Spatio-Temporal Spectrum Load Prediction Using Convolutional Neural Network and ResNet

Xiangyu Ren, *Graduate Student Member, IEEE*, Hamed Mosavat-Jahromi, *Member, IEEE*, Lin Cai, *Fellow, IEEE*, and David Kidston, *Senior Member, IEEE*

***Abstract*—Radio spectrum is a limited and increasingly scarce resource, which motivates alternative usage methods such as dynamic spectrum allocation (DSA). However, DSA requires an accurate prediction of spectrum usage in both time and spatial domains with minimal sensing cost. In this paper, we propose NN-ResNet prediction model to address this challenge in two steps. First, in order to make the best use of the sensors in the region, we deploy a deep learning prediction model based on convolutional neural networks (CNNs) and residual networks (ResNets), to predict spatio-temporal spectrum usage of the region. Second, to reduce sensing cost, the nearest neighbor (NN) interpolation is applied to recover spectrum usage data in the unsensed areas. In this case, fewer sensors are needed for prediction with the help of the reconstruction procedure. The model is verified through groups of comparison simulations in terms of the sensors' sparsity and the number of transmitters involved. In addition, the proposed model is compared with CNN and ConvLSTM prediction model. The results show that the proposed NN-ResNet model maintains a lower error rate under various sparse sensor circumstances.**

***Index Terms*—DSA, spatio-temporal prediction, CNN, ResNet, sparse sensor prediction.**

I. Introduction

RADIO spectrum is vital to wireless communication and at a premium. Spectrum scarcity became a major problem with the explosive growth of mobile services. Traditionally, the use of each spectrum band is statically allocated to a licensee over wide geographical regions within a long time duration, from months to years depending on the purposes. With static spectrum allocation and the proliferation of wireless services, the lack of available spectrum is becoming a severe problem.

Cognitive radio (CR) and dynamic spectrum allocation (DSA) are promising approaches to relieve spectrum scarcity [2]. CR allows a secondary user (SU) to use the *spectrum holes* or *white spaces* by channel sensing [3]–[5]. The cognitive radio system first senses the usage status of the channel, then allocates the idle channel to the SU. As the spectrum sensing procedure is time-consuming and may be unreliable, such as false detection and false alarm, due to channel fluctuations caused by path loss, shadowing, fading, or hidden-terminal issues, DSA based on spectrum prediction is a complementary approach. Predicting the activities in a frequency band in a specific location would allow it to be allocated to additional licensees for a short period of time it is predicted to be unused. Consequently, the spectrum can be efficiently utilized and the resources can be shared.

To apply DSA, it requires the knowledge of the spectrum usage pattern both temporally and spatially of the region [6]. For temporal prediction, the spectrum usage is predicted in the time domain by extracting the temporal correlation of the received signal power in the past. Accurate temporal spectrum prediction can also simplify the sensing process to save sensing time and energy, and improve sensing efficiency for CR [7]. On the other hand, the spatial prediction of the spectrum usage can provide the needed insights on spectrum occupancy and interference management in an area of interest, given the sensed data of a few locations nearby. With a comprehensive understanding of spectrum utilization temporally and spatially, we can further analyze the signal, interference, and quality-of-service for the whole region.

Generally speaking, spatial and temporal prediction of spectrum usage patterns are viewed as separate research topics. In the spatial domain prediction, the estimation of the target transmitter's location is based on the measurement of several parameters, e.g., received signal strength (RSS), time of arrival (TOA), and direction of arrival (DOA) [8]–[10]. Estimation models such as maximum likelihood estimation (MLE), ray tracing model [11]–[13] etc. have been later applied for localization. Such approaches require a large number of sensors to collect spectrum usage data and their performances are highly dependent on the accuracy and completeness of the data.

With the development of machine learning and neural networks, research interests have been more focused on the time domain prediction. Service providers are interested in having the knowledge of spectrum usage patterns over a period of time. Conventionally, autoregressive models based on Kalman filter are used for predicting the spectrum holes [14]–[16]. Recently, it has been shown that machine learning methods are useful for making predictions in the time domain. Support vector machines (SVM) and artificial neural networks (ANN)

Manuscript received September 29, 2021; accepted December 21, 2021. Date of publication December 29, 2021; date of current version June 9, 2022. This work was supported in part by the Communications Research Centre (CRC) Canada, the Natural Sciences and Engineering Research Council of Canada (NSERC), and Compute Canada. Part of this work has been presented at the Global Communication Conference (GLOBECOM 2020) [DOI: 10.1109/GLOBECOM42002.2020.9348001] [1]. The associate editor coordinating the review of this article and approving it for publication was Z. Xiao. *(Corresponding author: Lin Cai.)*

Xiangyu Ren and Lin Cai are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: jamesrxy@uvic.ca; cai@ece.uvic.ca).

Hamed Mosavat-Jahromi is with the Optical System Competency Centre, Huawei Technologies Canada Research Center, Ottawa, ON K2K 3J1, Canada (e-mail: seyedh.jahromi@huawei.com).

David Kidston is with the Innovation, Science and Economic Development Canada, Communications Research Centre Canada, Ottawa, ON K2H 8S2, Canada (e-mail: david.kidston@canada.ca).

Digital Object Identifier 10.1109/TCCN.2021.3139030

are among the most popular methods [17]–[19]. Other methods like logistic regression, and genetic algorithms are utilized to extract the data features and make predictions.

On the other hand, both spatial and temporal prediction requires a large number of sensors to sense the spectrum usage data to obtain good performances, which leads to high sensing cost. However, only a limited number of sensors can be deployed in practice in terms of economic and privacy issues. Therefore, how to accurately predict spectrum usage in both the time and spatial domains at the same time with minimum sensing cost remains a challenging open issue.

Deep learning is a cutting edge tool that has shown its success in many applications including both spatial and temporal predictions. Two main types of networks are widely used in deep learning: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are mainly exploited to capture spatial features in image classification and image analysis [20], [21] while RNNs are more focused on capturing temporal features like time series prediction [22]. In recent years, residual networks [23] (ResNets), a structural modification of CNN, have attracted attention thanks to their ability to address the vanishing gradient problem as the network goes deeper. Thus, ResNets are often used to address problems in the spatio-temporal domain [24]–[26]. Deep learning has been explored heavily to predict the spatio-temporal patterns in many literature, e.g., for road traffic prediction [20]–[22], but has not been applied to spectrum load prediction. With simple manipulation, the collected spectrum usage data of a given region can be organized into power matrices. Similar to the road traffic images where each pixel represents the traffic volume and location, the values in the power matrices represents the received signal power and sensor's location. On the other hand, unlike the road traffic data where the exact locations and number of vehicles are involved in the images, the received signal power at sensors is an aggregation of multiple signal sources without the knowledge of their exact location(s) and the number of users. In addition, only a limited number of sensors can be deployed in practice which introduces new challenges not addressed in the literature.

In this paper, we focused on solving this joint spatio-temporal prediction problem with a limited number of sensors. The main contributions of this paper are two-fold.

- First, to address the open issue of predicting spectrum load spatially and temporally at the same time, we proposed a novel neural network model combining both CNN and ResNet, which can accurately predict the spectrum usage pattern of a large region. Sensors are deployed in a region with a grid topology to record the signal power of transmitters. The recorded data are organized into matrices with their spatio-temporal information and later used for prediction.
- Second, on top of the above spatio-temporal issue, we further address a practical challenge, how to maintain prediction performance with a limited number of sensors, where the power matrices are incomplete due to the lack of active sensors. To reconstruct the power matrix, we applied two approaches, matrix completion and local interpolation, and their performance are compared and analyzed. With the proposed reconstruction procedure, the prediction model is capable of providing accurate prediction results with fewer sensors.

