Large-Scale Spectrum Occupancy Learning via Tensor Decomposition and LSTM Networks

Ismail Alkhouri*, Mohsen Joneidi*, Farzam Hejazi and Nazanin Rahnavard
Department of Electrical and Computer Engineering
University of Central Florida
Email: {ialkhouri@knights., mohsen.joneidi@, farzam.hejazikookamari@ nazanin@eecs.}ucf.edu

***Abstract*—A new paradigm for large-scale spectrum occupancy learning based on long short-term memory (LSTM) recurrent neural networks is proposed. Studies have shown that spectrum usage is a highly correlated time series over multi-dimensions. Therefore, revealing all these correlations using one-dimensional time series is not a trivial task. In this paper, we introduce a new framework for representing the spectrum measurements in a tensor format. Next, a time-series prediction method based on CANDECOMP/PARAFAC (CP) tensor decomposition and LSTM recurrent neural networks is proposed. Our proposed method is computationally efficient and is able to capture different types of correlation within the measured spectrum. Moreover, it is robust to noise and missing entries of sensed spectrum. The superiority of the proposed method is evaluated over a large-scale synthetic dataset in terms of prediction accuracy and computational efficiency.**

***Index Terms*—Spectrum occupancy learning, Tensor CP decomposition, LSTM time-series prediction.**

I. Introduction

Spectrum occupancy learning (SOL) aims to extract spectrum usage patterns at each frequency band over time. The learned model of spectrum occupancy facilitates the functionality of dynamic spectrum access. Spectrum sensing, optimal channel selection for opportunistic spectrum access, and resource allocation are some examples that can be performed more efficiently by the prediction of spectrum usage [1].

The SOL problem can be regarded as time series learning and prediction. Thus, its performance mainly depends on the underlying model for time series analysis. Many statistical models and methods for spectrum usage prediction have been proposed in the last decade [2]. Auto-regressive models, Markov models [3], [4] and neural networks [5], [6] are exploited as the core model for spectrum time-series prediction. However, spectrum usage is a non-stationary process whose characteristics are time-variant [7]. Other factors such as users' mobility and diverse demands of users make this process more complex. To overcome this challenging problem, deep learning methods are successfully implemented for capturing spectral usage patterns [8], [9]. Long short-term memory (LSTM) and convolutional neural networks (CNNs) are popular models for learning deep networks [10] in various applications such as computer vision [11], pattern recognition, radar, traffic classification [12], and other problems [13], [14]. However, these methods are still challenging for large-scale learning of spectrum time series. The correlation lag of spectrum occupancy over time can be occurred within a very large range. For example, averaged spectrum occupancy may correlate to that of one hour ago, but some network activities are daily or weekly [15]. Thus, spectrum occupancy at one time could be related to spectrum occupancy of one day or a week ago as well. Likewise, there might exist some spectrum patterns even in a larger scale over time. While conventional time-series prediction methods fail to reveal correlations in large lags, LSTM is able to capture these patterns. However, there are two issues in the large-scale data scenario. First, learning and prediction of an extremely long time series implies capturing all the spectrum correlations efficiently; thus, the computational burden of learning and updating the LSTM model may not be tractable for online tracking of spectrum occupancy. Second, dealing with missing entries in the learning phase is inevitable for a real data sequence as it affects the prediction accuracy in the test phase. We propose to utilize *tensor-based* data completion methods that has attracted many attentions for data processing in the presence of missing entries [16].

This paper proposes a new high-dimensional structure for sensed spectrum data in order to improve accuracy and scalability of LSTM for large-scale SOL. A joint problem of data interpolation and extrapolation (completion and prediction) is introduced. Tensor CP decomposition provides a reliable low-dimensional representation of data, and LSTM performs a fast prediction on the lower-dimension data (decomposed factors).

Correlations with long lags are vulnerable to be forgotten in the naive vector-based representation. However, these correlations can be identified in a much smaller lag in a new dimension of a tensor. In the present paper, tensor-based representation of time series is exploited in order to extract some basic time series known as CP factors of a tensor. These factors are robust to noise and missing entries. Large-scale prediction of all time series over long-time dimension only requires the prediction of CP factors of the measured tensor. This significant advantage can be considered as a big data reduction technique.

The main contributions of this paper are summarized as follows:

- A novel time-series prediction framework is proposed based on tensor decomposition and LSTM networks. Our framework can be employed for large-scale spectrum occupancy learning among many other large-scale time-series prediction applications.

* Indicates shared first authorship. This material is based upon work supported by the National Science Foundation under Grant No. CCF-1718195.

