

## Article

# Efficient Spectrum Occupancy Prediction Exploiting Multidimensional Correlations through Composite 2D-LSTM Models

Mehmet Ali Ayg  l <sup>1,\*</sup> , Mahmoud Nazzal <sup>1</sup> , Mehmet İzzet Saęlam <sup>2</sup>, Daniel Benevides da Costa <sup>3</sup>, Hasan Fehmi Ateş <sup>1</sup>, and H  seyin Arslan <sup>1,4</sup> 

<sup>1</sup> Department of Electrical and Electronics Engineering, Istanbul Medipol University, Istanbul, 34810, Turkey; mahmoud.nazzal@ieee.org (M.N.); hfates@medipol.edu.tr (H.F.A.)

<sup>2</sup> Department of Research & Development, Turkcell, Istanbul, 34880, Turkey; izzet.saglam@turkcell.com.tr (M.I.S.)

<sup>3</sup> Department of Computer Engineering, Federal University of Cear   (UFC), Sobral, 62010-560, Brazil; danielbcosta@ieee.org (D.B.C.)

<sup>4</sup> Department of Electrical Engineering, University of South Florida, Tampa, FL, 33620, USA; arslan.usf@gmail.com (H.A.)

\* Correspondence: mehmet.aygul@std.medipol.edu.tr; Tel.: +90-444-85-44

† This paper is an extended version of our paper published in [1].

**Abstract:** In cognitive radio systems, identifying spectrum opportunities is fundamental to efficiently use the spectrum. Spectrum occupancy prediction is a convenient way of revealing opportunities based on previous occupancies. Studies have demonstrated that usage of the spectrum has a high correlation over multidimensions which includes time, frequency, and space. Accordingly, recent literature uses tensor-based methods to exploit the multidimensional spectrum correlation. However, these methods share two main drawbacks. First, they are computationally complex. Second, they need to re-train the overall model when no information is received from any base station for any reason. Different than the existing works, this paper proposes a method for dividing the multidimensional correlation exploitation problem into a set of smaller sub-problems. This division is achieved through composite two-dimensional (2D)-long short-term memory (LSTM) models. Extensive experimental results reveal a high detection performance with more robustness and less complexity attained by the proposed method. The real-world measurements provided by one of the leading mobile network operators in Turkey validate these results.

**Keywords:** Cognitive radio, deep learning, multidimensions, real-world spectrum measurement, spectrum occupancy prediction.

## 1. Introduction

Accommodating exploding data traffic is among one of the biggest challenges for the communication systems in the fifth-generation (5G) and beyond. In 5G networks, data rates will be multiplied by ten compared to fourth-generation [2], and latency will go down to one millisecond or less [3]. Therefore, the ever-increasing demanding nature of higher rate communications causes an inherent gap with the scarcity of the available spectrum [4]. Cognitive radio (CR) is believed to be one of the key solutions to bridge this gap [5].

CR enables secondary users (SUs) to opportunistically use available spectrum bands (referred to as *spectrum holes*) unused by primary users (PUs) [6]. It is evident that this needs identifying spectrum usage states, in a process referred to as spectrum sensing. Spectrum prediction is an alternative to continuous spectrum sensing [7–10] where future occupancies are predicted based on previous occupancies. This alternative substantially saves the time, energy, and computation overheads required by continuous spectrum sensing [11].

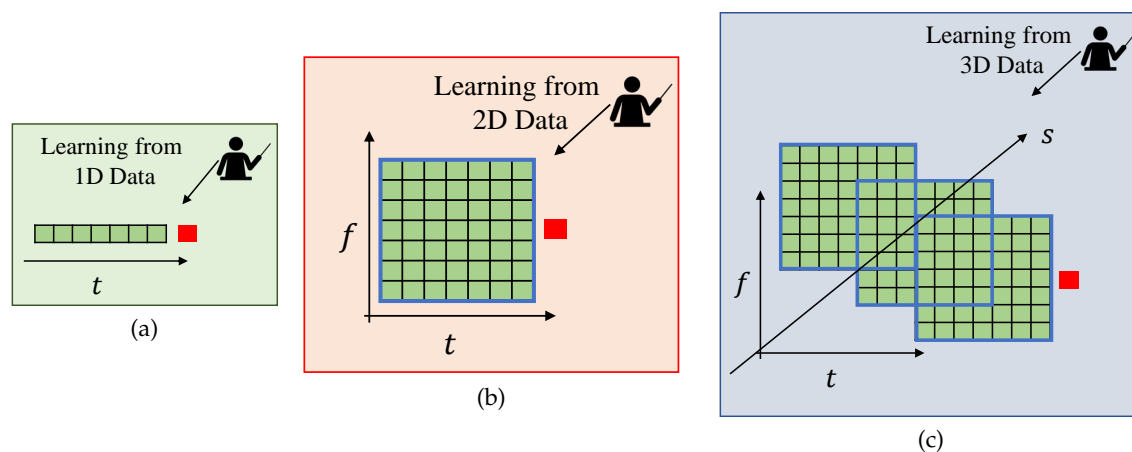
There are various spectrum occupancy prediction methods [12]. Early works used classical statistical prediction methods to predict holes. Examples include predicting holes using an exponential moving average model-based method [13], an autoregressive model (ARM)-based method [14], and a Bayesian inference (BIF)-based method [15]. More recently, machine learning (ML)-based methods have been preferred for this problem such as shallow artificial neural networks [16] and wavelet neural networks [17].

Recent literature views that spectrum occupancy is a non-stationary process [18]. However, the aforementioned methods may not always be capable of addressing this issue. This incapability has more strongly emerged with the diverse user types and increased user mobility envisaged in 5G and beyond. In view of these challenges, the non-stationary hidden Markov method exploited the time-varying feature of PU behaviors [19] and deep learning (DL) methods have been proposed as an advanced spectrum prediction framework for addressing this non-stationarity.

Advanced DL techniques, namely, convolutional neural networks (CNN) and long-short term memory (LSTM) are promising to exploit correlation with long lags. Accordingly, the phase and amplitude difference of data was used to train CNN classifiers for detecting the presence of radar signals with high accuracy [20]. Spectral and temporal correlations were used with LSTM models [21]. Moreover, the spectrum in a frequency hopping communication was predicted by an LSTM network [22]. This work was subsequently extended using the Taguchi approach [11]. Furthermore, deep neural networks, LSTM, and CNN-based models were designed. Then, their capabilities in spectrum occupancy prediction were compared [23]. In our previous work [1], time and frequency correlations were exploited to predict spectrum occupancy over real-world measurements.

Although the aforementioned approaches have been useful in the analysis of numerous cases, they consider correlations only in time, space, and/or frequency domain. However, these dimensions do not provide a detailed analysis of the non-stationary characteristics and multidimensional attributes of the wireless signals [24]. Jointly exploiting multidimensional (time, frequency, and space) correlations provides a promising perspective for spectrum prediction, as illustrated in Fig. 1. Recently, tensor analysis has been adopted as a framework to utilize multidimensional correlations for spectrum prediction. Along this line, [25] converted spectrum prediction into a third-order tensor completion problem. This approach achieved one-day-long predictions with a reasonable error margin. Another work considers merging CANDECOMP/PARAFAC tensor decomposition with LSTM for prediction [26]. Besides, multidimensional correlations were utilized jointly with convolutional long short-term memory (ConvLSTM) for a long-term temporal prediction [27].