To verify the proposed model's performance under different sparse sensor circumstances, three control groups with different sparsity of 0.4, 0.5, and 0.6 are compared and analyzed. In addition, we also considered the scenario where multiple transmitters are involved in the target region. How multiple transmitters will affect the performance is analyzed in a similar scheme with different levels of sparsity. Finally, the proposed prediction model is compared with the CNN prediction model to further verify its performance.

In the following sections, related works on temporal prediction and spatio-temporal prediction is introduced in Section II. Preliminaries and system model are introduced in Section III and Section IV respectively. Section V discusses the details of the proposed spatio-temporal prediction model based on deep learning. Introduction to the dataset, set up of experiment, and numerical results of the proposed prediction model are analyzed in Section VI. The concluding remarks and further research are summarized in Section VII.

II. Related Work

A. Temporal Prediction

1) ANN Based Method: Accurate prediction of spectrum occupancy is fundamental to the DSA system. In [19], an ANN-based prediction system is proposed to support opportunistic DSA for SUs. The system predicts the spectrum occupancy state of PUs in terms of busy and idle state. Once a request for transmission with time interval τ arrives at an SU, the SU makes the transmission decision based on the prediction result of the system. If the predicted remaining time is longer than the requested time interval τ, the transmission request is granted. The ANN architecture is decided based on a range of comparison results where the number of hidden layers and the number of perceptron nodes vary. In the end, a simple single-perceptron ANN architecture is selected due to its high accuracy and low time cost. The system has a low time complexity of $O(N)$ where N is the number of input features. In [19], 12 input features are selected, namely current life of idle period, SU transmission request time interval, and 10 previous idle periods. Sigmoid activation function, binary cross entropy loss function, and Nadam optimizer are applied in the perceptron. The system provides a binary output where 0 represents transmission denial and 1 represents transmission approval.

2) LSTM-Based Method: Long-term spectrum load prediction is also important to the DSA system. The authors in [27] proposed an LSTM-based prediction model to predict the fixed length and multiple-slot future spectrum load with previously measured data over a wide spectrum band. The LSTM structure is designed to have long-term memories of the input. LSTM learns the sequential correlations of the input data by a repeating module consisting of four components namely, memory cell, forget gate, input gate, and output gate.

- The memory cell maintains the history information of previous time slot that flows along the chain with minor linear changes.

- The forget gate decides what information needs to be dropped from the memory cell.
- The input gate decides what new information to be stored in the memory cell.
- The output gate decides what the output of the module.

The LSTM model in [27] takes in hybrid historical spectrum data with joint multi-dimensional dependencies in space, frequency and time to predict a long-term usage pattern of multiple channels at multiple locations.

B. Spatio-Temporal Prediction

To our best knowledge, how to predict the spatio-temporal features of spectrum load remains an open issue. Although road traffic load prediction has been explored heavily using deep learning tools, how to predict the spatio-temporal features of spectrum load is much more difficult.

1) Local CNN Model: To enable spatio-temporal feature extraction using CNN, a local CNN model is proposed in [26]. Generally speaking, in terms of regression problems, CNNs consist of two components including feature learning and prediction. The feature learning part will impose convolution and pooling filters to the input images, where the inner features are detected. Followed by a fully connected layer as the predictor, which gives prediction result at each pixel. The aforementioned two main components of CNNs consist of several subcomponents as follows.

- *Convolution Layers:* The convolution operation is performed on the input data by convolution kernels to extract the features of the input data.
- *Activation Functions:* The output of a convolution layer is passed through a nonlinear activation function, e.g., `ReLU`, `tanh`, and `sigmoid`.
- *Pooling Layer:* Pooling layer is responsible for a down-sampling operation which reduces the dimension of feature maps. The reason behind the pooling layer is to make the feature maps invariant to small shifts and distortions.
- *Fully-Connected Layer:* The fully-connected layer is mainly responsible for prediction. It is of note that each fully-connected layer is followed by an activation function.

In [26], a batch of images of successive time slots $I^{M\times N\times T}$ are forwarded to the local CNN model. At time slot *t*, the output of *k*th convolution layer Y_t^k is obtained with extracted spatial information. Then it is passed to a flatten layer to transform the tensor Y_t^k into a feature vector s_t at time *t*. Finally, a fully connected layer is applied to obtain the original size-output.

$$\hat{s}_t = f(W_t^{fc} s_t + b_t^{fc}), \tag{1}$$

where W_t^{fc} and b^{fc} are the learnable parameters of the fully connected layer and $f(\cdot)$ denotes the activation function used. In this way, for a range of successive time slots, both spatial and temporal information are kept.

2) SeqST-ResNet: ResNet [28] is initially designed to address the gradient vanishing problem [29] caused by the increment of network depth via residual blocks as shown in Fig. 1. The skip connection in a residual block can help to improve the network's performance by allowing the gradient to flow through shortcut and ensuring the higher layer will perform at least in the same level as the lower layer, not worse.

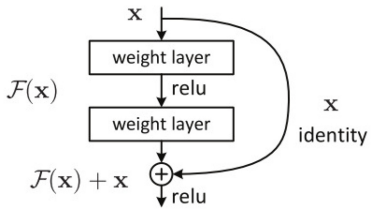


Fig. 1. Residual block.

$$\boldsymbol{X}_{out} = F(\boldsymbol{X}_{in}) + \boldsymbol{X}_{in} \tag{2}$$

Taking the advantages of ResNet in deep neural networks, Zhai *et al.* [30] proposed SeqST-ResNet to predict the traffic flow by capturing the spatial and temporal dependencies in historical data. In SeqSt-ResNet, sequences of task images $\{\mathbf{X}_{t-n}, \mathbf{X}_{t-n}, \ldots, \mathbf{X}_{t-1}\}$ are forwarded to the same network where each convolution layer is followed by multiple residual blocks.

In our work, a spectrum load spatio-temporal prediction model combining both CNN and ResNet is proposed. With the help of CNN and ResNet, the spatio-temporal features in the power matrices are captured and used for prediction.

III. Preliminaries

A. Path Loss Model

Path loss characterizes the attenuation of signal power as the propagation distance increases. The linear path loss is defined as the ratio of the transmitted signal power p_t over the received signal power p_r, i.e., $PL = p_t/p_r$. Specifically, path loss at a distance d in dB can be modeled by

$$PL(d) = PL(d_0) + 10\alpha \log_{10}(d/d_0) + X_g, \tag{3}$$

where d_0, $PL(d_0)$, α, and X_g are the reference distance, path loss at d_0, path loss exponent, and shadowing, respectively. α is determined by the propagation environment. The average received power with transmitter at distance d can be calculated by

$$p_r = p_t \cdot d^{-\alpha} + n, \tag{4}$$

where n is a zero-mean normal distribution caused by shadowing component X_g.

B. Data Reconstruction Methods

The key idea of data reconstruction is to model the dependencies between the missing values and the known ones. Two widely used techniques are filtering and matrix completion [31]. The performance of the two approaches are compared and analyzed in Section V.

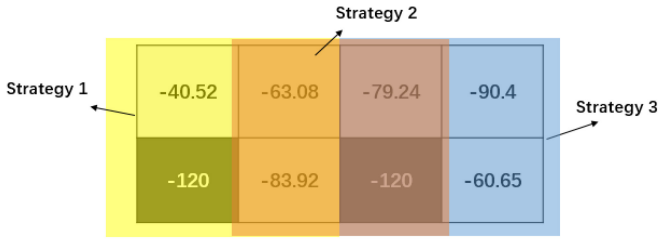


Fig. 2. Example of how NN interpolation works.

