(t\right)\right\} / E^2\left\{P_{ij}\left(t\right)\right\}}} \tag{2}$$

where $P_{ij}\left(t\right)$ is the power of the CIR between i-th Tx antenna and j-th Rx antenna, $E\left\{\cdot\right\}$ and $Var\left\{\cdot\right\}$ denote the mean and variance of $\left\{\cdot\right\}$, respectively.

3) RMS DS: The RMS DS is used to quantify the time dispersion of wireless channels. The RMS DS is defined as the root second central moment of the power delay profile (PDP). Here, a dynamic noise threshold is applied to the CIRs, which can enable the extraction of effective multipath components (MPCs). The calculation of RMS DS is given as [34]

$$DS_t^{ij} = \sqrt{\frac{\int PDP_{ij}\left(t,\tau\right)\tau^2 d\tau}{\int PDP_{ij}\left(t,\tau\right) d\tau} - \left(\frac{\int PDP_{ij}\left(t,\tau\right)\tau d\tau}{\int PDP_{ij}\left(t,\tau\right) d\tau}\right)^2} \tag{3}$$

where $PDP_{ij}\left(t,\tau\right)$ denotes the PDP with effective MPCs between i-th Tx antenna and j-th Rx antenna.

4) RMS AS: The severity of space dispersion is quantified by the RMS AS, which is defined as the root second central moment of the power angular spectrum (PAS). To obtain the PAS, the angular parameters such as angles of departure (AODs) and angles of arrival (AOAs) are estimated by a space-alternating generalized expectation-maximization (SAGE) algorithm. Here, we mainly focus on the RMS AS of the AOAs. The calculation of RMS AS is expressed as [36]

$$AS_t^{ij} = \sqrt{\frac{\int PAS_{ij}\left(t,\varphi\right)\varphi^2 d\varphi}{\int PAS_{ij}\left(t,\varphi\right) d\varphi} - \left(\frac{\int PAS_{ij}\left(t,\varphi\right)\varphi d\varphi}{\int PAS_{ij}\left(t,\varphi\right) d\varphi}\right)^2} \tag{4}$$

where $PAS_{ij}\left(t,\varphi\right)$ denotes the PAS with effective MPCs between i-th Tx antenna and j-th Rx antenna.

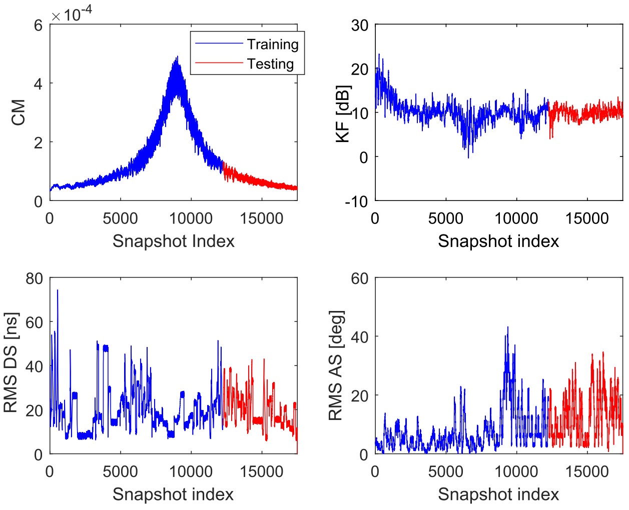

Fig. 3. Datasets of four channel parameters for the sub-channel from the first Tx antenna to the first Rx antenna.

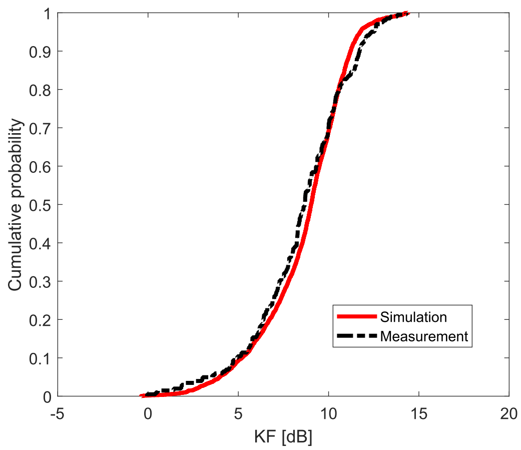

Fig. 4. CDF results of KF in the measurement and simulation.

After the parameter extraction, we generate the datasets with the four channel parameters for the 1024 sub-channels. Here, 17500 datasets in each sub-channel and 17500 × 1024 datasets in total are generated. Further, 70% and 30% of datasets, i.e., 12250 × 1024 and 5250 × 1024 datasets are divided into the training and testing datasets, respectively. Fig. 3 illustrates the generated datasets of four channel parameters for the sub-channel from the first Tx antenna to the first Rx antenna. It can be seen that the last 30% of the total datasets are selected as the training datasets.

Moreover, to validate the generated datasets, actual measurement results should be considered for comparison. However, we can only compare the results in SISO case because of the limited measurement data. Fig. 4 shows the cumulative distribution function (CDF) results of KF for the measured channel and the simulated sub-channel between the first Tx antenna and the first Rx antenna. Note that the results with the same segment (e.g., 0-250 m) in the measured and simulated scenario are used for comparison. We use the Jensen-Shannon divergent (JSD) to measure the distribution distance of CDFs for measured and simulated KF results. The JSD is in the range of 0 and 1, and the closer the value of JSD is to 0, the better the agreement of two distributions is. The calculated JSD value is 0.08, which confirms that there is a good match between the measured and simulated results. In addition, we have also validated the propagation-graph based simulation results in the cutting scenario, in terms of delay dispersion characteristics and frequency dispersion characteristics. Detailed validation results can be found in our previous work [34]. With the SISO validation, it can be concluded that the environment-related parameters in the propagation-graph based channel simulation, involving the number and distribution of scatterers, are consistent with the actual propagation environment information. In this case, it is reasonable to generate the effective mMIMO datasets when we extend the channel simulation to the mMIMO case by using the configuration of multiple antennas.

C. Prediction Problem Formulation

We aim to predict the time-varying mMIMO channel in the HSR cutting scenario, based on the historical data instead of the input parameters such as the coordinates of Tx and Rx antennas, Tx-Rx distance and carrier frequency. The advantage of this prediction is that the spatial and temporal information hidden in the datasets can be effectively exploited. However, it requires the historical data as the inputs of the networks. With regard to a communication system, the past channel estimation results can be utilized as the historical data.