Although tensor models provide a powerful and rich representation of a three-dimensional (3D) dataset, they share two common drawbacks. First is their high processing time [28,29]. Second, is that they assume 3D data can be provided at any time. However, sometimes, it is difficult to get



**Figure 1.** Spectrum occupancy prediction with (a) 1D data, (b) 2D data, (c) 3D data.

information from all of the base stations (BSs). For example, in the case of CR security threats (primary user emulation attack and jamming attack [30]), accurate information about spectrum occupancy cannot be provided from BSs. Also, in the case of a natural disaster, the information flow of some BSs can be cut. Therefore, such an assumption is not always valid or realistic.

To compensate for the aforementioned effects, more adaptive and flexible methods should be developed. In this regard, step-based methods [31], which divide the problem into smaller sub-problems, can be used. By the virtue of these methods, the complexity load in one model can be divided at the expense of temporal globality [32]. An attractive advantage of this setting is that it is not required to re-train the whole model once any element (BS) in the model fails to provide its information. Alternatively, with no information is temporarily removed from the model, and only a simple end-classifier, which will give the last decision about spectrum occupancy needs to be re-trained.

In view of the above discussion, we extend a preliminary version of this work that uses two-dimensional (2D)-LSTM for spectrum occupancy prediction, appearing in [1]. In this extension, we exploit multidimensional correlations and propose the usage of composite 2D-LSTM models to divide the 3D spectrum occupancy prediction problem into smaller sub-problems. Extensive experimental results demonstrate that the proposed method can predict spectrum occupancies with less complexity and small performance loss compared to tensor-based methods. Besides, the performance of the proposed method is superior to the one-dimensional (1D) and 2D-based methods. Also, the proposed method does not require complete re-training with the absence of data from any BS. These results are validated over real-world spectrum measurements provided by one of the leading mobile network operators in Turkey. These measurements are made in two scenarios; city center and village, to reflect different user density scenarios. The *precision*, *recall*, and  $F_1$ -score performance metrics are used in this validation. Moreover, the training and testing execution time is used as a computational complexity metric.

*Organization:* This paper is organized as follows. Section 2 presents the system model and preliminaries. Section 3 details the proposed method. The data generation and experiments conducted to evaluate the performance of the proposed method are presented in Section 4 and Section 5, respectively. Finally, the paper is concluded in Section 6.

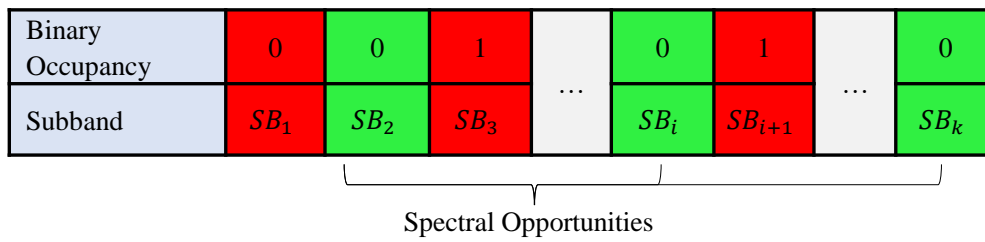
*Notation:* Scalars are represented by plain-faced letters. Vectors and matrices are denoted by bold-faced lower-case and bold-faced upper-case letters, respectively. In a vector  $\mathbf{x}$ , the symbol  $x_i$  denotes its  $i$ -th element.  $\odot$  denotes element-wise multiplication.

## 2. System Model and Preliminaries

This work aims at predicting spectrum occupancy states over a given frequency range depending on the previous occupancies. For modeling the spectrum access, the heterogeneous spectrum access model [33] is adopted. This model is demonstrated in Fig. 2. More specifically, the spectrum is split into  $k$  contiguous frequency subbands. In this model, the absence (presence) of a PU means signifies a hole (occupied spectrum). The following hypotheses ( $\mathcal{H}_0$  and  $\mathcal{H}_1$ ) formally states these cases.

$$\mathbf{y} = \begin{cases} \mathbf{n}, & \mathcal{H}_0 : \text{there is no PU} \\ \mathbf{H}\mathbf{x} + \mathbf{n}, & \mathcal{H}_1 : \text{a PU is present,} \end{cases} \quad (1)$$

where  $\mathbf{x}$  denotes PU transmitted signal and  $\mathbf{n}$  and  $\mathbf{H}$  represent the noise realization and channel matrix, respectively.



**Figure 2.** Spectrum subband (SB) occupancy modeling.

### 2.1. Prediction with Autoregressive Model

The ARM [14] is a linear predictor and it works as follows

$$\hat{x}_t = \sum_{i=1}^r \varphi_i x_{t-i} + \omega_t, \quad (2)$$

where  $r$  represent the model order,  $\varphi_i, i = 1, 2, \dots, r$ , is the model parameter,  $\hat{x}_t$  and  $\omega_t$  denote predicted state at a future time instant  $t$  and white noise at time  $t$ , respectively, and  $x_{t-i}$  is the observation at time instant  $t - i$ .

Predicting future states requires tuning the ARM parameters first. This can be achieved with several methods such as Yule-Walker equations or maximum likelihood estimation. Once these parameters are tuned, they are used along with the historical values to predict the future states  $\hat{X}_t$ .

### 2.2. Prediction with Bayesian-Inference

BIF is a prediction method that uses the Bayes rule to update the probability distribution of a hypothesis when additional evidence data is learned. In this prediction method, a posterior probability distribution  $P(\theta|X)$  is derived by a CR user according to a Bayes rule as follows

$$P(\theta|X) = \frac{P(X|\theta) \cdot P(\theta)}{P(X)}, \quad (3)$$

where  $X = \{x_1, x_2, \dots, x_n\}$  denotes the observed data. Afterward, the upcoming data is predicted by the derived posterior probability with the Bayes rule.

### 2.3. Prediction with Long Short-Term Memory

A more accurate data representation can be acquired by the use of multiple hidden layers in the deep architectures of DL models. The usage of multiple layers enables the model to magnify the intrinsic distinctive data features while suppressing the irrelevant information at each layer [34]. Thus, a primary advantage of DL is that it works directly on raw data. This means alleviating the human effort needed in any feature crafting/engineering. LSTM is an artificial recurrent neural network and it can be used as a DL model. This DL model is well-suited for handling grid-like data either in one, two, or multiple dimensions.

LSTM models have a memory block which includes cells and gates. This memory block makes the model capable to use long short-term dependencies [35]. The gates are categorized under three

parts according to their practical functionalities. These are input gates  $i_t$ , forget gates  $f_t$ , and output gates  $o_t$ . Their transition equations are shown as follows

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i), \quad (4)$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f), \quad (5)$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o), \quad (6)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1}, \quad (7)$$

$$h_t = o_t \odot \tanh(c_t), \quad (8)$$

where  $x_t$  represents input vector at time  $t$ ,  $h_t$  represents hidden layer at  $t$  time step,  $c_t$  represents a memory cell,  $\sigma$  denotes the logistic sigmoid function, and biases, weights and recurrent weights are denoted by  $b$ ,  $U$ ,  $W$ , respectively.

The above equations show that an input gate controls how a new value flows into the memory. On the other hand, how much of the past information to keep in the memory is decided by a forget gate and an output gate determines the weighting of the values to be used for computing the output activation of the block. Finally, the model can learn how to represent information over multiple time scales since the values of the gating variables vary for each vector element [36].

#### 2.4. Prediction with Convolutional Long Short-Term Memory

ConvLSTM is a type of LSTM cell. More specifically, convolution takes place within the LSTM cell, and matrix multiplication is replaced with the convolution operation. The application of convolution allows capturing the spatial features from the image or data grid. The convolutional structures can be observed in ConvLSTM in both the input-to-state transition and state-to-state transition. It has been applied to activation from previous timestamps and input of the current timestamp. Further details can be found in [37].