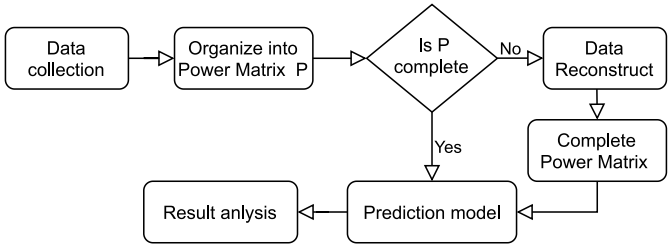


Fig. 3. Flow chart of the system model.

1) Nearest Neighbor Interpolation: Various filtering methods have been proposed to extract the local interdependence of the matrix such as median filtering, moving average, Kriging, and nearest neighbor etc [32], [33]. According to the path loss model in Eq. (3), it is clear that the wireless signals decay exponentially over distance. Since the average received power (ignore fast fading) in nearby locations are close to each other, experiencing higher spatial correlation, which will be considered in data reconstruction and spatial predictions. In other words, only the sensors close to the unsensed area are useful for data recovery. Thus, the nearest neighbor (NN) interpolation method is selected to reconstruct the power matrix under the sparse sensor condition. The NN interpolation uses a sliding window of size 2×2 to recover the missing values. An example of the filter is shown in Fig. 2.

2) Matrix Completion: Since in practical environments, the wireless transmission will mutually interfere with each other, so typically each transmission will occupy a certain spectrum exclusively. Given the sparsity of sensors due to either data loss or deployment cost, the power matrices can be incomplete under some circumstances. The matrix completion (MC) method naturally solves the problem. It focused on formulating an optimization problem and recover the missing values by minimizing the rank of the target matrix [34]–[36].

$$\begin{aligned} \min_{Z} \quad & \sum_{\text{observed } i,j} \left(X_{i,j} - Z_{i,j}\right)^2 \\ \text{s.t.} \quad & \text{rank}(Z) = r, \end{aligned} \tag{5}$$

where X is the incomplete matrix and Z is the recovered matrix. However, the constraint rank(Z) $= r$ makes the problem non-convex and high computation cost. It is replaced with the nuclear norm of Z which is the tightest relaxation of rank(Z) [37]. Therefore, the optimization problem can be rewritten as follows in Lagrange form:

$$\min_{Z} \frac{1}{2}\|P_{\Omega}(X) - P_{\Omega}(Z)\|_F^2 + \lambda||Z||_*, \tag{6}$$

where $P_{\Omega}(\cdot)$ is the projection on the observed entries (the known values in the matrix), $||\cdot||_F$ is the Frobenius norm, and $||\cdot||_*$ is the nuclear norm. In this way, the non-convex problem is relaxed to a non-constrained semi-definite program (SDP) problem with much lower computation cost.

IV. System Model

The system model of this work is shown in Fig. 3. In this paper, we consider the following scenario: In a rectangular-shaped region, 100 sensors were deployed uniformly into a 10×10 array, which divides the region into 81 grids, and each grid is called an area. The size of an area is 1 km in width and 2 km in length. Sensors will sense and record the spectrum usage data periodically. The length of each time slot is 150 ms. Each transmitter keeps emitting signals while taking a random walk mobility pattern. In the mobility model, the transmitter will move at a constant speed following the same direction for a duration of 2 seconds, and then randomly select a new direction and speed. The selection of the speed is within the range between 0.69 m/s and 1.44 m/s. Note that the mobility range of each transmitter over the experiment duration is constrained in a small part of the region mainly affecting its neighboring sensors, and the trajectory of the transmitter is continuous both spatially and temporally. The sensed data at the sensors contains the *reception time*, *received packet's label*, and *received power*. Each experiment lasts for 15 minutes, during which all the spectrum usage data of the whole region is recorded by the sensors. In each time slot, the data is organized into the form of a 10×10 matrix and each element represents the received power.

TABLE I
Dataset Configuration

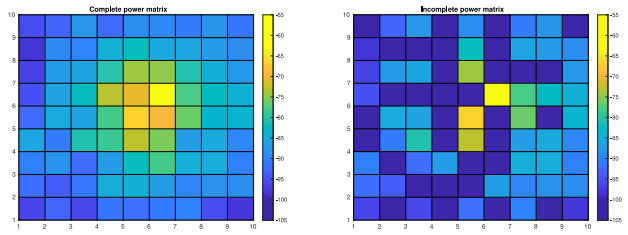
Case	Quantity	Bandwidth (kHz)	Power (mW)	Range (km)
1.	27	50	37	20
2.	25	25	37	20

Two different cases are set up to obtain our dataset. The configuration of the two cases are in Table I.

In Case-1, 23 transmitters move continuously within a 20×20 km square region. The power level of the signal is 37 mW and the utilized spectrum bandwidth is 50 kHz. Each transmitter uses the same channel during the entire experiment and the sensors record the received power of each transmitter accordingly.

In Case-2, the number of transmitters changed to 27 and the utilized spectrum bandwidth changed to 25 kHz. The rest of the configurations remain the same.

To make the dataset an appropriate input to the prediction model, we focus on the received power at different sensors. In the case of an idle time slot when nothing is recorded, the corresponding time slot will be filled by noise (-110 dBm). At the end of the unification process, a power matrix of the 100 sensors, $\mathbb{P}_{10\times10\times T} \in \mathbb{R}^3$ can be obtained, where T is the total number of time slots. The prediction model is trained and tested against the organized dataset. A total length of 15 minutes of data is applied to verify the model. Overall, there are 6000 time slots in the power matrix. We allocated 80% of the



(a) Complete power matrix (b) Incomplete power matrix

Fig. 4. An example of complete and incomplete spectrum sensing data.

TABLE II
DEFINITION OF SYMBOLS

Symbols	Meaning
R	Sparsity, the ratio of unused sensors
$\mathbb{P}$	Power matrix
D_0	Complete dataset
$\tilde{D}$	Incomplete dataset
$\hat{D}$	Reconstructed dataset
D_1	Incomplete dataset with $R = 0.4$
D_2	Incomplete dataset with $R = 0.5$
D_3	Incomplete dataset with $R = 0.6$
S_c	Working sensor
S_u	Unused sensor
A	Single transmitter scenario
B	Multiple transmitters scenario

dataset to the training phase, and the rest for the test phase. This pile of data are referred to as the complete sensor data D_0, where all of the 100 sensors' data are used for prediction. An example of complete sensor data D_0 is shown in Fig. 4a.

In order to verify the capability of the proposed model under the circumstance where the sensors are sparse, spectrum usage data are collected randomly from the sensor network. The sensors selected to collect data are referred to as working sensors, S_c. The unused sensors are referred to as S_u. Therefore, the organized data of each time slot is incomplete with `null` value in some elements. The incomplete dataset is referred as $\tilde{D}$, as shown in Fig. 4b, where the navy blue color stands for the unsensed area. To use $\tilde{D}$ as the input of the prediction model, the `null` first needs to be recovered to obtain a complete dataset. Considering that the received power at each sensor is geographically related to its neighboring sensors, the local interpolation approach is adopted to recover the missing values in the power matrix after comparing with the matrix completion algorithm. More details about the data reconstruction work can be found in Section VI-A. The reconstructed dataset $\hat{D}$ is then forwarded for prediction. The symbols used in the paper to describe the problem are summarized in Table II.

V. SPATIO-TEMPORAL PREDICTION USING DEEP LEARNING

In this Section, the construction of our prediction model is explained in three parts: the first part compares the performance of two data reconstruction approaches from both accuracy and time aspects, and one approach is selected in the prediction model. The second part illustrates the structure of the proposed prediction model. The last part describes how the model parameters are determined.