- Computational burden is decreased by solely performing prediction on low-dimensional CP factors rather than high-dimensional raw data.
- The problem of missing samples in the time-series prediction is addressed using tensor completion techniques.

Throughout this paper, $\mathcal{X}$ denotes a three-way tensor and $\boldsymbol{X}$ denotes a matrix. Mode-n fiber of a tensor is a vector obtained by fixing all modes except the n^{th} mode and mode-n matricized version of a tensor is denoted by $\boldsymbol{X}_{(n)}$. $\boldsymbol{x}$ and x represent a vector and a scalar, respectively. Hadamard product, outer product, and Khatri-rao product are denoted as $*$, $\circ$, and $\odot$, respectively [17].

The remaining of this paper is organized as follows. Section II presents a brief description of the CP decomposition of tensors, and other prerequisites. The proposed method is presented in Section III. In Section IV, we will show the experimental results, followed by conclusion in Section V.

II. Background and System Model

In this section, the prerequisite background is presented, then the system model for spectrum aggregation is explained.

A. Tensor CP Decomposition

A Tensor is a multi-dimensional array. Since their introduction, tensors have been utilized in plethora of applications as they bring a concise mathematical framework for formulating problems involving high-dimensional data or big data especially in signal processing literature [18], [19]. The CP decomposition factorizes a 3-dimensional tensor $\mathcal{X} \in \mathbb{R}^{F\times T\times N}$ of rank R into a sum of R number of rank-1 tensors which can be represented as [17]

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \quad \triangleq < \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} >, \tag{1}$$

where, $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ are the CP factors of the r^{th} component and the r^{th} column of factor matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively. In other words, $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \ldots \boldsymbol{a}_r] \in \mathbb{R}^{F\times R}$. Similarly, $\boldsymbol{B} \in \mathbb{R}^{T\times R}$ and $\boldsymbol{C} \in \mathbb{R}^{N\times R}$ are defined.

The tensor $\mathcal{X}$ can be matricized as follows [17]

$$\begin{aligned} \mathbf{X}_{(1)} &= \mathbf{A}(\mathbf{C}\odot\mathbf{B})^T, \\ \mathbf{X}_{(2)} &= \mathbf{B}(\mathbf{C}\odot\mathbf{A})^T, \\ \mathbf{X}_{(3)} &= \mathbf{C}(\mathbf{B}\odot\mathbf{A})^T. \end{aligned} \tag{2}$$

A powerful property of high-order tensors is that their rank decomposition is unique under milder conditions compared to matrices [20]. This interesting property of tensors is attractive for communication systems [21] and specially in channel estimation and blind coding in MIMO systems [22], [23].

The CPD used for this paper is done by the alternating least squares (ALS) method proposed by Carrol, and Harshman [24], [25]. The goal is to calculate a CPD with R components that approximates $\mathcal{X}$ accurately, i.e., to obtain

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\| \quad \text{s.t.} \ \ \hat{\mathcal{X}} = \sum_{r=1}^{R} = \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{3}$$

ALS iteratively fixes $\mathbf{B}$ and $\mathbf{C}$ to solve for $\mathbf{A}$, then fixes $\mathbf{A}$ and $\mathbf{C}$ to solve for $\mathbf{B}$, and then fixes $\mathbf{A}$ and $\mathbf{B}$ to solve for $\mathbf{C}$ [17]. We refer to this algorithm as the plain CP algorithm.

B. LSTM Network

A neural network is a powerful tool for activity pattern learning [26]. Similar to other neural network structures, an LSTM network consists of an input layer, hidden layer(s), and an output layer [27]. An RNN network provides the feature of internal memory maintenance i.e., it saves the information of previous time slots. This technique has addressed the problem of gradient explosion which means that the network overwrites its memory in an uncontrolled manner. However, LSTM fixes the issue encountered in the conventional RNNs by adding an adaptive memory unit, which is its key component. This adaptive memory unit controls saving dominant samples and forgetting obsolete data which enables LSTM to track information over longer periods of time.

C. System model

Consider a spectral sensor that captures the power spectrum density (PSD) of an RF band of F frequency bins at T times a day. Therefore, we will have a matrix of size $T \times F$ for each day of recording. Additionally, assume data is recorded for N days. Corresponding to one frequency bin f, there exists a matrix of size $T \times N$ that presents occupancy changes over all times. Similarly, corresponding to one instant of time t, a matrix of length $F \times N$ represents the occupancy of spectrum over all channels and days in a specific time of day. After N days (frontal slices), the occupancy of the f^{th} frequency channel at t^{th} time slot is the subject of prediction for the upcoming day. The proposed data arrangement is shown in Fig.1.