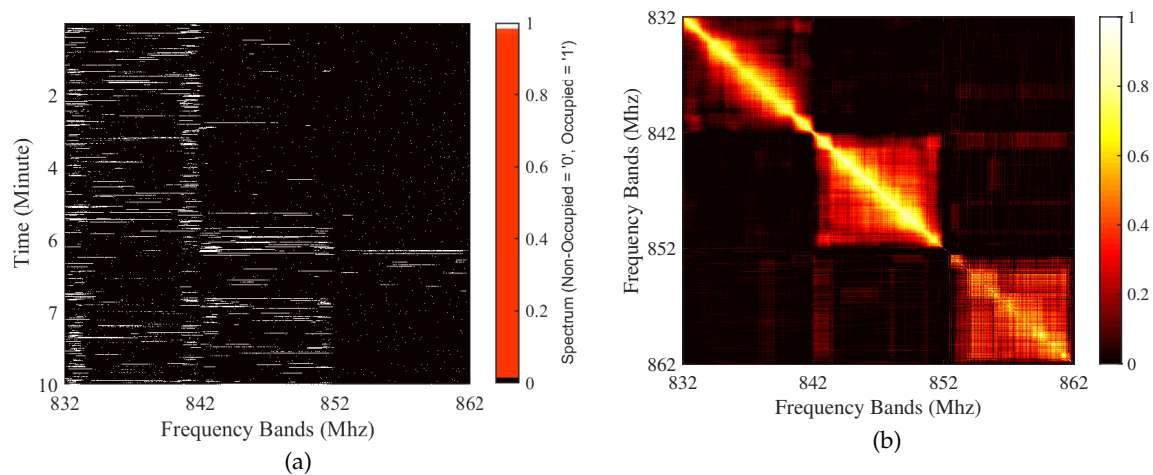
### 3. The Proposed Method for Spectrum Occupancy Prediction Exploiting Time, Frequency, and Space Correlations

#### 3.1. Motivations for Time, Frequency, and Space Correlation Exploitation and Problem Sub-Division

Time series prediction depends on correlation over time. Similarly, the inherently existing frequency and space correlation can be exploited. The following test quantifies this correlation across a time-frequency grid. First, a day-long record of spectrum measurements has been obtained according to the measurement setup detailed in [1]. Figure 3 (a) represents spectrum occupancy distributions for all frequency bands and time instants. Besides, the spectrum occupancy correlations were calculated for each frequency point. These results are plotted in Fig. 3 (b). We here note that in these figures, each 10 MHz band belongs to a different operator.

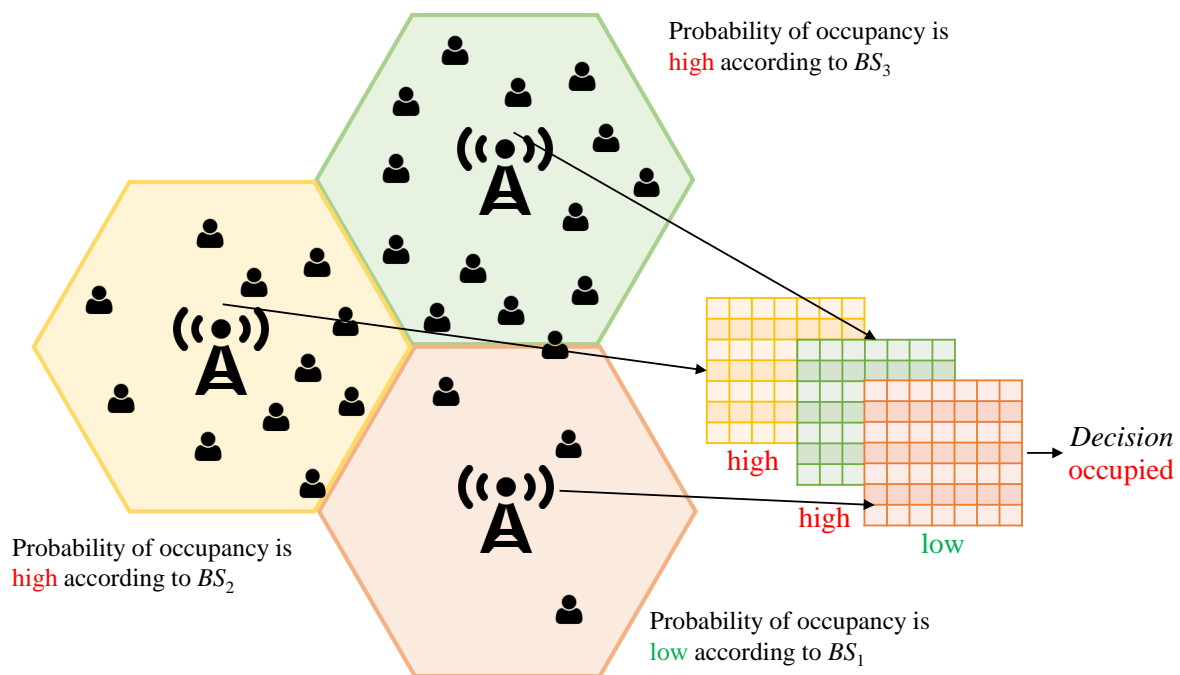
The correlations in both time and frequency are shown in Fig. 3 (a) where the color bar shows the state of spectrum occupancy. More specifically, the vertical (horizontal) lines represent the correlation in time (frequency). Correspondingly, the correlation coefficient is demonstrated by the block-correlation pattern in Fig. 3 (b). As seen in the figure, this correlation is high in the neighboring frequency bands of each operator. Therefore, the existence of strong correlations across both time and frequency can be observed for each operator individually.

In addition to the time and frequency correlations, the exploitation of spatial correlation is advantageous for the spectrum occupancy prediction problem, as illustrated in Fig. 4. It can be seen from this figure that when  $BS_1$  (main BS) is trained without any prior information from  $BS_2$  and  $BS_3$  (neighboring BSs), it decides on the future spectrum as non-occupied. However,  $BS_2$  and  $BS_3$  are full, and the user of  $BS_2$  and  $BS_3$  can occupy the spectrum of the  $BS_1$  for the upcoming intervals, so the



**Figure 3.** An example spectrum's (a) occupancy distributions and (b) correlation.

number of occupied bands will increase for  $BS_1$ . Exploiting the information from this dimension, spectrum occupancy can be predicted more reliably.



**Figure 4.** An illustrative motivation for space correlation.

Motivated by the aforementioned existence of a multidimensional correlation, current literature uses tensor-based methods for spectrum occupancy prediction. However, these methods exhibit high degrees of computational complexity. Alternatively, a *divide-and-conquer* approach can be used. This decentralized approach divides the original problem into smaller sub-problems and solves each one individually. Then, it integrates the solutions to obtain an eventual holistic solution. The following experiment is conducted to investigate the validity of this idea. A dataset is collected according to the measurement setup for the Taksim area as detailed in Section 4. Then, the networks are trained with 2D and 3D datasets. We here note that hyperparameters of these networks are detailed in Section 5. Finally, complexity comparison is made in terms of training and testing execution times. The results are listed in Table 1. This table shows that the computational complexity of the 3D-based method is much higher than the 2D-based method.