TABLE III
RELATIVE ERROR OF TWO APPROACHES

Sparsity	NN interpolation	Time (s)	MC	Time (s)
R = 0.4	3.3397	0.926	4.8545	99.853
R = 0.5	3.5638	0.983	7.6761	124.586
R = 0.6	5.0581	1.025	21.9675	153.794

A. Reconstruction of Incomplete Data

Both local interpolation method and matrix completion algorithm show satisfactory results in reconstructing images. In this section, we first compare the two approaches from two aspects, error rate and time cost, and select the better approach for data reconstruction. Relative error (RE) is used for measuring the error rate.

$$RE = \frac{||X - \hat{X}||_F \times 100}{||X||_F}, \tag{7}$$

where $||X||_F = \sqrt{\mathrm{Tr}(XX^T)}$. The two approaches are compared under various sparse rate 0.4, 0.5, and 0.6. The sparse rate represents the proportion of the missing matrix values. The higher the sparse rate is, the less sensors are used for data collection.

1) Nearest Neighbor Interpolation Method: The local interpolation approach is used in reconstructing the incomplete power matrix. According to the path loss model in Eq. (4), the missing values are more related to their neighbor sensors given the comparatively lower distance. The interpolation model to recover the missing values can be formulated as follows:

$$\begin{aligned} p_r &= \omega_1(d_i)p_r^{[1]} + \omega_2(d_2)p_r^{[2]} + \omega_3(d_3)p_r^{[3]} + n \\ &= \mathbf{W} \cdot \mathbf{P_r} + n, \end{aligned} \tag{8}$$

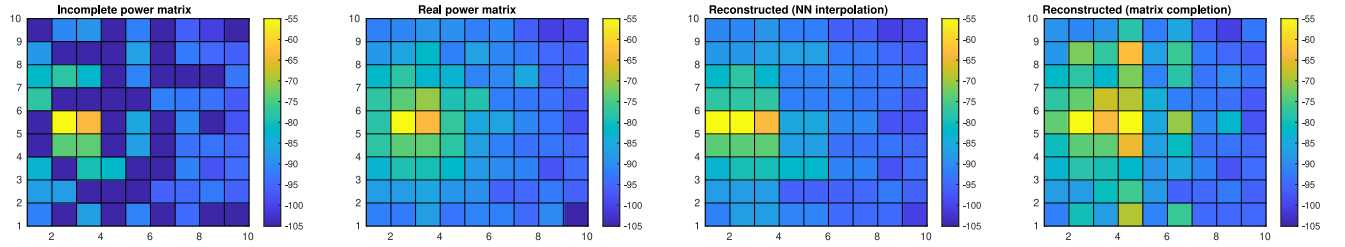
where $p_r^{[i]}$ is the signal power received at a neighbor sensor, $\omega_i(d_i)$ is the weight assigned to the sensor proportional to the distance, and *n* is the Gaussian noise caused by shadowing.

The reconstruction results are shown in Fig. 5 and the numerical evaluations are summarized in Table III. It can be seen that the nearest neighbor interpolation is able to recover the missing values with low error rate and low time cost. It achieves a 3.9872 error rate at speed of 0.926 seconds per hundred matrices in average regardless of the sparse rate.

2) Matrix Completion Method: As for the MC method, we first re-formulate our problem as follows.

$$\min_{\hat{D}} \ \frac{1}{2}\left\| P_\Omega(D') - P_\Omega(\hat{D}) \right\|_F^2 + \lambda||\hat{D}||_*, \tag{9}$$

where the input of the algorithm is the observed incomplete power matrix D_1 and the goal is to obtain the reconstructed power matrix $\hat{D}$. In order to solve the non-constrained SDP problem, a solver named *Templates for First-Order Conic Solvers* (TFOCS) proposed by Becker *et al.* in [36] is applied. The solver is capable of handling cone convex problems flexibly. An example of the reconstructed matrix is shown in Fig. 5 and the numerical evaluations are summarized in Table III. The



(a) Incomplete power matrix (b) Real power matrix (c) Reconstructed (NN interpolation) (d) Reconstructed (matrix completion)

Fig. 5. Power matrix reconstruction results comparison.

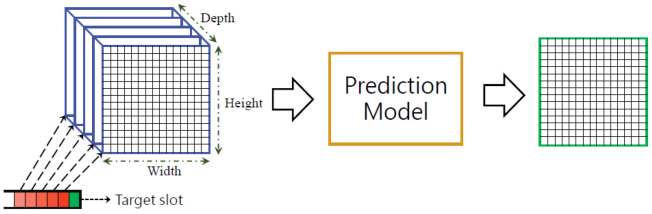


Fig. 6. Prediction model.

matrix completion approach, however, only maintains a reasonable error rate at 4.8545 when the sparse rate is 0.4. Both error rate and time cost increases as the sparse rate raises. The reason why MC underperforms in this scenario is that it fails to emphasize the local dependence among the missing values. TFOCS gives the optimal solution to the optimization problem under a global view including power data that are spatially far away from the missing point. However, these data can hardly contribute to the meaningful estimation of the missing values, which degrades the reconstruction performance.

In conclusion, the nearest neighbor interpolation approach outperforms the matrix completion method in both the error rate and time cost aspects. So it is chosen for the data reconstruction procedure in the proposed model.

B. Prediction Model

In order to predict spectrum usage in a future time slot, we focus on the usage of previous time slots as shown in Fig. 6. In this figure, width and height of the input data correspond to the sensor grid size (10×10), and the depth is how many previous time slots are being used in the prediction process, i.e., the window length. For a certain window length and according to the input, the prediction model should be trained to generate an output, $\hat{X}_t$ that has the minimum error compared with the groundtruth, X_t. During the process of searching for the minimum error, the weights of the model is updated. The mean squared error function is chosen as the loss function of the prediction model. It is defined as:

$$\mathcal{L}(\theta) = \left\| \hat{X}_t - X_t \right\|_2^2, \tag{10}$$

where θ is the trainable parameters used in the prediction model.

The prediction model consists of convolution blocks and residual blocks. The convolution blocks are used in the model to capture the spatial correlations among the sensors. As mentioned earlier, CNNs is powerful in capturing spatial features of an image (here a 10×10 power matrix). Each convolution block is a combination of convolution layers, activation function layers, and pooling layers. The convolution layer is responsible for capturing the features via a number of filters or convolutional kernels.

Each filter is responsible for extracting one feature from the input image. Due to the size of the image, the filter size in the proposed model is selected to be 2×2. Intuitively, the more filters in a convolution block are set, the better performance it could achieve. However, increasing the number of filters also means increasing the number of weights to be calculated and updated, which results in heavy training cost. In Section V-C, more details about determining the number of filters are explained. In the convolution blocks, the "`same`" padding strategy has been employed in the pooling layer. This padding procedure is necessary when the size of the pooling window does not match the image size. "`same`" padding adds an extra zero-column to the matrix, so that the last column of data is preserved.

Increasing the number of convolution blocks results in a deep CNN which may encounter the vanishing gradient problem as introduced in Section II. According to [24], [38], ResNet can solve this problem. As shown in Fig. 1, the skip connection structure that bypass the convolution layers helps the neural network focus more on the variation of data $F(x)$ rather than the output $H(x)$. For instance, the initial input is denoted as x_i, the output is denoted as $H(x_i) = x_{i+1}$, then within one residual block, the output can be written as:

$$x_{i+1} = H(x) = F(x_i) + x_i, \tag{11}$$

where $F(x_i)$ is the combination of convolution and activation nonlinear mapping process. With the addition of x_i, the difference between the output $H(x)$ and input x increases, and thus alleviates the reduction of the gradient. In the proposed model several residual blocks are stacked to prevent the vanishing gradient problem as the neural network goes deeper.