Here, the matrix $\boldsymbol{X}_t$ is used to represent a certain channel parameter (CM, KF, RMS DS, or RMS AS) for $M_1 \times M_2$ sub-channels at snapshot t, where M_1 and M_2 are the numbers of Tx and Rx antennas. When a planar array with same antenna spacing is employed, the channel parameter for M_1 sub-channels from different Tx antennas to the j-th Rx antenna can be written as

$$\boldsymbol{X}_t^j = \begin{bmatrix} x_t^{11,j} & x_t^{12,j} & \cdots & x_t^{1n,j} \\ x_t^{21,j} & x_t^{22,j} & \cdots & x_t^{2n,j} \\ \vdots & \vdots & \ddots & \vdots \\ x_t^{m1,j} & x_t^{m2,j} & \cdots & x_t^{mn,j} \end{bmatrix} \tag{5}$$

where m and n are row and column indices of Tx array, respectively, and $m \times n = M_1$. As for the prediction of $\boldsymbol{X}_t^j$, the prediction problem can be expressed as

$$\hat{\boldsymbol{X}}_{t+1}^j = F\left(\boldsymbol{X}_t^j, \boldsymbol{X}_{t-1}^j, \cdots, \boldsymbol{X}_{t-T+1}^j\right) \tag{6}$$

where $\hat{\boldsymbol{X}}_{t+1}^j$ denotes the predicted result at next snapshot, $F(\cdot)$ indicates a predicting function or predictor, and T is the time window length of historical data used for prediction. It is worth noting that the inputs of the predictor are the historical data in the spatial and temporal dimensions. Note that the spatial dimension means the array dimension. Thus, the temporal information and spatial information at Tx side hidden in $\boldsymbol{X}_t^j$ can be utilized by the predictor.

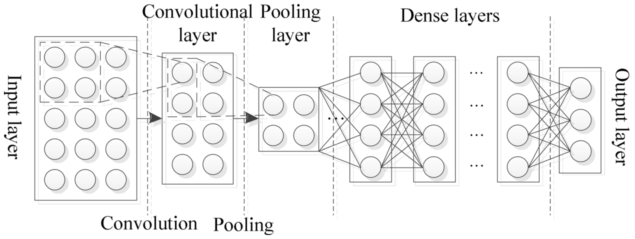

Fig. 5. A basic architecture of CNN.

Furthermore, as for the prediction of $\boldsymbol{X}_t$, the spatial information at RX side can be also exploited. Thus, the prediction problem can be updated as

$$\hat{\boldsymbol{X}}_{t+1} = F'\left(\hat{\boldsymbol{X}}_{t+1}^1, \hat{\boldsymbol{X}}_{t+1}^2, \cdots, \hat{\boldsymbol{X}}_{t+1}^j, \cdots, \hat{\boldsymbol{X}}_{t+1}^{M_2}\right) \tag{7}$$

where $F'(\cdot)$ represents another predicting function that employs the outputs of predictor $F(\cdot)$ as the inputs. Note that the outputs of predictor $F(\cdot)$ mean the predicted results for M_2 Rx antennas.

Finally, we combine the two predictors to create a new predictor $G(\cdot)$, which can incorporate the spatial information at both Tx and Rx sides and temporal information. In this paper, both single-step ahead prediction and multi-step ahead prediction problems are considered, which can be respectively formulated as

$$\hat{\boldsymbol{X}}_{t+1} = G\left(\boldsymbol{X}_t, \boldsymbol{X}_{t-1}, \cdots, \boldsymbol{X}_{t-T+1}\right) \tag{8}$$

and

$$\left(\hat{\boldsymbol{X}}_{t+1}, \hat{\boldsymbol{X}}_{t+2}, \cdots, \hat{\boldsymbol{X}}_{t+k}\right) = G\left(\boldsymbol{X}_t, \boldsymbol{X}_{t-1}, \cdots, \boldsymbol{X}_{t-T+1}\right) \tag{9}$$

where k denotes the prediction length for the multi-step predictor.

III. Deep-Learning-Based Spatial–Temporal Prediction Model for HSR mMIMO Channels

A. CNN for Spatial Channel Prediction

Spatial correlation could exist between different antennas or sub-channels. The spatial correlation is related to the antenna spacing and the propagation environment. If the antenna spacing is smaller and the propagation environment has less scatterers, the spatial correlation will be higher. In the HSR scenario, there could exist the strong spatial correlation due to the feature of LOS dominance. This means that the spatial correlation information is hidden in the HSR mMIMO channel datasets. If this relationship can be effectively captured by the channel predictor, its performance will be significantly improved.

To utilize the spatial correlation information of channel datasets, we resort to the CNN. The CNN improves the traditional deep neural network (DNN) by inserting convolutional and pooling layers before the dense layers. A basic architecture of CNN is illustrated in Fig. 5 [37]. Each output in the convolutional layer is obtained by dot product between a certain filter matrix and an input matrix comprised of several neurons in the upper layer. This filter matrix is known as the convolution kernel and the output of the convolutional layer is called as feature map. According to the convolution operation, the spatial relationship hidden in the datasets can be learned. The pooling layer is to reduce the dimensionality of the feature map but retains the most important information by averaging or searching maximum in a group of neurons in the convolutional layer. However, the pooling layer is usually omitted for the channel prediction in order to maintain the forecasting accuracy [22].

Assuming multiple convolutional layers are used, the mathematical relationship for the input feature map $\boldsymbol{H}^l$ and output feature map $\boldsymbol{H}^{l+1}$ in the $l+1$-th convolutional layer can be expressed as

$$\begin{aligned}\boldsymbol{H}^{l+1}(i,j) &= \left[\boldsymbol{H}^l * \boldsymbol{w}^{l+1}\right](i,j)\\ &= \sum_{x=1}^{K}\sum_{y=1}^{K}\left[\boldsymbol{H}^l(si+x, sj+y)\,\boldsymbol{w}^{l+1}\right] \quad (10)\end{aligned}$$

where $\boldsymbol{H}^{l+1}(i,j)$ denotes the element of feature map, $*$ means the convolution operation, $\boldsymbol{w}^{l+1}$ indicates the convolution kernel of the $l+1$-th convolutional layer, $K \times K$ represents the convolution kernel size, and s is convolution step. In the above equation, $(i,j) \in \{0, 1, \cdots, L_{l+1}\}$, where $L_{l+1} \times L_{l+1}$ is the size of the output feature map, $L_{l+1} = (L_l + 2P - K)/s + 1$, and P is the padding number.

It is noteworthy that the convolution kernel size, the convolution step, and the padding number are the most important parameters of CNN. The convolution kernel size determines the range of the receptive field, which directly affects the performance of feature extraction. As for the spatial channel prediction, the selection of convolution kernel size is based on the similarity between different sub-channels, which will be analyzed in the next section. The convolution step influences the resolution of feature extraction. We usually set the convolution step as the smallest value, i.e., $s = 1$. By setting a suitable padding number, it can compensate the dimensional change of input and output feature maps after the convolution.

B. LSTM for Temporal Channel Prediction

In addition to the spatial correlation, the strong temporal correlation also exists in the HSR mMIMO channel, due to the high mobility of Rx. It is necessary to exploit this temporal correlation information hidden in the HSR mMIMO channel datasets, which is beneficial to enhance the forecasting accuracy for the predictor.