**Table 1.** Training and testing execution time (seconds) comparison of the tensor and 2D-based method.

Method	Execution Time (s)	
	Training	Testing
The tensor-based method	608	2.8
2D-LSTM-based method	57.8	0.7

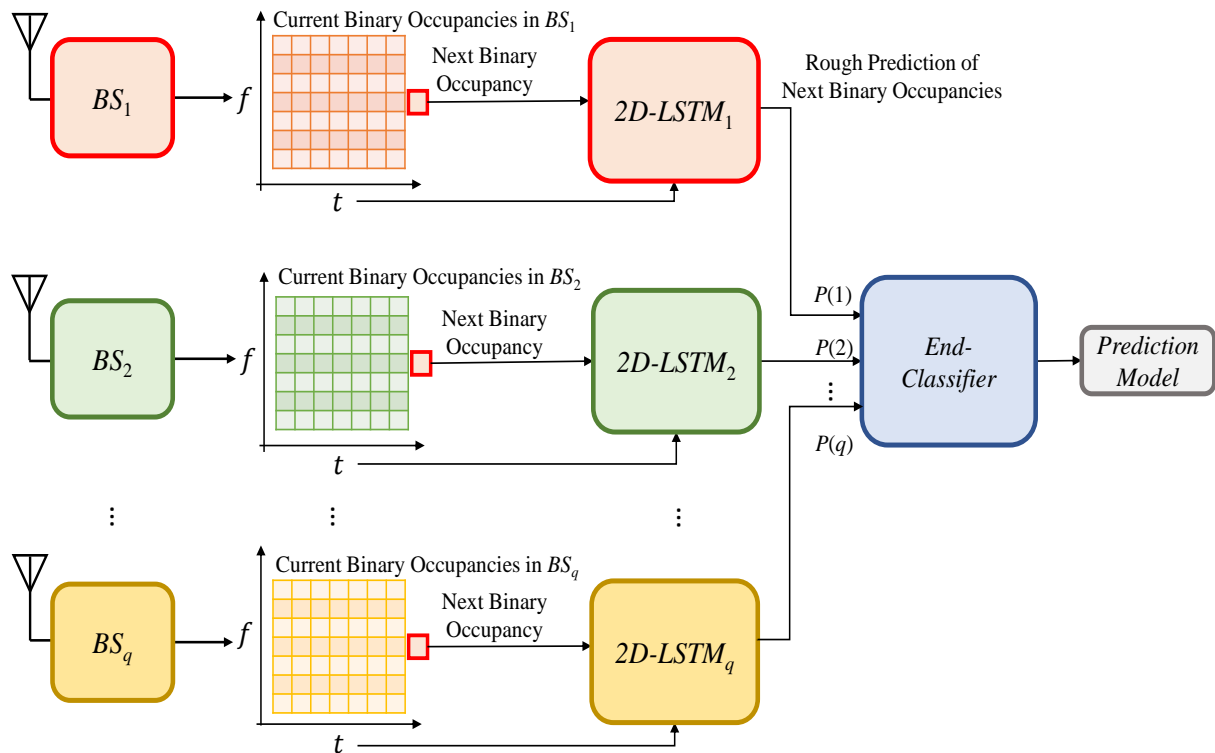
### 3.2. The Proposed Method

The proposed method uses a learning-based strategy. This strategy composites of two stages are referred to as training and testing. In the training stage, the dataset is collected and DL models are configured and trained. Then, in the testing stage, the spectrum occupancy prediction is performed. These stages are detailed below.

In the training stage, a set of spectrum measurements for training are collected by several BSs. Then, the binary occupancies of these measurements are acquired with the comparison of the measured received signal strength indicator (RSSI) with a specific threshold. After collecting the measurements, the dataset is obtained as follows. A sample of input is formed by a sliding window that sweeps the time and frequency<sup>1</sup>. Moreover, the output data of this input is the RSSI value in the medial frequency value for the upcoming time interval, which is to be predicted. According to this setting, RSSI measurements are stored in a 2D shape referring to time and frequency as an input dataset for each BS. Besides, their corresponding occupancies of the main BS are stored as an output dataset.

Once training data is obtained, the DL model is configured to have a total number of  $q$  models, where  $q$  denotes the number of spatially-correlated BSs, and each model accepting as input a

<sup>1</sup> For instance, if the length of the sliding window is set to 7, then each input will have a  $7 \times 7$  matrix of binary occupancies. This matrix will sweep time and frequency axes.

**Figure 5.** The proposed method-training stage.

time-frequency occupancy grid. As mentioned earlier, this configuration simultaneously incorporates space correlation and time-frequency correlations. More specifically, each LSTM model is trained over a set of pairs of a given time-frequency occupancy grid and the corresponding ground-truth occupancy status of the main BS. Then, each DL model calculates a probability of occupancy ( $P$ ). The occupancy probabilities of all  $q$  models are augmented to form an occupancy probability feature vector. Afterward, the occupancy probability vector, along with the true occupancy state form the training data pair of the end-classifier to incorporate space correlation. Finally, the end-classifier is trained over this data pair. These processes are presented in Fig. 5.

The run-time operation of the proposed method is represented by the testing stage. In this stage, spectrum measurements for time, frequency, and space lags are used to predict the corresponding spectrum occupancy probability in the upcoming time instant. This is achieved by feeding the binary occupancies of the 2D dataset as a grid to the DL models already trained, which generate the occupancy as numeric values (probabilities). Then, analogous to the setting in the training stage, the predicted  $q$  grid probabilities are augmented in the shape of an occupancy probability vector. Finally, the predicted occupancy probability vector is fed to the end-classifier to yield the eventual occupancy prediction. Figure 6 illustrates these processes.

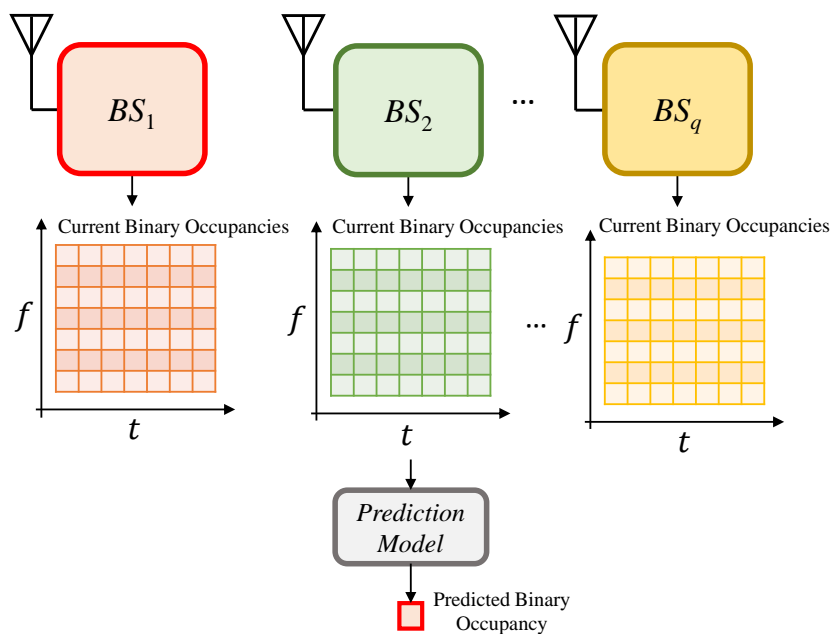


Figure 6. The proposed method-testing stage.

### 3.3. A Note on Computational Complexity

The computational complexity of the proposed method can be roughly quantified in terms of the required execution time for training and testing stages. For the training stage, the time computational complexity can be approximated as

$$T(n) = \mathcal{O}_l + \mathcal{O}_s, \quad (9)$$

where  $\mathcal{O}_l$  and  $\mathcal{O}_s$  represent time complexity of a 2D-LSTM model and an end-classifier, respectively. It is noted that although composite LSTM models ( $q$  number of LSTM models) are used in the proposed method, their total execution times equals an LSTM model execution time since they are independent and can work in parallel.