As shown in Fig. 7, each residual contains two activation functions and two convolution blocks. The deployed activation function in the residual blocks is the linear rectifier, `reLU`. Note that each activation function block is followed by a `BatchNormalization` block to normalize the data in each step and improve the convergence of the model.

C. Configurations of the Model

In this subsection, more detailed explanations about configuring the model parameters including window length, number

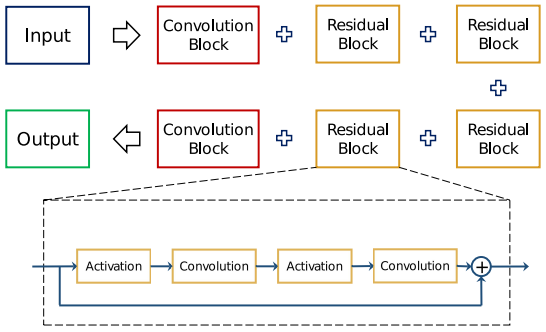


Fig. 7. Block diagram of the prediction model, Convolution, and residual blocks.

of ResNet blocks and number of filters are addressed. Root mean squared error (RMSE) and mean absolute percentage error (MAPE) are used to evaluate the impact of these parameters to the model's performance.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{X}_i - X_i)^2}, \tag{12}$$

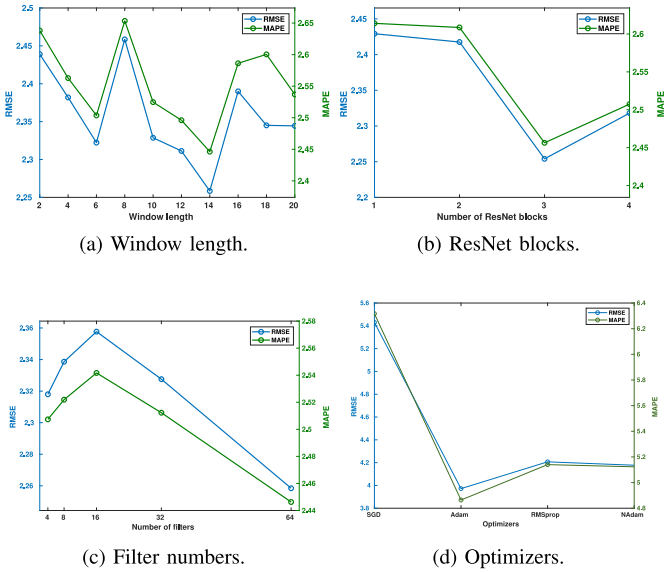
$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{X}_i - X_i}{X_i}\right| \times 100. \tag{13}$$

The data used to show how the model parameters are determined as an example comes from Case 2 described in Section IV.

Window Length: The window length parameter is highly dependent on the temporal correlations of the data. The length of each predicting window determines how long the historical data can be used for learning. The correlation between the spectrum pattern of the future time slot and each previous time slot varies due to spectrum occupancy, channel fluctuation, and transmitter mobility. If the data have a long memory, the more historical data are used, the more accurate the prediction is. Given the random walk mobility model, the transmitter will change its moving speed and direction after 2 seconds. Thus, only a certain range of the successive historical data are strongly related to the future time slot. Above that, the historical data may degrade prediction performance. In order to choose the optimal window length, a group of comparison experiments is conducted. Fig. 8(a) shows the comparison of RMSE and MAPE metrics with respect to the window length. The total number of convolutional filters and residual blocks in this figure are fixed to 64 and 4, respectively. The result shows that the minimum RMSE and MAPE is achieved when 14 previous time slots are utilized in the prediction process.

Number of ResNet Blocks: The skip connection idea in ResNet block helps to alleviate the gradient vanishing problem. To determine the number of ResNet blocks in the model, a comparison of model performance in terms of the number of ResNet blocks is given in Fig. 8(b). It can be observed that the minimum RMSE and MAPE is achieved with three ResNet blocks.

Number of Filters in Convolution Layer: Fig. 8(c) shows the results of the prediction model as the number of filters in



(a) Window length. (b) ResNet blocks.

(c) Filter numbers. (d) Optimizers.

Fig. 8. Determination of parameters of the proposed model.

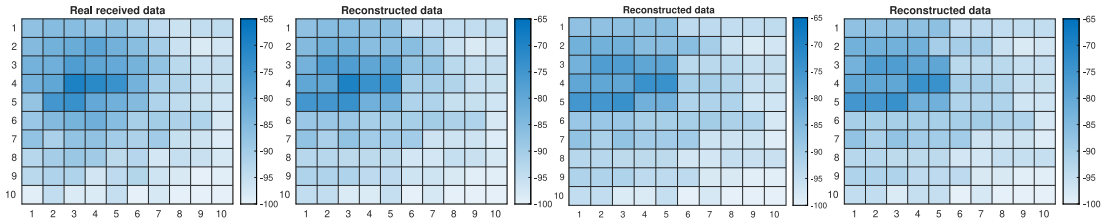
TABLE IV
MODEL PARAMETERS USED IN THE TRAINING PROCESS

Learning Parameters	Value
Width	10
Height	10
Depth	14
Number of filters	64
Filters' size	2×2
Number of epochs	60
Learning rate	0.0005
Batch size	32
Activation function	`tanh` , `ReLU`
Optimizer	`Adam`
Loss function	Mean Squared Error

the convolution blocks increases. Each filter extracts a type of feature from the image. As expected, increasing the total number of filters reduces both RMSE and MAPE. However, a large number of filters results in a deeper neural network, heavier computation costs, and severer over-fitting problem. Thus, the number of filters does not exceed 64 in the proposed prediction model.

Comparison of Optimizers: Fig. 8(d) shows the impact of applying different optimizers to the model's performance. Here we compared a few popular optimizers applied in deep learning [39] namely SGD, Adam, RMSprop, and NAdam. All of them converge at a similar speed, however, the Adam optimizer provides the best performance in terms of accuracy as shown in the figure. Therefore, the Adam optimizer is applied in our model.

In conclusion, the model parameters of the proposed model is summarized in Table IV. Since the deep learning-based model is naturally sensitive to datasets, adjusting these parameters according to different datasets is necessary. For instance, when the variation of spectrum usage pattern changes dramatically, a shorter window length is preferred, and vice versa. Therefore, modifications of the parameters may be needed



(a) Real power matrix (b) Reconstructed R=0.4 (c) Reconstructed R=0.5 (d) Reconstructed R=0.6

Fig. 9. Single transmitter scenario: Data reconstruction results of different levels of sensor sparsity.

Algorithm 1 Spatio-Temporal Spectrum Load Prediction Model

Input: collected sensor data
Output: spatio-temporal spectrum load prediction
procedure START
  Organize sensor data into $\mathbb{P}^{M\times N\times T}$.
  if $\mathbb{P}$ is incomplete **then**
    Apply data reconstruction procedure in Eq. (8).
    Obtain $\hat{\mathbb{P}}$.
  else
    Continue.
  Adjust model parameters according to $\hat{\mathbb{P}}$ ($\mathbb{P}$).
  Forward $\hat{\mathbb{P}}$ ($\mathbb{P}$) to NN-ResNet prediction model.
  Evaluate prediction results.

in different scenarios. Algorithm 1 describes the complete process of the proposed prediction model.

In this section, we described the proposed data reconstruction filter and spatio-temporal spectrum prediction model in details. In addition, an example to determine the model parameters by evaluating the model's accuracy with RMSE and MAPE is explained. In the next section, a group of simulations to verify the performance of the proposed model are explained and analyzed in details.