Different from feedforward neural networks such as DNN and CNN, the RNN is an ANN with self-connections, which adds a feedback path to the feedforward neural network to provide information from the previous input. With the recurrent structure, the output of the network depends not only on the current input, but also on the hidden state learned from the previous input, which enables the RNN to learn the temporal correlation information of the datasets. A basic structure of RNN is shown in Fig. 6 [37]. It is worth noting that the time window length of input dataset T should be determined depending on its autocorrelation, which will be described in the next section. The hidden state $\boldsymbol{h}_t$ and the output $\boldsymbol{y}_t$ are calculated as

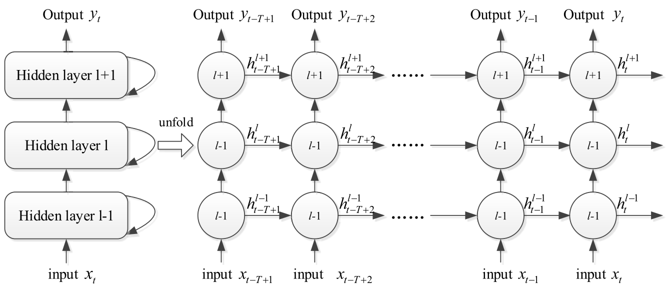

Fig. 6. A basic structure of RNN.

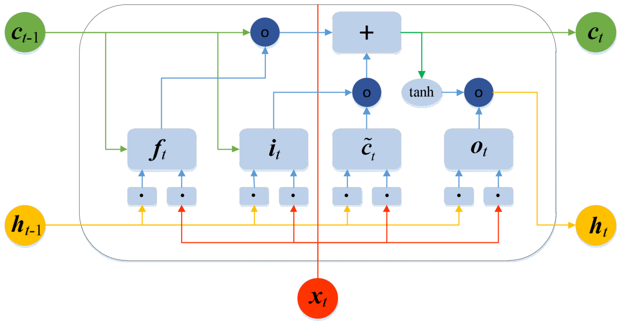

Fig. 7. The inner structure of a LSTM cell.

$$\boldsymbol{h}_t = f(\boldsymbol{U}\cdot\boldsymbol{x}_t + \boldsymbol{W}\cdot\boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \quad (11)$$

$$\boldsymbol{y}_t = g(\boldsymbol{V}\cdot\boldsymbol{h}_t + \boldsymbol{b}_y) \quad (12)$$

where $\boldsymbol{x}_t$ denotes the input vector to this recurrent layer, $\boldsymbol{h}_t$ and $\boldsymbol{h}_{t-1}$ indicate the hidden state from the current step and the previous time step respectively, $\boldsymbol{U}$, $\boldsymbol{W}$ and $\boldsymbol{V}$ represent the connection weight matrices from the input layer to the hidden layer, the hidden layer to itself and the hidden layer to the next hidden layer or the output layer respectively, $\boldsymbol{b}_h$ and $\boldsymbol{b}_y$ are the bias vectors, $f(\cdot)$ and $g(\cdot)$ are the activation functions.

However, the traditional RNN cannot discover relations contained in long-term sequences due to the gradient exploding and vanishing problems in training process. To alleviate this drawback, the LSTM network, as a special designed RNN network, has been introduced to keep a long-term memory inside each RNN unit by adding three gates to the cell unit, including forgetting gate, input gate and output gate [38]. Fig. 7 illustrates the inner structure of a LSTM cell.

The forgetting gate $\boldsymbol{f}_t$ is used to determine what information should be discarded in the cell state at the previous time step while the input gate $\boldsymbol{i}_t$ decides the input at the current moment, which can be described as

$$\boldsymbol{f}_t = \sigma_f(\boldsymbol{W}_{xf}\cdot\boldsymbol{x}_t + \boldsymbol{W}_{hf}\cdot\boldsymbol{h}_{t-1} + \boldsymbol{W}_{cf}\cdot\boldsymbol{c}_{t-1} + \boldsymbol{b}_f) \quad (13)$$

$$\boldsymbol{i}_t = \sigma_i(\boldsymbol{W}_{xi}\cdot\boldsymbol{x}_t + \boldsymbol{W}_{hi}\cdot\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ci}\cdot\boldsymbol{c}_{t-1} + \boldsymbol{b}_i) \quad (14)$$

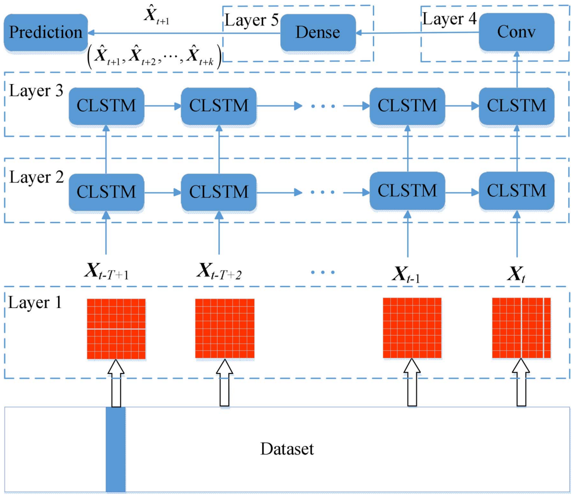

Fig. 8. Structure of the proposed Conv-CLSTM model.

where σ represents the sigmoid function, $\boldsymbol{x}_t$ denotes the input data at time t, $\boldsymbol{c}_{t-1}$ indicates the cell state before time t, $\boldsymbol{h}_{t-1}$ is the hidden state before time t.

The update algorithm of cell state $\boldsymbol{c}_t$ is related to the $\boldsymbol{c}_{t-1}$ and $\tilde{\boldsymbol{c}}_t$, which can be calculated as

$$\tilde{\boldsymbol{c}}_t = \tanh\left(\boldsymbol{W}_{xc} \cdot \boldsymbol{x}_t + \boldsymbol{W}_{hc} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \quad (15)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t \quad (16)$$

where $\tilde{\boldsymbol{c}}_t$ is the candidate value for updating cell state, and the operator $\circ$ denotes the Hadamard product (element-wise multiplication).

The output gate $\boldsymbol{o}_t$ decides what information in the current cell state $\boldsymbol{c}_t$ should be added to the hidden state. The process can be expressed as

$$\boldsymbol{o}_t = \sigma_o\left(\boldsymbol{W}_{xo} \cdot \boldsymbol{x}_t + \boldsymbol{W}_{ho} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{W}_{co} \cdot \boldsymbol{c}_t + \boldsymbol{b}_o\right) \quad (17)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh\left(\boldsymbol{c}_t\right) \quad (18)$$

where $\boldsymbol{h}_t$ is the updated hidden state at time t which can be input to next layer of the LSTM. Note that $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrices and bias vectors, respectively.