The total number of parameters,  $p_l$ , in a standard LSTM network, including one cell in each memory block, neglecting the biases, is as follows

$$p_l = 4n_c n_c + 4n_i n_c + n_c n_o + 3n_c, \quad (10)$$



where  $n_i$ ,  $n_o$ , and  $n_c$  represent the number of input units, output units, and memory cells, respectively.

The computational complexity of an LSTM models per weight is  $\mathcal{O}(1)$  in training stage [38]. Thus, training computational complexity per time step is  $\mathcal{O}(p_l)$ , and the total complexity of the LSTM models is  $\mathcal{O}(p_l)$ .

As an end-classifier, let us consider a two-layer neural network. The number of parameters for this classifier,  $p_s$ , is  $lk + ml$ , where  $k$ ,  $l$ , and  $m$  denotes the number of neurons at the input, hidden, and output layers, respectively. The computational complexity of an end-classifier ( $\mathcal{O}_s$ ) can be negligible since the number of parameters is very small compared to LSTM. Consequently, the overall complexity of the training stage is  $\mathcal{O}(qp_l)$ . Besides, the computational complexity of training per sample is approximately two times more as compared to the complexity of testing per sample [39].

#### 4. Dataset Generation

##### 4.1. Measurement Setup

The measurements of the uplink private frequency band belong to one of the leading mobile network operators in Turkey were collected by directly measuring the RSSI of the spectrum as a function of time, frequency, and space. Afterward, the binary occupancies were obtained by thresholding the RSSI measurements, where “0” represents hole, and “1” represents the occupied spectrum. The measurement setups used for Taksim and Silivri datasets are shown in Figs. 7 (a) and (b), respectively. The BSs measure noise and interference power on the traffic channels (physical uplink shared channel). The minimum measurable value by the BSs is  $-121$  dBm [40]. On the other hand, a margin of 3 dB is considered to account for any variations and unforeseen effects [41,42]. Therefore,  $-118$  dBm is set as a threshold in the overall paper. In a certain frequency, if the measured RSSI is below this adopted threshold, then this band is considered as occupied and if it is above the threshold, then this band is considered as a hole.



**Figure 7.** Pictures of the spectrum measurement setups for (a) Taksim and (b) Silivri.

The measurement is conducted on a physical resource block (PRB) basis. This means that each measurement addresses 180 kHz. The accumulated noise and interference power of each PRB is averaged during the recording period. The time resolution of the recording period is 15 minutes. The Comba ODI-065R17M18JJJ-G receiving antenna [43] measures received signals between 852-862 MHz, where the frequency-division duplex is considered as an operational mode. We here note that two neighboring BSs are selected to exploit space correlations. While choosing these neighboring BSs, the number of handover attempts in a one-week period prior to the time of measurement was taken into consideration<sup>2</sup>. The numbers of handover attempts of two BSs where the most handover attempts are made with the main BSs are given in Table 2.

**Table 2.** The number of handover attempts between main BS and neighboring BSs in one week.

BS	Area	
	Taksim	Silivri
$BS_2$	6429	3178
$BS_3$	4442	3138

#### 4.2. Measurement Procedure and Geographical Locations

The measurements were taken in two regions in Istanbul-Turkey with varying levels of traffic, to investigate the generalizability of the proposed method. These areas are detailed further below.

##### 4.2.1. City Center (Taksim)

For the measurements with high user activity, data were collected simultaneously for the whole spectrum at Taksim Square, Istanbul. The measurements were started at midnight local Istanbul time (GMT+3) on August 07, 2020, and ended at midnight on August 14, 2020. It is worth mentioning that Taksim has one of the most crowded streets in the city, with contiguous buildings around. Taksim is a

<sup>2</sup> The BSs that have made the highest number of handover attempts with the main BSs between July 31, 2020, and August 07, 2020, were selected as neighboring BSs.



**Figure 8.** Picture of the environment for Taksim.

flat street within a high altitude overlooking high-rise areas in general. Also, it has a central metro station, and it is an important point for transportation. That makes the area very crowded around commute times and rush hours. Furthermore, there are many dining places which make it crowded also at lunch and dinner times. Figure 8 shows a picture of the environment for the Taksim area, which has been taken from the corresponding BS.

#### 4.2.2. Rural Area (Silivri)

For the measurements with low user activity, data were collected at Silivri where it is located in the rural parts of Istanbul. The measurements were made during the same time as the measurements taken at Taksim. The area is not as crowded as the city center. It has slightly wavy hills with a height of about 60 meters at most. Figure 9 shows a picture of the environment for the Silivri area that has been taken from the corresponding BS.



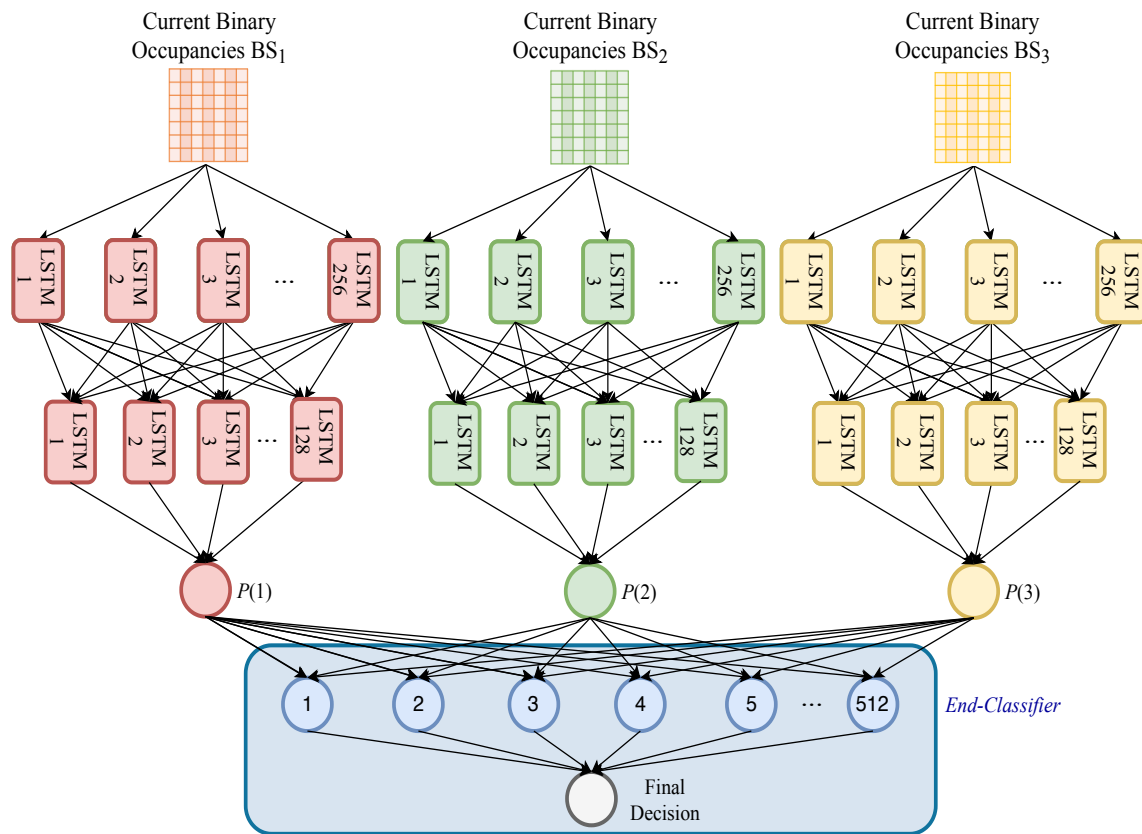
**Figure 9.** Picture of the environment for Silivri.