VI. Spectrum Usage Prediction Results

We first describe the configurations of the four types of datasets used for simulations in Section VI-A. Apart from one group of complete sensor dataset, three control groups with different sparsity $R = 0.4, 0.5, 0.6$ are established. Next, a detailed explanation of the simulation is introduced in Section VI-B. Four groups of datasets *complete sensor dataset* D_0, *incomplete dataset* D_1 with $R = 0.4$, D_2 with $R = 0.5$, and D_3 with $R = 0.6$ are used for comparison experiments. Last but not the least, in Section VI-C and Section VI-D, two different application scenarios: *single transmitter A* and *multiple transmitters B*, are analyzed respectively to explore the effect of multiple transmitters to the model's performance. Finally, in Section VI-F we compared the proposed NN-ResNet model with the original CNN model using the same datasets in both single and multiple transmitters scenarios.

A. Description of the Dataset

In this paper, the dataset generated from Case-1 is used for conducting the following simulations to test our model. First, the dataset with the complete usage of sensors is generated. Next, In order to verify the proposed model's capacity under sparse sensor circumstances, three control groups are generated following a similar procedure. Instead of using the complete set of sensors, a smaller number of sensors are selected randomly from the sensor networks. The sensor data are collected and organized as introduced in Section IV. To maintain a 10×10 power matrix, the unsensed entries in the power matrix are replaced with noise-level power values. These datasets are later used to verify the performance of the proposed model.

B. Configurations of the Experiments

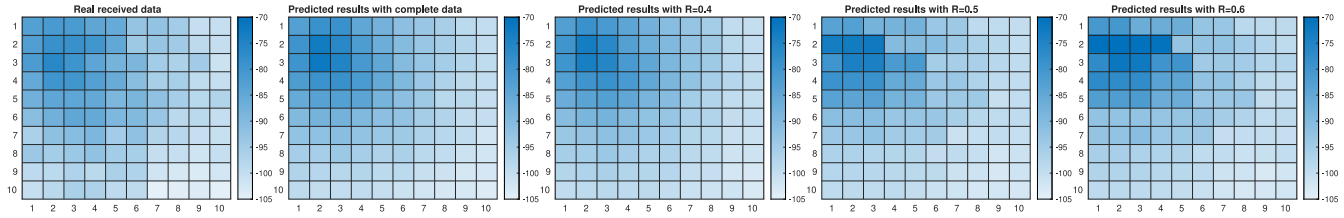
Two experiments, single transmitter scenario *A* and multiple transmitters scenario *B* are established to validate the performance of the proposed model. In each experiment, four groups of datasets are used for simulations. The complete sensor dataset D_0 uses all the sensors to collect spectrum usage data and the data reconstruction process is skipped for this group. Three control groups using the incomplete sensor dataset with different sparse rates are generated. D_1, D_2, D_3 has the sparsity of 0.4, 0.5 and 0.6, respectively. The control group datasets are first reconstructed using the NN interpolation methods before forwarded for prediction. The control groups are set up to examine the capability of the proposed model under different levels of sparsity.

All of the simulation results are compared with the real received signal power recorded at the sensors in both experiments. RMSE and MAPE are used to measure the error rate.

C. Experiment A: Single Transmitter Scenario

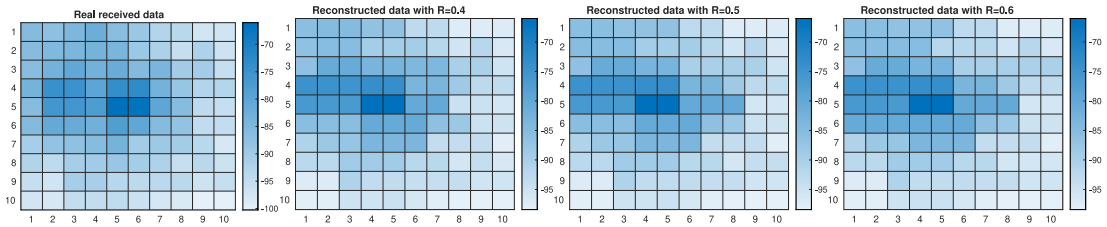
In this experiment, only one transmitter is considered in the target region. The transmitter takes a random walk mobility pattern as explained in Section IV. First, the datasets are collected from the sensors and categorized as $D_{A0}, D_{A1}, D_{A2}, D_{A3}$ respectively. The incomplete datasets are reconstructed before prediction. A few examples of the reconstructed datasets are shown in Fig. 9(b-d). In comparison, the real received data is shown in Fig. 9(a). It can be observed that as the sparsity grows, the performance of the proposed reconstruction method degrades slightly. In general, it does a good job in recovering the missing data. Especially in the area close to the hot spot where spectrum is used intensively.

Next, the reconstructed power matrices are forwarded to the proposed model for prediction. During the prediction process,



(a) Real power matrix (b) Prediction with D_{A0} (c) Prediction with D_{A1} (d) Prediction with D_{A2} (e) Prediction with D_{A3}

Fig. 10. Single transmitter scenario: prediction results.



(a) Real power matrix (b) Reconstructed R=0.4 (c) Reconstructed R=0.5 (d) Reconstructed R=0.6

Fig. 11. Multiple transmitters scenario: Data reconstruction results of different levels of sensor sparsity.

TABLE V
PREDICTION RESULTS EVALUATION SUMMARY

		Single transmitter				**multiple transmitters**			
		Complete	R=0.4	R=0.5	R=0.6	Complete	R=0.4	R=0.5	R=0.6
Incomplete data prediction v.s.	RMSE	-	9.0785	10.875	13.4725	-	10.0606	10.944	11.3066
Real power matrix	MAPE	-	6.6156	8.863	10.2573	-	8.2954	9.7504	10.13
CNN prediction v.s.	RMSE	4.115	4.3951	5.9837	6.0072	4.1257	4.5493	6.3481	6.9831
Real power matrix	MAPE	3.4068	3.42	4.928	4.9892	3.5679	4.2332	5.8176	6.8197
ConvLSTM prediction v.s.	RMSE	3.6883	4.2368	4.7652	5.3127	4.1027	4.4434	5.2821	5.591
Real power matrix	MAPE	2.8411	3.3278	4.5893	4.7376	3.3432	4.1376	5.1262	5.6179
NN-ResNet prediction v.s.	RMSE	3.7079	4.0907	4.2044	4.6983	4.0147	4.3512	4.9811	5.0775
Real power matrix	MAPE	2.9224	3.2831	4.1695	4.3158	3.3495	4.0310	4.8067	5.0498

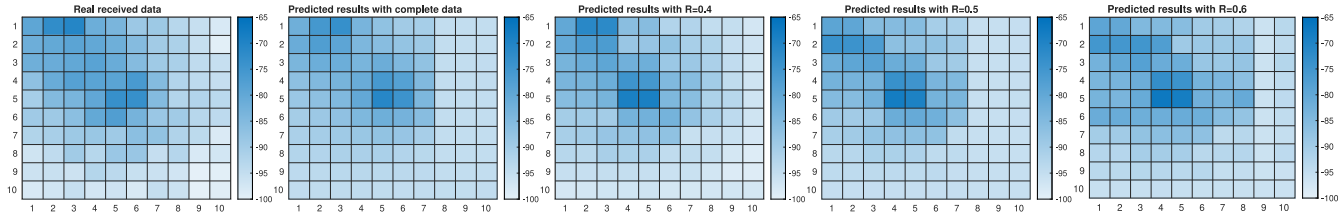
the previous 80% of the data is used for training and the rest for testing. Fig. 10(c-e) show the prediction results with the incomplete dataset D_{A1}, D_{A2}, and D_{A3} respectively. In comparison, Fig. 10(b) shows the prediction results with the complete data D_A and Fig. 10 (a) is the real received spectrum usage data. When comparing the first row and second row in Fig. 10, we can observe that the performances of the prediction model are dependent on the input matrices. In other words, more complete and accurate reconstructed power matrices lead to a more accurate prediction of the target region. The numerical evaluations of the prediction results are summarized in Table V.