C. Novel Conv-CLSTM Model for Spatial-Temporal Channel Prediction

Taking the advantages of CNN and LSTM, we propose a novel Conv-CLSTM model to achieve the spatial-temporal prediction for the HSR mMIMO channel. Fig. 8 illustrates the structure of the proposed model. The Conv-CLSTM model is designed as five layers, including an input layer, two CLSTM layers, a convolutional layer, and an output layer. The input layer is used to adjust the structure of the spatial-temporal channel data according to the Tx array structure and the time window length, and normalize the input data in order to meet the input requirement of the network. The CLSTM is employed in the layers 2 and 3, which is an extension of LSTM with convolutional structures, and possesses the learning ability of both spatial and temporal correlation information [24]. To further learn the spatial correlation information of RX array, one more convolutional layer is inserted in the model. The input structure of this convolutional layer should be adjusted according to the RX array structure. Finally, the output layer adopts a dense layer to obtain the predicted results, involving the single-step ahead and multi-step ahead forecasting results. Due to the combination of CLSTM and CNN, the proposed Conv-CLSTM model can fully exploit the spatial information at both Tx and Rx sides and temporal information concealed in the HSR mMIMO channel datasets.

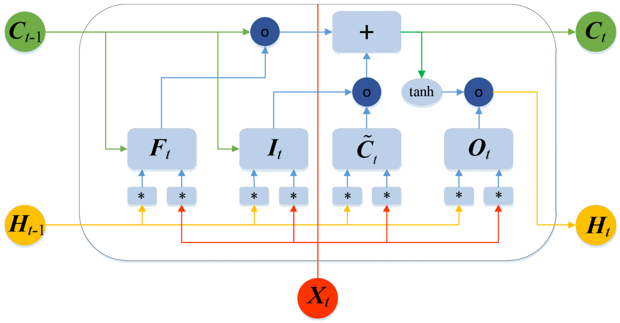

Fig. 9. Structure of the CLSTM cell.

The CLSTM cell can be implemented by replacing the fully connected calculation in the LSTM cell with the convolution operation [24]. The structure of the CLSTM cell is similar with the LSTM cell, as shown in Fig. 9. The mathematical expression of calculation process can be simply described as

$$\boldsymbol{F}_t = \sigma_f\left(\boldsymbol{W}_{xf} * \boldsymbol{X}_t + \boldsymbol{W}_{hf} * \boldsymbol{H}_{t-1} + \boldsymbol{W}_{cf} \cdot \boldsymbol{C}_{t-1} + \boldsymbol{b}_f\right) \quad (19)$$

$$\boldsymbol{I}_t = \sigma_i\left(\boldsymbol{W}_{xi} * \boldsymbol{X}_t + \boldsymbol{W}_{hi} * \boldsymbol{H}_{t-1} + \boldsymbol{W}_{ci} \cdot \boldsymbol{C}_{t-1} + \boldsymbol{b}_i\right) \quad (20)$$

$$\tilde{\boldsymbol{C}}_t = \tanh\left(\boldsymbol{W}_{xc} * \boldsymbol{X}_t + \boldsymbol{W}_{hc} * \boldsymbol{H}_{t-1} + \boldsymbol{b}_c\right) \quad (21)$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t \circ \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \circ \tilde{\boldsymbol{C}}_t \quad (22)$$

$$\boldsymbol{O}_t = \sigma_o\left(\boldsymbol{W}_{xo} * \boldsymbol{X}_t + \boldsymbol{W}_{ho} * \boldsymbol{H}_{t-1} + \boldsymbol{W}_{co} \cdot \boldsymbol{C}_t + \boldsymbol{b}_o\right) \quad (23)$$

$$\boldsymbol{H}_t = \boldsymbol{O}_t \circ \tanh\left(\boldsymbol{C}_t\right) \quad (24)$$

From the above equations, it is known that the future state of a certain CLSTM cell is determined through the past state and the current input of neighbors. Thus, the CLSTM can not only learn the time correlation of data through historical data, but also considers the impact of neighboring data in space on the current state. It should be mentioned that to ensure that the state has the same size as the inputs, we perform zero-padding before applying the convolution operation.

We adopt the back-propagation through time (BPTT) algorithm [39] to update the weights and biases contained in the Conv-CLSTM model. If results obtained by the output layer are not as expected, the errors will propagate back to the hidden layers and input layer, then the weights and biases will be modified according to the gradients calculated by the errors. The outputs of the network will gradually approach the expected outputs until they are consistent with the expected outputs or the loss entropy is less than a certain threshold.

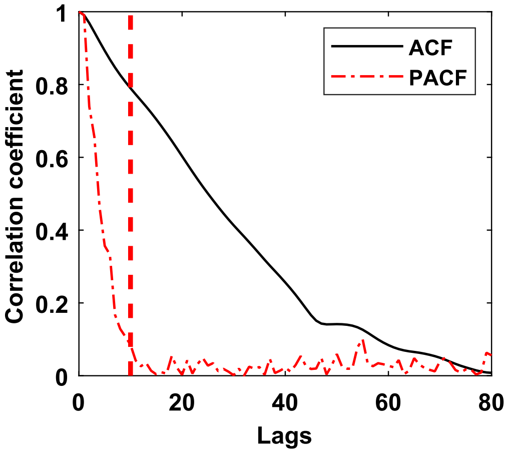

Fig. 10. ACF and PACF results of KF.

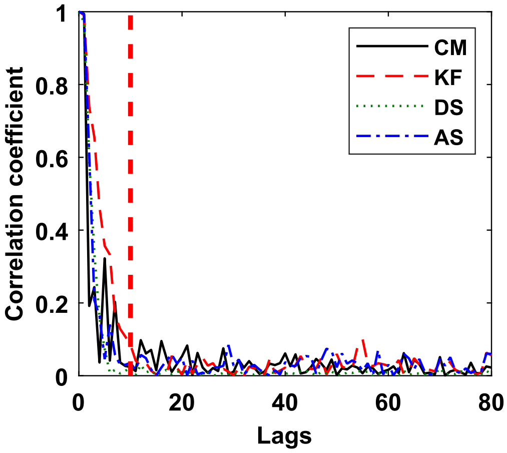

Fig. 11. PACF results for the four channel parameters.

The hyper-parameters of the Conv-CLSTM model such as the time window length, the convolution kernel size, the number of hidden neurons and the number of hidden layers should be carefully configured, which will directly affect the prediction performance. The determination of these hyper-parameters will be described in the next section.

IV. Hyper-Parameters Determination of the Proposed Model

A. Autocorrelation Analysis

Autocorrelation analysis is usually used to determine the order of AR models for time series. Similarly, the autocorrelation analysis can be employed to decide the time window length for the Conv-CLSTM model. Autocorrelation function (ACF) and its partial derivative (PACF) are two measures used for the autocorrelation analysis. The ACF can be regarded as a complete ACF, which describes the correlation of different lags in a time series and involves direct and indirect correlation information. The PACF gives the partial correlation of the time series with its own lagged values, which removes the effect of any indirect correlations due to the terms at shorter lags. In other words, the PACF measures the linear correlation of a series $\{x_t\}$ and a lagged version of itself $\{x_{t-k}\}$ with the linear dependence of $\{x_{t-1}, x_{t-2}, \ldots, x_{t-k+1}\}$ removed. The advantage of using the PACF is that it can solve the multicollinearity problem when building the prediction models [40].