## 5. Parameter Settings and Experimental Results

The experimental setup used is given in Section 4. 180 kHz frequency resolution of the frequency bands within 852-862 MHz was used for the dataset. We here note that 20 kHz was used as a guard band between the subbands [44]. The frequency bands were measured every fifteen minutes for one week. Therefore, 6720000 points were measured in total. The binary occupancies were assigned for the related frequencies according to the threshold. A lag value was empirically set to 7 according to the performance and generalization capability of the prediction models. The first five-day dataset is used for training, the next day dataset for validation, and the remaining for testing in all of the experiments. In other words, approximately the first 71.5% of the dataset was used for training, the next 14.25% for validation, and the remaining 14.25% for testing in all of the experiments.

### 5.1. Hyperparameters of Deep Learning Models

All of the DL models for spectrum occupancy predictions were implemented by Keras [45], an open-source ML library under the Python environment. All of the models were trained and tested on an MSI computer with Intel® Core™ i7-7700HQ central processing unit (CPU) @ 2.80 GHz CPU, 16 GB RAM, GeForce GTX 1050 Ti graphical processing unit, and Windows 10 operating system. All parameters of the DL models were empirically set with the consideration of the performance and generalization capability of these models. DL models were individually trained for Taksim and Silivri



**Figure 10.** Composite 2D-LSTMs in spectrum occupancy prediction.

datasets. However, the same hyperparameters were used for both since the effect of hyperparameters were negligible for scenarios. Three types of DL models; 1D-LSTM, ConvLSTM, and 2D-LSTM models, and an end-classifier are briefly described below. We note that three 2D-LSTM models (composite 2D-LSTM models) and an end-classifier are used in the proposed method. The overall model that is used for the proposed method is illustrated in Fig. 10.

- **1D-LSTM:** This model uses two LSTM hidden layers and an output layer. Particularly, 256 and 128 hidden units are used in the first and second hidden layers, respectively. The rectified linear unit (ReLU) are used as activation functions. Afterward, the probability of the occupancies is calculated in the output layer which uses a sigmoid activation function with 1 unit. We here note that one unit is enough to represent the occupancies since there are only two classes (spectrum is occupied “1” or not “0”). In total, 461441 parameters are used. Finally, the model is trained with a batch size of 256 and 18 epochs. Efficient adaptive moment estimation (ADAM) is used with an optimum learning rate of 0.0001 during the training. Also, the logarithmic loss function is used for binary classification.
- **ConvLSTM:** The ConvLSTM model is used with 3D data as the state-of-the-art method. The model includes two ConvLSTM layers, a flatten layer, and an output layer. In the first and second ConvLSTM layers, 256 and 128 units are used, respectively. Afterward, a flatten layer is used to prepare a vector for the output layer. Finally, an output layer is used with one unit. In the output layer, the sigmoid function is used. In total, 4142721 parameters are used. A batch size of 256 and 15 epochs are used to train the model. ADAM is used for adaptive learning rate optimization with an optimum learning rate of 0.00005. Besides, the logarithmic loss function is used for binary classification.
- **2D-LSTM:** This model uses two LSTM hidden layers and an output layer. More specifically, with ReLU activation functions 256 and 128 hidden units are used in the first and second LSTM hidden layers, respectively. Afterward, an output layer is used to calculate the probability of the occupancy. The sigmoid function is used in the output layer. In total, 467585 parameters are used. Finally, the



DL model is trained with a batch size of 256 and 15 epochs. ADAM is used for adaptive learning rate optimization and the optimum learning rate in this model is found at 0.00005. Again, for binary classification, the logarithmic loss function is employed.

- *An end-classifier*: A standard two-layer feed-forward network [46] is used as an end-classifier. This classifier consists of a hidden layer and an output layer. The sigmoid functions are used as activation functions. The MATLAB Neural-Network-Toolbox “nprtool” [46] is used for implementation. Scaled conjugate gradient (trainsscg), and cross-entropy (crossentropy) are used for training and performance metrics, respectively. The number of hidden neurons is set to 512 while the number of output neurons is set to one. Therefore, 25600 parameters are used in total.

## 5.2. Performance Evaluation and Discussion

Six spectrum occupancy prediction methods are compared. These prediction methods are ARM, BIF, 1D-LSTM using only time correlation, 2D-LSTM using time and frequency correlations, ConvLSTM using multidimensions as a tensor, and the proposed method using multidimensions.

The performance of a classifier model can be evaluated in terms of *precision* ( $\pi$ ), *recall* ( $\psi$ ), and  $F_1$ -score performance metrics. The quantification of the percentage of positive results that are actually positive is measured by the *precision* metric, the percentage information of true positives that are identified correctly as positive are quantified by the *recall*, and the overall measures for the accuracy of classifier models are given by the  $F_1$ -score since it gives the harmonic average of *precision* and *recall*. These metrics are defined as follows

$$\pi = \frac{\xi}{\xi + v}, \psi = \frac{\xi}{\xi + \mu}, F_1\text{-score} = 2 \times \frac{\pi \times \psi}{\pi + \psi}, \quad (11)$$

where  $\xi$ ,  $v$ , and  $\mu$  represent the numbers of true positive, false positive, and false negative, respectively.

The results can be seen in Tables 3 and 4. These tables show that the proposed method is superior to ARM, BIF, 1D-LSTM, and 2D-LSTM-based methods, and its performance is close to the ConvLSTM method. This is consistently true in terms of all quality metrics. Besides, the general performance (performance of all methods) is better in the Taksim scenario since most of the bands are occupied and so the algorithm has a stronger tendency to predict the bands as occupied.

**Table 3.** The *precision* ( $\pi$ ), *recall* ( $\psi$ ), and  $F_1$ -score spectrum occupancy performances of ARM, BIF, 1D-LSTM, 2D-LSTM, ConvLSTM, and composite 2D-LSTMs (the proposed method) for Taksim scenario.

Method	Measure		
	$\pi$	$\psi$	$F_1$ -score
ARM	0.9210	0.9681	0.9440
BIF	0.9264	0.9738	0.9495
1D-LSTM	0.9602	0.9780	0.9690
2D-LSTM	0.9720	0.9732	0.9726
ConvLSTM	0.9760	0.9763	0.9762
Composite 2D-LSTMs	0.9727	0.9742	0.9735

The complexity performance analysis based on the execution times of the training and testing stages is presented in Tables 5 and 6. We here note that the value of testing complexity is given for the total computation complexity of all of the testing samples. It is evident that the proposed method has smaller execution times and faster convergence in both training and testing stages.

The generalization capability of an ML model is an important success criterion; from an ML perspective, a trained model should not memorize the training samples. The quality of a trained ML classifier model is shown in Fig. 11 as in the training and testing losses and accuracies versus epochs for the spectrum occupancy prediction when the Taksim dataset is used with the 2D-LSTM. The figure

**Table 4.** The *precision* ( $\pi$ ), *recall* ( $\psi$ ), and  $F_1$ -score spectrum occupancy performances of ARM, BIF, 1D-LSTM, 2D-LSTM, ConvLSTM, and composite 2D-LSTMs (the proposed method) for Silivri scenario.