It can be concluded that the proposed model is capable of predicting spectrum usage both spatially and temporally with the help of the reconstruction process, even when missing parts of the sensor data. It can be also concluded that the performance of the proposed model is dependent on the completeness of the input power matrices and the accuracy of the reconstruction process. The error rate is only 3.7079 when the complete sensor data D_{A0} were used. In this subsection, the performance of the proposed model of a single spectrum user scenario is verified and proved to be effective. In the next subsection, a more complicated scenario where multiple transmitters are involved to test the model.

D. Experiment B: Multiple Transmitters Scenario

In this experiment, two transmitters are considered as our analysis objects in the target region. Thus, the spectrum usage data is more likely to be missed in a sparse sensor circumstance. Following the same procedure introduced in the previous experiment *A*, datasets were collected and categorized into four groups: D_{B0}, D_{B1}, D_{B2}, and D_{B3}. The reconstruction process was applied to the incomplete datasets before prediction. The reconstructed results are shown in Fig. 11. Although more data about the spectrum usage were lost as the sparsity grows bringing more challenges to predictions, with the help of the reconstruction procedure, the lost data can be recovered to some extent.

These reconstructed power matrices are later used as the input to the proposed prediction model to predict the future spectrum usage. The prediction results are presented in Fig. 12. Fig. 12(c-e) show the prediction results of the incomplete dataset. In comparison, Fig. 12 (b) shows the prediction results with the complete data D_B and Fig. 12 (a) is the real received spectrum usage data. Similar to what we can observe from experiment *A*, it can be seen from Fig. 12 that the proposed model is capable of predicting future spectrum usage of the region based on the reconstructed dataset. However, due to the loss of data, the proposed model's performance degrades slightly in some area. Even though, the proposed model can



(a) Real power matrix (b) Prediction with D_{B0} (c) Prediction with D_{B1} (d) Prediction with D_{B2} (e) Prediction with D_{B3}

Fig. 12. Multiple transmitters scenario: prediction results.

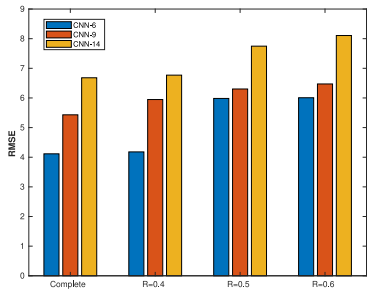


Fig. 13. CNN models comparison.

provide us with the general distribution of spectrum usage in the target region. Regardless of data loss, the prediction with complete sensor data as shown in Fig. 12 (b) shows that the proposed model maintains a good performance even under multiple transmitters scenario. Numerical evaluations based on the averaged RMSE and MAPE of the experiment's results are summarized in Table V.

E. Impact of Vanishing Gradient Problem

To show the advantages of applying ResNet to address the vanishing gradient problem, a group of CNN prediction model with different depth are tested. As shown in Fig. 14, we compared the performances of CNN with 6, 9, and 14 layers under different sparse sensor conditions, receptively. Under the same sparsity condition, the prediction error rate increases with the increase of network depth. On the other hand, the prediction error of a shallow CNN model (e.g., layer-6 CNN) increases quickly with sparser sensor observation. Therefore, the original CNN model has limited capacity to predict the spatio-temporal spectrum load with sparse data.

F. Comparison With Other Prediction Models

To further verify the performance of the proposed model, we compared the proposed NN-ResNet model with CNN (layer 6) and ConvLSTM prediction model under various sparsity conditions using the same reconstructed datasets. The ConvLSTM model has the same depth as NN-ResNet. The comparison results are shown in Table V.

The results show that the three models have relatively lower error rate using the complete dataset for prediction. In the single-transmitter scenario, NN-ResNet and ConvLSTM have similar performance while LSTM performs slightly better with RMSE 3.6883 thanks to its memory-preserving property. CNN gives RMSE of 4.115. However, the performance of the three models start to differ with sparser sensor observations, i.e., with the increase of sparsity *R*. The performances of CNN and ConvLSTM degrade quickly where the RMSE of CNN increased from 4.3951 to 6.0072 and ConvLSTM increased from 4.2368 to 5.3127. NN-ResNet on the other hand, maintains a relatively more stable prediction accuracy with the increase of *R*: the RMSE increased gradually from 4.0907 to 4.6983. The superiority of NN-ResNet is more obvious in the multiple-transmitter scenario. The three models show similar performance when trained with the complete dataset where NN-ResNet gives the best result with RMSE 4.017. When the sparsity increases, the RMSE of both CNN and ConvLSTM prediction model increase quickly from 4.5493 and 4.4434 to 6.9831 and 5.591, respectively. NN-ResNet maintains smaller error rates and increment from 4.3512 to 5.0775. In addition, we also compared the performance of ResNet without the nearest neighbor reconstruction procedure, i.e., using the incomplete dataset directly for training and prediction. As it is shown in the first row of Table V, the prediction has very high error rate in both single and multiple-transmitter scenarios and worsens with the increase of sparsity. The results show the importance of the reconstruction procedure.

VII. Conclusion and Future Work

In this paper, we focused on the spectrum load prediction both spatially and temporally. We proposed a spatio-temporal prediction model based on deep learning. In our work, CNNs and ResNets were utilized in the prediction process, since the former performs well in capturing both spatial and temporal features, while the latter is helpful in avoiding the vanishing gradient problem. The combination of CNN and ResNet guarantees stable weight updating during the back-propagation process. The model performed well in predicting an overall spectrum usage pattern spatially and temporally. It is also worth mentioning that the configurations of the parameters, such as the settings of window length, number of filters, and number of ResNet blocks, are dependent on the dataset itself. Different datasets may require different configurations to obtain the best performance of the model.

In addition, we studied the cases where a limited number of sensors are utilized for prediction to avoid high sensing cost. In order to maintain good performance, two widely used data reconstruction approaches are compared and analyzed. As a result, the nearest neighbor interpolation method outperforms the matrix completion method in both accuracy and time cost aspects. The capability of the proposed prediction

model under sparse sensor circumstances are verified through two experiments: single transmitter scenario *A* and multiple transmitters scenarios *B*. With the help of the reconstruction process, the proposed prediction model is able to maintain good performances under different levels of sensor sparsity.

However, the proposed model only focused on the short-term prediction due to the limitation of the dataset. The performance of the long-term prediction needs further verification. On the other hand, the proposed model has a similar problem as other deep learning models that the model parameters are fixed to particular scenarios dependent on the dataset fed to the model. Transfer learning is a promising tool to migrate models across different scenarios. In future work, transfer learning could be added to the proposed model. The transfer learning module tunes the existing model's parameters accordingly when the target scenario is changed.

ACKNOWLEDGMENT

The authors would like to thank Li Li, Humphrey Rutagemwa, and Maoyu Wang from CRC for the constructive comments and suggestions.

REFERENCES

[1] X. Ren, H. Mosavat-Jahromi, L. Cai, and D. Kidston, "Spatio-temporal spectrum load prediction using convolutional neural network and Bayesian estimation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[2] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[3] M. López-Benítez and F. Casadevall, "An overview of spectrum occupancy models for cognitive radio networks," in *Proc. Netw. Workshops*, Aug. 2011, pp. 32–41.

[4] V. M. Patil and S. R. Patil, "A survey on spectrum sensing algorithms for cognitive radio," in *Proc. Int. Conf. Adv. Human Mach. Interact. (HMI)*, Kodigehalli, India, Mar. 2016, pp. 1–5.