As an example, the ACF and PACF results of KF are shown in Fig. 10. It can be seen that the correlation coefficient of PACF rapidly reduces to a small value with the increase of time lag, whereas the ACF curve is slowly decreasing. This confirms that the PACF is more suitable to capture the effective correlation information than the ACF. The PACF results for the four channel parameters are obtained, as shown in Fig. 11. It can be found that when the time lag is over 10, the PACF values of all the channel parameters decrease to below 0.1, which means the correlation is not obvious. Thus, the time window length is recommended as 10 in the Conv-CLSTM model.

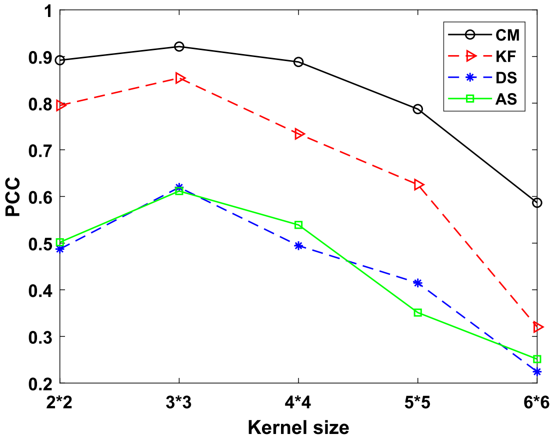

Fig. 12. PCC results for the four channel parameters.

B. Similarity Analysis

The determination of the convolution kernel size is based on the similarity between different antennas. To analyze this spatial similarity, we resort to Pearson correlation coefficient (PCC). This metric measures how highly correlated are two variables and is measured from -1 to $+1$. A PCC of -1 or $+1$ indicates that the two variables are perfectly negatively or positively correlated while a score of 0 means that the two variables are not correlated. Note that the PCC reflects the level of linear correlation. Considering the convolution kernel with a $K \times K$ size, the PCCs between each two antennas in the $K \times K$ range should be calculated and averaged. Then, the averaged PCC is used to evaluate the similarity between the $K \times K$ antennas. In addition, we should slide this window with $K \times K$ antennas to traverse the entire array and calculate the averaged PCC again. Our aim is to determine the convolution kernel size based on the final PCC result.

Fig. 12 illustrates the PCC results of different kernel sizes for the four channel parameters. Note that only the result of similarity analysis at Tx side is plotted. It is observed that as for the four channel parameters, the PCC has the largest value when the kernel size is 3×3, and it gradually decreases with the kernel size increased. This means that the similarity between the antennas in the 3×3 range is the highest. We also obtain the same result at Rx side. Therefore,

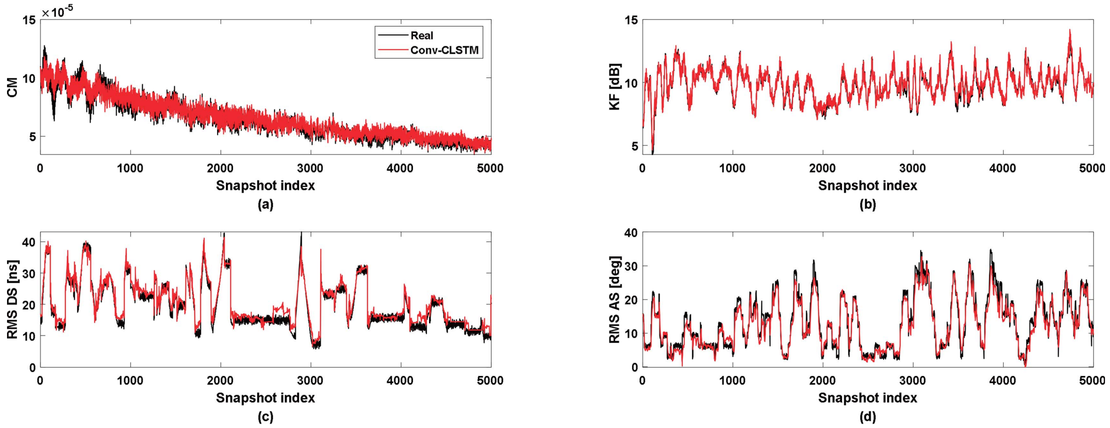

Fig. 13. Comparison of real and predicted results for the four channel parameters.

the convolution kernel size of the Conv-CLSTM model is suggested to be 3 × 3.

C. Cross-Validation

The k-fold cross-validation, as an advanced validation strategy, is adopted to verify the selection of the time window length and the convolution kernel size from the theoretical analysis, and to further decide the other optimal hyper-parameters such as the number of hidden neurons and the number of hidden layers. The setting of $k = 5$ is employed and the testing datasets are used as the validation datasets. Note that we choose the root mean square error (RMSE) as the metric for evaluating the prediction accuracy.

In the training process, we use MSELoss as the loss function, and initialize the weights with batch normalization and the biases are initially set to zero [41]. As for the optimizer, we choose RMSprop to update the weights and biases. The initial learning rate and decay rate are set to 0.001 and 0.9, respectively. The learning rate is adjusted every 50 epochs and the maximum epoch is 500. In addition, the mini-batch is set to 128.

Focusing on the KF, the cross-validation results of the time window length, convolution kernel size, the number of hidden neurons and the number of hidden layers are listed in Table II. It can be seen that the RMSE reaches the lowest value when the time window length and convolution kernel size are set to 10 and 3 × 3, respectively. This confirms the results of the theoretical analysis. In addition, 128 hidden neurons and 2 hidden layers are found to be the best selection. If more hidden neurons and hidden layers are added, it could cause the over-fitting problem, leading to the deterioration of the prediction performance. Table III summarizes the results of the optimal hyper-parameters for the four channel parameters. It is observed that except the number of hidden neurons, the optimal hyper-parameters are the same for the four channel parameters.