Method	Measure		
	$\pi$	$\psi$	$F_1$ -score
ARM	0.8863	0.8704	0.8783
BIF	0.9336	0.9130	0.9232
1D-LSTM	0.9465	0.9165	0.9312
2D-LSTM	0.9462	0.9216	0.9338
ConvLSTM	0.9479	0.9298	0.9388
Composite 2D-LSTMs	0.9476	0.9233	0.9353

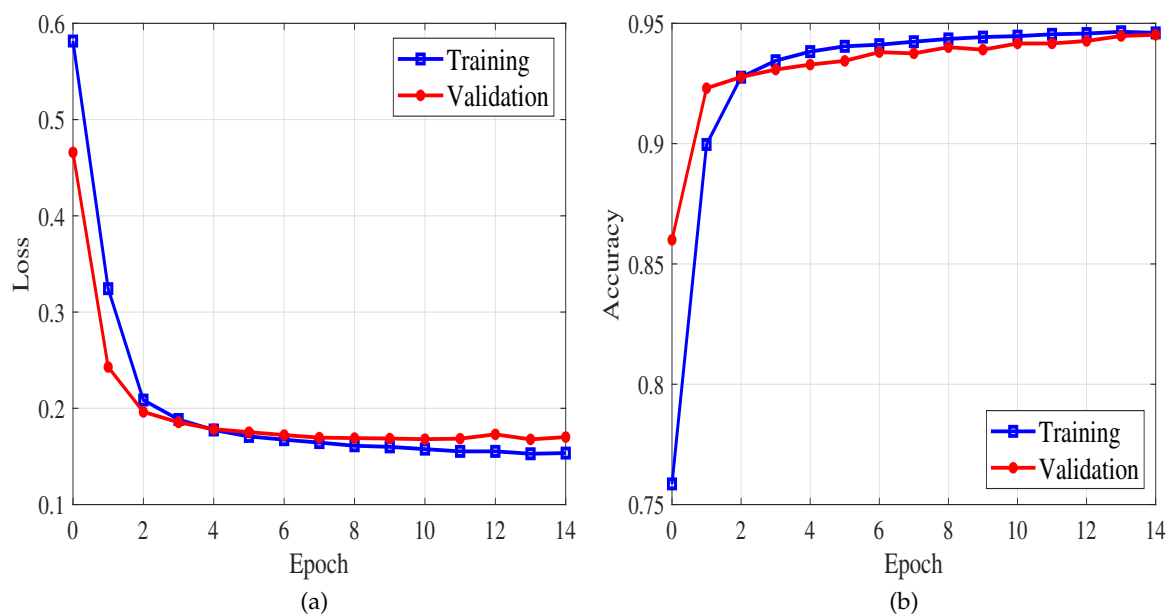
**Table 5.** Complexity analyses of ConvLSTM and the proposed method for Taksim scenario.

Method	Execution Time (s)	
	Training	Testing
The tensor-based method	608.7	2.7
Composite 2D-LSTMs	58.1	0.7

**Table 6.** Complexity analyses of ConvLSTM and the proposed method for Silivri scenario.

Method	Execution Time (s)	
	Training	Testing
The tensor-based method	610.3	2.9
Composite 2D-LSTMs	58.9	0.8

clearly shows that the accuracy of both training and test sets converge to similar values. These results validate the generalizability of the proposed model, as there is no overfitting observed. It is worth mentioning that loss and accuracy graphs are provided only for one 2D-LSTM to prevent repetition. Similar behavior is observed for the graphs of other models and end-classifiers.



**Figure 11.** (a) Loss and (b) accuracy graphs for 2D-LSTM.



6. Conclusions

This paper has demonstrated the advantage of exploiting occupancy correlation over time, frequency, and space, for spectrum occupancy prediction. 3D exploitation was achieved using composite 2D-LSTM models that incorporate previous time and frequency spectrum measurements to predict the following spectrum occupancy. It has been shown that the accuracy of the 2D model can be improved by using the 3D model. Also, the proposed method provides additional accuracy over the 2D model without incurring a substantial increase in complexity. Besides, only a simple re-training for the end-classifier is required when data is not completely available in the proposed method. Extensive experimental results revealed performance improvements over classical prediction methods and DL models that use time and joint time/frequency correlations. On the other hand, the performance of the proposed method is close to the tensor-based method, but at a less computational cost. These results were validated in terms of the *precision*, *recall*, and  $F_1$ -score performance metrics over real-world spectrum measurements. Besides, we conducted complexity analyses in terms of training and testing execution times. The reduction in computational complexity that is achieved by the proposed method is demonstrated by these analyses. Moreover, this paper has also shown the practicality of proposed DL-based and other well known spectral occupancy prediction methods when applied to real-world measurements.

**Author Contributions:** Mehmet Ali Aygöl and Mahmoud Nazzal performed computer-based simulations, wrote the paper, and developed the system model. Mehmet İzzet Sağlam provided the datasets. The development of the manuscript was supervised by Daniel Benevides da Costa, Hasan Fehmi Ateş, and Hüseyin Arslan. All of the authors have read and approved the contents of this manuscript.

**Acknowledgments:** This work was supported by the Scientific and Technological Research Council of Turkey under Grant No. 5200030.

**Conflicts of Interest:** The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
5G	fifth-generation
ADAM	adaptive moment estimation
ARM	autoregressive model
BIF	Bayesian inference
CNN	convolutional neural networks
ConvLSTM	convolutional long short-term memory
CPU	central processing unit
CR	cognitive radio
DL	deep learning
LSTM	long-short term memory
ML	machine learning
PRB	physical resource block basis
PUs	primary users
ReLU	rectified linear unit
RSSI	received signal strength indicator
SA	spectrum analyzer
SB	subband
SUs	secondary users

## References

1. Aygül, M.A.; Nazzal, M.; Ekti, A.R.; Görçin, A.; da Costa, D.B.; Ateş, H.F.; Arslan, H. Spectrum occupancy prediction exploiting time and frequency correlations through 2D-LSTM. In Proceedings of the IEEE 91st Vehicular Technology Conference (VTC2020-Spring). IEEE, 2020, pp. 1–5. doi:10.1109/VTC2020-Spring48590.2020.9129001.
2. Ancans, G.; Bobrovs, V.; Ancans, A.; Kalibatiene, D. Spectrum considerations for 5G mobile communication systems. *Procedia Comput. Sci.* **2017**, *104*, 509–516. doi:10.1016/j.procs.2017.01.166.
3. Dahlman, E.; Mildh, G.; Parkvall, S.; Peisa, J.; Sachs, J.; Selén, Y.; Sköld, J. 5G wireless access: requirements and realization. *IEEE Commun. Mag.* **2014**, *52*, 42–47. doi:10.1109/MCOM.2014.6979985.
4. Amjad, M.; Rehmani, M.H.; Mao, S. Wireless multimedia cognitive radio networks: A comprehensive survey. *IEEE Commun. Surv. Tuts.* **2018**, *20*, 1056–1103. doi:10.1109/comst.2018.2794358.
5. Hong, X.; Wang, J.; Wang, C.X.; Shi, J. Cognitive radio in 5G: A perspective on energy-spectral efficiency trade-off. *IEEE Commun. Mag.* **2014**, *52*, 46–53. doi:10.1109/MCOM.2014.6852082.
6. Haykin, S. Cognitive radio: brain-empowered wireless communications. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 201–220. doi:10.1109/JSAC.2004.839380.
7. Yucek, T.; Arslan, H. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Commun. Surv. Tuts.* **2009**, *11*, 116–130. doi:10.1109/SURV.2009.090109.
8. Banavathu, N.R.; Khan, M.Z.A. Optimization of  $N$ -out-of- $K$  rule for heterogeneous cognitive radio networks. *IEEE Signal Process. Lett.* **2019**, *26*, 445–449. doi:10.1109/LSP.2019.2893999.
9. Arienzo, L.; Tarchi, D. Statistical modeling of spectrum sensing energy in multi-hop cognitive radio networks. *IEEE Signal Process. Lett.* **2015**, *22*, 356–360. doi:10.1109/LSP.2014.2360234.
10. Arjoune, Y.; Kaabouch, N. A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions. *Sensors* **2019**, *19*, 126. doi:10.3390/s19010126.
11. Yu, L.; Chen, J.; Ding, G.; Tu, Y.; Yang, J.; Sun, J. Spectrum prediction based on Taguchi method in deep learning with long short-term memory. *IEEE Access* **2018**, *6*, 45923–45933. doi:10.1109/ACCESS.2018.2864222.
12. Naikwadi, M.H.; Patil, K.P. A Survey of Artificial Neural Network based Spectrum Inference for Occupancy Prediction in Cognitive Radio Networks. 4th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2020, pp. 903–908. doi:10.1109/icoei48184.2020.9143053.
13. Lin, Z.; Jiang, X.; Huang, L.; Yao, Y. A energy prediction based spectrum sensing approach for cognitive radio networks. In Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WICOM). IEEE, 2009, pp. 1–4. doi:10.1109/wicom.2009.5302514.
14. Wen, Z.; Luo, T.; Xiang, W.; Majhi, S.; Ma, Y. Autoregressive spectrum hole prediction model for cognitive radio systems. In Proceedings of the IEEE International Conference on Communications Workshops (ICC), 2008, pp. 154–157. doi:10.1109/ICCW.2008.34.
15. Xing, X.; Jing, T.; Huo, Y.; Li, H.; Cheng, X. Channel quality prediction based on Bayesian inference in cognitive radio networks. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2013, pp. 1465–1473. doi:10.1109/INFCOM.2013.6566941.
16. Tumuluru, V.K.; Wang, P.; Niyato, D. Channel status prediction for cognitive radio networks. *Wireless Commun. Mob. Comput.* **2012**, *12*, 862–874. doi:10.1002/wcm.1017.
17. Eltholth, A.A. Spectrum prediction in cognitive radio systems using a wavelet neural network. In Proceedings of the International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2016, pp. 1–6. doi:10.1109/SOFTCOM.2016.7772181.
18. Ding, G.; Jiao, Y.; Wang, J.; Zou, Y.; Wu, Q.; Yao, Y.D.; Hanzo, L. Spectrum inference in cognitive radio networks: Algorithms and applications. *IEEE Commun. Surv. Tuts.* **2017**, *20*, 150–182.
19. Chen, X.; Zhang, H.; MacKenzie, A.B.; Matinmikko, M. Predicting spectrum occupancies using a non-stationary hidden Markov model. *IEEE Wireless Commun. Lett.* **2014**, *3*, 333–336. doi:10.1109/lwc.2014.2315040.
20. Selim, A.; Paisana, F.; Arokiam, J.A.; Zhang, Y.; Doyle, L.; DaSilva, L.A. Spectrum monitoring for radar bands using deep convolutional neural networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM). IEEE, 2017, pp. 1–6. doi:10.1109/glocom.2017.8254105.