[5] A. Ranjan, Anurag, and B. Singh, "Design and analysis of spectrum sensing in cognitive radio based on energy detection," in *Proc. Int. Conf. Signal Inf. Process. (IConSIP)*, Nanded, India, Oct. 2016, pp. 1–5.

[6] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[7] A. Glandon, S. Ullah, L. Vidyaratne, M. Alam, C. Xin, and K. M. Iftekharuddin, "Prediction of spatial spectrum in cognitive radio using cellular simultaneous recurrent networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–7.

[8] Y. Zeng, V. Chandrasekaran, S. Banerjee, and D. Giustiniano, "A framework for analyzing spectrum characteristics in large spatio-temporal scales," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput.*, 2019, pp. 1–16.

[9] M. Hartmann, O. Pfadenhauer, L. Patino-Studencka, H.-M. Tröger, A. Heuberger, and J. Thielecke, "Antenna pattern optimization for a RSSI-based direction-of-arrival localization system," in *Proc. ION Pac. PNT Meeting*, 2015, pp. 429–433.

[10] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *J. Comput. Sci. Technol.*, vol. 25, no. 2, pp. 274–297, Mar. 2010.

[11] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, "Cooperative received signal strength-based sensor localization with unknown transmit powers," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1389–1403, Mar. 2013.

[12] I. Kashiwagi, T. Taga, and T. Imai, "Time-varying path-shadowing model for indoor populated environments," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 16–28, Jan. 2010.

[13] L. Liu, H. Mosavat-Jahromi, L. Cai, and D. Kidston, "Hierarchical agglomerative clustering and LSTM-based load prediction for dynamic spectrum allocation," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2021, pp. 1–6.

[14] X. Xing, T. Jing, W. Cheng, Y. Huo, and X. Cheng, "Spectrum prediction in cognitive radio networks," *IEEE Wireless Commun.*, vol. 20, no. 2, pp. 90–96, Apr. 2013.

[15] K. Deng, A. W. Moore, and M. C. Nechyba, "Learning to recognize time series: Combining ARMA models with memory-based learning," in *Proc. IEEE Symp. Comput. Intell. Robot. Autom. Towards New Comput. Principles Robot. Autom.*, Monterey, CA, USA, Jul. 1997, pp. 246–251.

[16] H. Mosavat-Jahromi, Y. Li, L. Cai, and J. Pan, "Prediction and modeling of spectrum occupancy for dynamic spectrum access systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 715–728, Sep. 2021.

[17] H. Hong, B. Pradhan, M. N. Jebur, D. T. Bui, C. Xu, and A. Akgun, "Spatial prediction of landslide hazard at the Luxi area (China) using support vector machines," *Environ. Earth Sci.*, vol. 75, no. 1, p. 40, Dec. 2015.

[18] Z. Zhang, K. Zhang, F. Gao, and S. Zhang, "Spectrum prediction and channel selection for sensing-based spectrum sharing scheme using online learning techniques," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Hong Kong, Dec. 2015, pp. 355–359.

[19] A. Sahoo, "A machine learning based scheme for dynamic spectrum access," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Nanjing, China, May 2021, pp. 1–7.

[20] 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, p. 818, Apr. 2017.

[21] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," 2017, *arXiv:1712.06080*.

[22] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *Proc. World Wide Web Conf.*, Aug. 2019, pp. 2181–2191.

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

[24] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1655–1661.

[25] S. Targ, D. Almeida, and K. Lyman, "ResNet in ResNet: Generalizing residual architectures," 2016, *arXiv:1603.08029*.

[26] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 2588–2595.

[27] B. S. Shawel, D. H. Woldegebreal, and S. Pollin, "Convolutional LSTM-based long-term spectrum prediction for dynamic spectrum access," in *Proc. Eur. Signal Proces. Conf. (EUSIPCO)*, A Coruna, Spain, Sep. 2019, pp. 1–5.

[28] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2016, pp. 550–558.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, Aug. 2017, pp. 4278–4284.

[30] D. Zhai, A. Liu, S. Chen, Z. Li, and X. Zhang, "SeqST-ResNet: A sequential spatial temporal ResNet for task prediction in spatial crowdsourcing," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Apr. 2019, pp. 260–275.

[31] W. Ruan, P. Xu, Q. Z. Sheng, N. J. Falkner, X. Li, and W. E. Zhang, "Recovering missing values from corrupted spatio-temporal sensory data via robust low-rank tensor completion," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Mar. 2017, pp. 607–622.

[32] O. Rukundo, "Evaluation of rounding functions in nearest-neighbor interpolation," 2020, *arXiv:2003.06885*.

[33] H.-E. Zeng and S.-X. Huang, "Research on spatial data interpolation based on Kriging interpolation," *Eng. Surv. Mapp.*, vol. 5, pp. 2–8, May 2007.

[34] Y. Liu, A. Sagan, A. Bernstein, R. Yang, X. Zhou, and Y. Zhang, "Matrix completion using alternating minimization for distribution system state estimation," 2019, *arXiv:1909.12459*.

[35] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," in *Proc. ACM Spec. Int. Group Data Commun. (SIGCOMM)*, Oct. 2011, pp. 267–278.

[36] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Math. Program. Comput.*, vol. 3, no. 3, p. 165, Jul. 2011.

[37] S. Boyd and L. Vandenberghe, "Advances in convex optimization: Theory, algorithms, and applications," in *Proc. IEEE Int. Symp. Inf. Theory*, 2002, pp. 1–51.

[38] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Compt. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 3147–3155.
[39] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On empirical comparisons of optimizers for deep learning," 2019, *arXiv:1910.05446*.

Xiangyu Ren (Graduate Student Member, IEEE) received the B.Sc. degree from the Department of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2019. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada.

His research interests include deterministic networks, software-defined network, vehicular networks, machine learning, and optimization with applications in networking. He was a recipient of the Graduate Student Fellowship Award from the University of Victoria in 2021.

Hamed Mosavat-Jahromi (Member, IEEE) received the B.Sc. degree from the Iran University of Science and Technology in 2012, the M.Sc. degree in electrical engineering from the University of Tehran in 2015, and the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada.

He is currently a Research Engineer with Huawei Technologies Canada Research Center, Ottawa, ON, Canada. His research interests include vehicular networks, optical networks, machine learning, and optimization with applications in networking. He was a recipient of the Transportation Electronics Fellowship Award from the IEEE Vehicular Technology Society in 2020.

Lin Cai (Fellow, IEEE) received the M.A.Sc. and Ph.D. degrees (Outstanding Achievement Award in Graduate Studies) in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical and Computer Engineering, University of Victoria, where she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things. She was a recipient of the NSERC Discovery Accelerator Supplement Grants in 2010 and 2015, respectively. She has co-founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She was elected to serve the IEEE Vehicular Technology Society Board of Governors from 2019 to 2024. She has served as an Area Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a member of the Steering Committee of the IEEE TRANSACTIONS ON BIG DATA and IEEE TRANSACTIONS ON CLOUD COMPUTING, and a Distinguished Lecturer of the IEEE VTS and ComSoc Societies. In 2020, she was elected as a member of the Royal Society of Canada's College of New Scholars, Artists, and Scientists, and a 2020 "Star in Computer Networking and Communications" by N2Women. She is an NSERC E.W.R. Steacie Memorial Fellow and an Engineering Institute of Canada Fellow.

David Kidston (Senior Member, IEEE) received the Ph.D. degree in systems and computer engineering from Carleton University, Ottawa, Canada, in 2008. Since 2008, he has been working as a Research Scientist with the Communications Research Centre Canada, a government research laboratory in Ottawa. His current research interests include network and system level design of spectrally efficient communication and monitoring systems, especially in low bandwidth, and congested environments.