TABLE II
CROSS-VALIDATION RESULTS FOR KF

Hyper-parameters	Value	RMSE
Window length	5	0.3951
	10	0.3131
	15	0.3902
	20	0.3905
Convolution kernel size	2×2	0.3609
	3×3	0.3131
	4×4	0.5384
	5×5	1.1636
	6×6	1.3651
The number of hidden neurons	32	0.6074
	64	0.4908
	128	0.3131
	256	0.3372
The number of hidden layers	1	0.3448
	2	0.3131
	3	0.4594
	4	0.5375

TABLE III
RESULTS OF THE OPTIMAL HYPER-PARAMETERS FOR THE FOUR CHANNEL PARAMETERS

Hyper-parameters	CM	KF	DS	AS
Window length	10	10	10	10
Convolution kernel size	3×3	3×3	3×3	3×3
The number of hidden neurons	32	128	256	256
The number of hidden layers	2	2	2	2

V. PERFORMANCE EVALUATION OF THE PROPOSED MODEL

A. Prediction Accuracy

Based on the optimal hyper-parameters, we evaluate the prediction accuracy of the proposed Conv-CLSTM model in terms of RMSE. Fig. 13 compares the real and predicted results for the four channel parameters. It can be seen that there appears a high degree of fitting between the real and predicted curves. This implies that the proposed model has a good forecasting effect. To further assess the performance, we compare the results of the proposed model and the traditional ML and DL models, such as recursive least square (RLS), SVM, DNN, CNN, RNN, and LSTM. The RMSE results of different models for the four channel parameters are listed in Table IV. Note that each model uses the respective optimal hyper-parameters. It is observed that the Conv-CLSTM model outperforms the other models, which has the lowest RMSE value. Besides, the performance of RLS, SVM, and DNN models is much

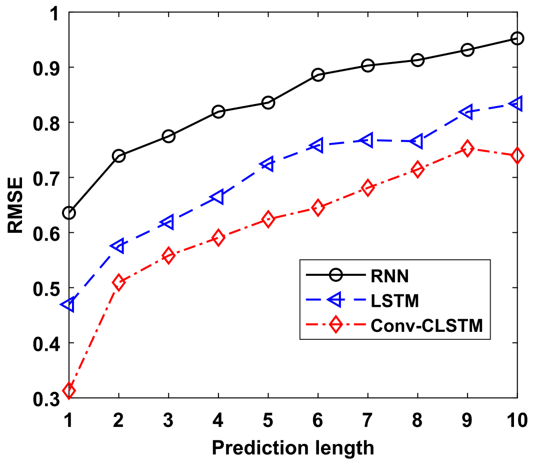

Fig. 14. Results of multi-step ahead prediction for different models.

TABLE IV
RMSE RESULTS OF DIFFERENT MODELS FOR THE FOUR CHANNEL PARAMETERS

Model	CM	KF	DS	AS
RLS	3.3202	3.2966	6.5056	5.7565
SVM	2.7269	3.4151	5.0492	4.8724
DNN	3.3249	5.6657	5.3771	3.0787
CNN	0.2251	0.5024	2.2135	2.2863
RNN	0.0961	0.6357	0.9967	0.8832
LSTM	0.0845	0.4696	0.8197	0.7528
Conv-LSTM	0.0421	0.3131	0.6812	0.6925

worse than the spatial or temporal models such as CNN, RNN, and LSTM. This confirms that exploiting the spatial and temporal correlation information hidden in the datasets is able to improve the forecasting performance.

In addition to the single-step ahead prediction, the performance of multi-step ahead prediction for the RNN, LSTM, and Conv-CLSTM models is also evaluated, as shown in Fig. 14. Note that the RMSE results of KF are plotted as an example, and one prediction length corresponds to 0.5 ms as the channel sampling rate is 2 kHz. It is found that the RMSE curves of the three models gradually rise with the prediction length increased. This is because the temporal correlation weakens due to the longer prediction length, which causes the deterioration of the prediction accuracy. It can be also seen that the proposed model still has better performance than the RNN and LSTM models in terms of multi-step prediction.

The impact of the moving speed on the forecasting performance for the temporal models is also evaluated. Table V lists the RMSE results with three moving speeds for the RNN, LSTM, and Conv-CLSTM models. Note that the single-step prediction is considered and the RMSE results of KF are listed as an example. It can be found that the RMSE gradually reduces with the moving speed increased. This is because the higher velocity leads to the shorter temporal correlation time, which deteriorates the prediction performance of the models. We can also confirm that the proposed Conv-CLSTM model still outperforms the RNN and LSTM models in the case of different velocities.

TABLE V
RMSE RESULTS WITH DIFFERENT MOVING SPEEDS FOR DIFFERENT MODELS

Speed (km/h)	RNN	LSTM	Conv-CLSTM
200	0.6357	0.4696	0.3131
250	0.7248	0.5584	0.4326
300	0.8193	0.6492	0.5273

B. Computational Complexity

The computational complexity of a DL model can be evaluated from space and time aspects. Here, the number of parameters and the number of floating point operations (FLOPs) are chosen as the measures of the space and time complexity, respectively.

The number of parameters, also called the capacity of a model, is usually used to evaluate the memory usage of a computer. The computational complexity of the model per parameter and time step is defined as $\mathcal{O}\left(1\right)$. Taking the Conv-CLSTM model as an example, its complexity of per time step in the training process can be measured by $\mathcal{O}\left(N_{Conv-LSTM}\right)$. The number of trainable parameters including weights and biases for the CNN, RNN, LSTM and Conv-CLSTM models can be respectively calculated as

$$\begin{aligned}N_{CNN} = {}& C_i \times K^2 \times C^{(1)} + C^{(1)}\\ &+ C^{(1)} \times K^2 \times C^{(2)} + C^{(2)}\\ &+ C^{(2)} \times K^2 \times C_o + C_o, \end{aligned}\tag{25}$$

$$\begin{aligned}N_{RNN} = {}& (n_i + n_h^{(1)}) \times n_h^{(1)} + n_h^{(1)}\\ &+ (n_h^{(1)} + n_h^{(2)}) \times n_h^{(2)} + n_h^{(2)}\\ &+ n_h^{(2)} \times n_o + n_o, \end{aligned}\tag{26}$$

$$\begin{aligned}N_{LSTM} = {}& 4[(n_i + n_h^{(1)}) \times n_h^{(1)} + n_h^{(1)}]\\ &+ 4\left[(n_h^{(1)} + n_h^{(2)}) \times n_h^{(2)} + n_h^{(2)}\right]\\ &+ n_h^{(2)} \times n_o + n_o, \end{aligned}\tag{27}$$

$$\begin{aligned}N_{Conv-CLSTM} = {}& 4[(n_i + n_h^{(1)}) \times k^2 + n_h^1]\\ &+ 4\left[(n_h^{(1)} + n_h^{(2)}) \times n_h^{(2)} + n_h^{(2)}\right]\\ &+ n_h^{(2)} \times k^2 \times n_h^{(3)} + n_h^{(3)}\\ &+, n_h^{(3)} \times k^2 \times n_o + n_o \end{aligned}\tag{28}$$

where $K \times K$ is the kernel size, C_i and C_o denote the number of input channels at input layer and the number of output channels at output layer, respectively, $C^{(l)}$ indicates the number of convolutional kernels at l-th hidden layer, $H^{(l)} \times W^{(l)}$ represents the size of feature map at l-th hidden layer, n_i, n_o and $n_h^{(l)}$ denote the number of input units, output units and hidden states at l-th hidden layer, respectively, $l = 1, 2, 3$.