21. Shawel, B.; Woledgebre, D.; Pollin, S. Deep-learning based cooperative spectrum prediction for cognitive networks. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2018, pp. 133–137. doi:10.1109/ICTC.2018.8539570.
22. Yu, L.; Chen, J.; Ding, G. Spectrum prediction via long short term memory. In Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC). IEEE, 2017, pp. 643–647. doi:10.1109/compcomm.2017.8322623.
23. Omotere, O.; Fuller, J.; Qian, L.; Han, Z. Spectrum occupancy prediction in coexisting wireless systems using deep learning. In Proceedings of the IEEE 88th Vehicular Technology Conference (VTC-Fall). IEEE, 2018, pp. 1–7. doi:10.1109/VTCFall.2018.8690575.
24. Hisham, A.M.; Arslan, H. Multidimensional signal analysis and measurements for cognitive radio systems. 2008 IEEE Radio and Wireless Symposium. IEEE, 2008, pp. 639–642. doi:10.1109/rws.2008.4463573.
25. Sun, J.; Wang, J.; Ding, G.; Shen, L.; Yang, J.; Wu, Q.; Yu, L. Long-term spectrum state prediction: An image inference perspective. *IEEE Access* **2018**, *6*, 43489–43498. doi:10.1109/access.2018.2861798.
26. Alkhouri, I.; Joneidi, M.; Hejazi, F.; Rahnavard, N. Large-scale spectrum occupancy learning via tensor decomposition and LSTM networks. In Proceedings of the IEEE International Radar Conference (RADAR). IEEE, 2020, pp. 677–682. doi:10.1109/radar42522.2020.9114785.
27. Shawel, B.S.; Woldegebreal, D.H.; Pollin, S. Convolutional LSTM-based long-term spectrum prediction for dynamic spectrum access. In Proceedings of the 27th European Signal Processing Conference (EUSIPCO). IEEE, 2019, pp. 1–5. doi:10.23919/eusipco.2019.8902956.
28. Ioannidou, A.; Chatzilari, E.; Nikolopoulos, S.; Kompatsiaris, I. Deep learning advances in computer vision with 3D data: A survey. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–38. doi:10.1145/3042064.
29. Gezawa, A.S.; Zhang, Y.; Wang, Q.; Yunqi, L. A review on deep learning approaches for 3D data representations in retrieval and classifications. *IEEE Access* **2020**, *8*, 57566–57593. doi:10.1109/access.2020.2982196.
30. Fragkiadakis, A.G.; Tragos, E.Z.; Askoxylakis, I.G. A survey on security threats and detection techniques in cognitive radio networks. *IEEE Commun. Surv. Tut.* **2013**, *15*, 428–445. doi:10.1109/surv.2011.122211.00162.
31. Bassoy, S. Self-organised multi-objective network clustering for coordinated communications in future wireless networks. PhD thesis, University of Glasgow, 2020.
32. Jaqaman, K.; Loerke, D.; Mettlen, M.; Kuwata, H.; Grinstein, S.; Schmid, S.L.; Danuser, G. Robust single-particle tracking in live-cell time-lapse sequences. *Nature methods* **2008**, *5*, 695–702. doi:10.1038/nmeth.1237.
33. Khalfi, B.; Hamdaoui, B.; Guizani, M.; Zorba, N. Efficient spectrum availability information recovery for wideband DSA networks: A weighted compressive sampling approach. *IEEE Trans. Wireless Commun.* **2018**, *17*, 2162–2172. doi:10.1109/twc.2018.2789349.
34. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Process. Systems*, 1990, pp. 396–404.
35. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. doi:10.1162/neco.1997.9.8.1735.
36. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* **2015**. doi:10.3115/v1/p15-1150.
37. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances Neural Inf. Process. Systems*, 2015, pp. 802–810.
38. Sak, H.; Senior, A.W.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Proceedings of the Annual Conference of the International Speech Communication Association, (INTERSPEECH)* **2014**, pp. 338–342.
39. He, K.; Sun, J. Convolutional neural networks at constrained time cost. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5353–5360.
40. 3rd Generation Partnership Project (3GPP). *Evolved Universal Terrestrial Radio Access (E-UTRA); (Release 16)*, 2020.

41. Qaraqe, K.A.; Celebi, H.; Gorcin, A.; El-Saigh, A.; Arslan, H.; Alouini, M. Empirical results for wideband multidimensional spectrum usage. In Proceedings of the IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2009, pp. 1262–1266. doi:10.1109/pimrc.2009.5450192.
42. Zhang, G.; Wang, J.; Xiang, Y. Method, apparatus and system for spectrum prediction, 2014. US Patent 8,687,516.
43. Comba. *Outdoor directional quad-band antenna (ODI-065R17M18JJJ-G)*.
44. Arimas. LTE guard band calculation. <https://arimas.com/lte-guard-band-calculation/>. Accessed: 2020-08-17.
45. Chollet, F. Keras. <https://keras.io>. Accessed: 2020-08-14.
46. Beale, M.H.; Hagan, M.T.; Demuth, H.B. Neural network toolbox user's guide. *The Mathworks Inc* **2010**.