The number of FLOPs is obtained by counting how many computations the model does, which determines the computational time of the model. It usually counts addition, subtraction, multiplication, division, exponentiation and square root as a single FLOP. Note that the operations contained in

TABLE VI
EVALUATION RESULTS OF COMPUTATIONAL COMPLEXITY FOR DIFFERENT MODELS

Model	Space complexity		Time complexity	
	Number of parameters	Memory usage (MB)	Number of FLOPs	Computational time (us)
CNN	65,856	0.251	168,897	0.001
RNN	1,053,696	4.020	2,098,113	0.014
LSTM	3,818,496	14.566	7,603,137	0.049
Conv-CLSTM	1,179,216	4.498	30,244,680	0.197

activation functions are not considered since they only take up a small fraction of the overall computations. Thus, the number of FLOPs for the CNN, RNN, LSTM and Conv-CLSTM models can be respectively computed as

$$\begin{aligned} F_{CNN} &= 2C_i \times K^2 \times H^{(1)} \times W^{(1)} \times C^{(1)} \\ &\quad + 2C^{(1)} \times K^2 \times H^{(2)} \times W^{(2)} \times C^{(2)} \\ &\quad + 2C^{(2)} \times K^2 \times C_o, \end{aligned} \tag{29}$$

$$\begin{aligned} F_{RNN} &= 2(n_i + n_h^{(1)}) \times n_h^{(1)} \\ &\quad + 2(n_h^{(1)} + n_h^{(2)}) \times n_h^{(2)} \\ &\quad + 2n_h^{(2)} \times n_o, \end{aligned} \tag{30}$$

$$\begin{aligned} F_{LSTM} &= 8(n_i + n_h^{(1)}) \times n_h^{(1)} \\ &\quad + 8(n_h^{(1)} + n_h^{(2)}) \times n_h^{(2)} \\ &\quad + 2n_h^{(2)} \times n_o, \end{aligned} \tag{31}$$

$$\begin{aligned} F_{Conv-CLSTM} &= 8K^2 \times H^{(1)} \times W^{(1)} \times (n_i + n_h^{(1)}) \\ &\quad + 8K^2 \times H^{(2)} \times W^{(2)} \times (n_h^{(1)} + n_h^{(2)}) \\ &\quad + 2K^2 \times H^{(3)} \times W^{(3)} \\ &\quad + 2K^2 \times n_o \end{aligned} \tag{32}$$

where $H^{(l)} \times W^{(l)}$ denotes the size of feature map at l-th hidden layer.

Table VI lists the evaluation results of computational complexity for the CNN, RNN, LSTM and Conv-CLSTM models. As for the space complexity, it can be seen that the Conv-CLSTM model has smaller number of parameters than the LSTM model. This is because the parameter sharing achieved by the introduction of the convolution operation in the proposed model reduces the number of parameters. Due to the presence of three gates that introduces numerous weights and biases, the LSTM and Conv-CLSTM models have more parameters than the CNN model. It can be also found that the Conv-CLSTM model has a similar space complexity with the RNN model. The memory usage per time step for training the Conv-CLSTM is 4.498 MB in case of a 32-bit system. With regard to the time complexity, the proposed model has the largest number of FLOPs among the DL models since the FLOPs could not be reduced by the parameter sharing. It is estimated that the computational time of the Conv-CLSTM model is 0.197 us when the Intel i5-6300U CPU with 153.6 Giga floating point operations per second (GFLOPS) is used. In this case, the proposed model can be suitable for real-time applications in the smart HSR communication network.

VI. CONCLUSION

In this paper, DL based spatial-temporal prediction of CSI and CSCs for the future smart HSR communication network has been investigated. The datasets of CM, KF, RMS DS and RMS AS for the mMIMO channel in the HSR cutting scenario has been generated depending on the propagation-graph based channel simulation and validated according to the channel measurement results. Then, the formulation of single-step ahead and multi-step ahead prediction problems for the HSR mMIMO channel has been presented. With the combination of CNN and CLSTM, the novel Conv-CLSTM model has been proposed for spatial-temporal channel prediction, which can fully utilize the temporal and spatial correlation information hidden in the HSR mMIMO channel datasets. In addition, autocorrelation and similarity analysis and cross-validation have been used to determine the optimal hyper-parameters such as the time window length, the convolution kernel size, the number of hidden neurons and the number of hidden layers. Finally, the performance of the proposed model has been assessed by the prediction accuracy and the space and time computational complexity. It has been shown that compared with the typical ML and DL models, the Conv-CLSTM model has the highest prediction accuracy but the acceptable computational complexity.

REFERENCES

[1] S. Lin, Y. Jia, and S. Xia, "Research and analysis on the top design of smart railway," *J. Phys., Conf. Ser.*, vol. 1187, no. 5, pp. 1–7, Apr. 2019.

[2] F. Hasegawa *et al.*, "High-speed train communications standardization in 3 GPP 5G NR," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 44–52, Mar. 2018.

[3] B. Ai, A. F. Molisch, M. Rupp, and Z.-D. Zhong, "5G key technologies for smart railways," *Proc. IEEE*, vol. 108, no. 6, pp. 856–893, Jun. 2020.

[4] R. Chen, W.-X. Long, G. Mao, and C. Li, "Development trends of mobile communication systems for railways," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3131–3141, Jul. 2018.

[5] R. Li *et al.*, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[6] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6G AI-native air interface," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 76–81, May 2021.

[7] "6G vision and candidate technologies," IMT-2030 (6G) Promotion Group, Beijing, China, White Paper, 2021.

[8] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proc. IEEE*, vol. 95, no. 12, pp. 2299–2313, Dec. 2007.

[9] J.-Y. Wu and W.-M. Lee, "Optimal linear channel prediction for LTE-A uplink under channel estimation errors," *IEEE Trans. Veh. Technol.*, vol. 62, no. 8, pp. 4135–4142, Oct. 2013.

[10] W. Jiang and H. D. Schotten, "A comparison of wireless channel predictors: Artificial intelligence versus Kalman filter," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[11] R. O. Adeogun, P. D. Teal, and P. A. Dmochowski, "Parametric channel prediction for narrowband mobile MIMO systems using spatio-temporal correlation analysis," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Las Vegas, NV, USA, Sep. 2013, pp. 1–5.

[12] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118112–118124, 2019.

[13] T. Ding and A. Hirose, "Fading channel prediction based on combination of complex-valued neural networks and chirp Z-transform," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1686–1695, Sep. 2014.

[14] Y. Chen, Z. Dou, Y. Lin, and Y. Li, "Prediction of V2V channel quality under double-Rayleigh fading channels," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, Antwerp, Belgium, May 2020, pp. 1–6.

[15] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2735–2747, Jul. 2010.

[16] D. Wu, G. Zhu, and B. Ai, "Application of artificial neural networks for path loss prediction in railway environments," in *Proc. 5th Int. ICST Conf. Commun. Netw. China*, Beijing, China, 2010, pp. 1–5.

[17] Y. Zhang, J. Wen, G. Yang, Z. He, and X. Luo, "Air-to-air path loss prediction based on machine learning methods in urban environments," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Jun. 2018.

[18] J. Huang, C. X. Wang, L. Bai, J. Sun, and Y. Yang, "A big data enabled channel model for 5G wireless communication systems," *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 211–222, Jun. 2020.

[19] X. Zhao *et al.*, "Playback of 5G and beyond measured MIMO channels by an ANN-based modeling and simulation framework," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 9, pp. 1945–1954, Sep. 2020.

[20] W. Jiang and H. D. Schotten, "Recurrent neural network-based frequency-domain channel prediction for wideband communications," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Kuala Lumpur, Malaysia, Apr. 2019, pp. 1–6.

[21] W. Jiang and H. D. Schotten, "Deep learning for fading channel prediction," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 320–332, 2020.

[22] L. Bai *et al.*, "Predicting wireless mmWave massive MIMO channel characteristics using machine learning algorithms," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–12, Aug. 2018.

[23] C. Xue, T. Zhou, H. Zhang, L. Liu, and C. Tao, "Deep learning based channel prediction for massive MIMO systems in high-speed railway scenarios," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Helsinki, Finland, Apr. 2021, pp. 1–5.

[24] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 802–810.

[25] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatial-temporal data," in *Proc. ACM SIGSPATIAL Interface Conf. Adv. Geo. Infor. Sys.*, Los Angeles, CA, USA, 2016, pp. 1–4.

[26] L. Fang, X. Cheng, H. Wang, and L. Yang, "Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3091–3101, Aug. 2018.

[27] T. Zhou, C. Tao, S. Salous, L. Liu, and Z. Tan, "Channel sounding for high-speed railway communication systems," *IEEE Commun. Mag.*, vol. 53, no. 10, pp. 70–77, Oct. 2015.

[28] K. Guan *et al.*, "Towards realistic high-speed train channels at 5G millimeter-wave band—Part I: Paradigm, significance analysis, and scenario reconstruction," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9112–9128, Oct. 2018.

[29] D. He *et al.*, "Channel measurement, simulation, and analysis for high-speed railway communications in 5G millimeter-wave band," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3144–3158, Oct. 2018.

[30] Y. Liu, C.-X. Wang, J. Huang, J. Sun, and W. Zhang, "Novel 3-D nonstationary mmWave massive MIMO channel models for 5G high-speed train wireless communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2077–2086, Mar. 2019.

[31] T. Zhou, C. Tao, S. Salous, and L. Liu, "Geometry-based multi-link channel modeling for high-speed train communication networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1229–1238, Mar. 2020.

[32] T. Zhou, Y. Yang, L. Liu, C. Tao, and Y. Liang, "A dynamic 3-D wideband GBSM for cooperative massive MIMO channels in intelligent high-speed railway communication systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2237–2250, Apr. 2021.

[33] L. Tian, X. Yin, Q. Zuo, J. Zhou, Z. Zhong, and S. Lu, "Channel modeling based on random propagation graphs for high speed railway scenarios," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sydney, NSW, Australia, Sep. 2012, pp. 1746–1750.

[34] T. Zhou, C. Tao, S. Salous, Z. Tan, L. Liu, and L. Tian, "Graph-based stochastic model for high-speed railway cutting scenarios," *IET Microw., Antennas Propag.*, vol. 9, no. 15, pp. 1691–1697, Dec. 2015.

[35] B. Ai *et al.*, "Radio wave propagation scene partitioning for high-speed rails," *Int. J. Antennas Propag.*, vol. 2012, pp. 1–7, Sep. 2012.

[36] T. Zhou, C. Tao, S. Salous, and L. Liu, "Measurements and analysis of angular characteristics and spatial correlation for high-speed railway channels," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 357–367, Feb. 2018.

[37] T. Zhou, H. Li, Y. Wang, L. Liu, and C. Tao, "Channel modeling for future high-speed railway communication systems: A survey," *IEEE Access*, vol. 7, pp. 52818–52826, 2019.

[38] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[39] P. Campolucci, A. Uncini, and F. Piazza, "Causal back propagation through time for locally recurrent neural networks," in *Proc. IEEE Int. Symp. Circuits Syst. Circuits Syst. Connecting World*, Atlanta, GA, USA, May 1996, pp. 531–534.

[40] A. Garg and K. Tai, "Comparison of regression analysis, artificial neural network and genetic programming in handling the multicollinearity problem," in *Proc. Int. Conf. Modern, Identificat. Control*, Wuhan, China, Jun. 2012, pp. 353–358.

[41] T. Zhou, Y. Wang, C.-X. Wang, S. Salous, L. Liu, and C. Tao, "Multi-feature fusion based recognition and relevance analysis of propagation scenes for high-speed railway channels," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8107–8118, Aug. 2020.

Tao Zhou (Member, IEEE) received the B.E. degree from the Changchun University of Science and Technology, Changchun, China, in 2009, and the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2016.

From 2014 to 2015, he was a visiting Ph.D. student at the Centre for Communication Systems, School of Engineering and Computing Sciences, Durham University, U.K. Since 2016, he has been an Associate Professor with the School of Electronics and Information Engineering, Institute of Broadband Wireless Mobile Communications, Beijing Jiaotong University. His current research interests include propagation channel characterization, channel sounding, and modeling for high-speed railway communication systems.

Haitong Zhang received the B.S. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2019, where she is currently pursuing the M.S. degree in information and communication systems. Her current research interests include wireless channel characterization and applications of machine learning in wireless communications.

Bo Ai (Fellow, IEEE) is the Professor and the Doctoral Supervisor of Beijing Jiaotong University. He is also the Deputy Director of the State Key Laboratory of Rail Traffic Control and Safety. He has published six Chinese academic books, three English books, and over 150 IEEE journal articles. His research results has been involved in five national standards. He is mainly engaged in the research and application of the theory and core technology of broadband mobile communication and rail transit dedicated mobile communication systems (GSM-R, LTE-R, 5G-R, and LTE-M).

He is the fellow of Chinese Institute of Electronics, China Institute of Communications, and IET; the Chair of IEEE BTS Xi'an Branch; the Vice Chair of IEEE VTS Beijing Branch; an IEEE VTS Distinguished Lecturer; and an Expert of the 5G Industry Expert Group of the China Mobile Group Technical Advisory Committee and the 6G Group in China. He has obtained 13 international paper awards, including IEEE VTS Neil Shepherd Memorial Best Propagation Award and IEEE GLOBECOM 2018 Best Paper Award; 32 invention patents, 23 proposals adopted by the ITU and 3GPP; and nine provincial and ministerial-level science and technology awards.

Chen Xue received the B.S. degree in communication engineering and the M.S. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2018 and 2021, respectively. His research interests include wireless channel characterization and applications of deep learning in wireless communications.

Liu Liu (Member, IEEE) received the B.E. and Ph.D. degrees from Beijing Jiaotong University (BJTU), Beijing, China, in 2004 and 2010, respectively.

He was a Post Ph.D. Researcher with the School of Electronics and Information Engineering, Institute of Broadband Wireless Mobile Communications, BJTU, from 2010 to 2012, where he has been a Full Professor since 2018. His current research interests include channel measurement and modeling for different propagation environments and signal processing of wireless communication in time-varying channel.