To forecast the values of next time steps of a sequence, we utilize a predictor that trains a regression network. Four types of training networks are used in this paper, the auto-regressive model (AR), support vector machines (SVM), convolutional neural networks (CNN), and LSTM.

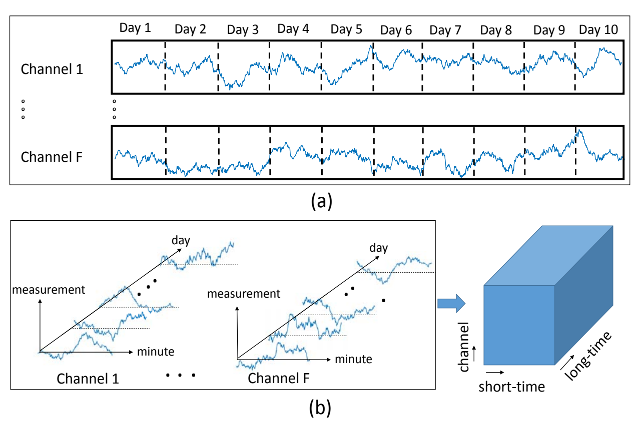


Fig. 1: Comparing two representing methods of F time series. (a) Matrix-based representation. (b) Tensor-based representation.

III. Proposed Method

Tensor decomposition learns a few principle factors for each way, such that all fibers in the corresponding way of the tensor can be reconstructed using the linear combination of the learned fibers. Tensor CP representation is a concise model, and it is robust to perturbations and missing entries. It is shown that a low-rank tensor can be recovered from a small number of entries using CP decomposition. In other words, the CP factors of the original tensor and the CP factors of the partial and noisy replica of the original tensor are close to each other [28]. These attractive characteristics of structured data in a high-dimensional tensor motivate us to employ tensor CP decomposition for dynamic spectrum completion and prediction.

Consider a rank-1 tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{F\times T\times N}$ with a third dimension over long time variable n in Fig. 2. This tensor consists of FT time series (fibers) alongside its third way. Because $\boldsymbol{\mathcal{X}}$ has rank 1, all these time series are a scale of a vector $\boldsymbol{c}$, which is a basic time series. This vector is broken down into two parts, the given part, $\boldsymbol{c}_L$, which corresponds to the known part of tensor, and the unknown part, $\boldsymbol{c}_P$, which corresponds to the part of the tensor to be predicted. Prediction of all unknown variables of tensors is equivalent to prediction of $\boldsymbol{c}_P$. For a general rank-R tensor, there exist R basic time series that span the space of all fibers of the tensor in the third way. Thus, the prediction of R temporal factors enables us to predict any time series of the tensor. Suppose our source

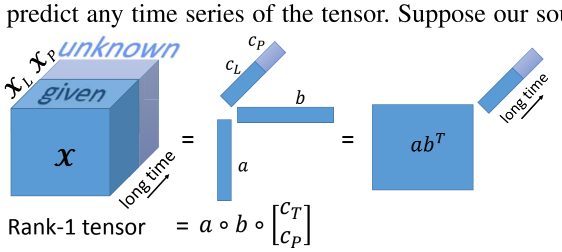


Fig. 2: A rank-1 tensor is the outer product of 3 vectors and it can be cast as the modulation of a rank-1 matrix with a temporal pattern.

of data is dynamic, therefore, obsolete data might degrade the result of prediction. To tackle this problem, only recent slices are considered for learning. The number of slices for each epoch of prediction is referred as the length of training, N_L. Likewise, we define the length of prediction, N_P, where $N = N_P + N_L$ is equal to the size of the third dimension of the underlying tensor. The proposed tensor-based prediction solves two following consecutive problems to predict the unknown entries of the tensor over time:

$$(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_L) = \underset{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}_L}{\operatorname{argmin}} \ \|\mathcal{X}_L - <\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_L>\|_F^2, \tag{4a}$$

$$\mathcal{X}_P = \ <\boldsymbol{A}, \boldsymbol{B}, h(\boldsymbol{C}_L, \Omega)>, \tag{4b}$$

in which, $h(.\,,.)$ represents a model for time-series prediction and Ω is the set of model's parameters. We will investigate the effect of the prediction model on the performance of the whole framework. AR, SVM, CNN, and LSTM are studied as core models for prediction. However, our main proposed algorithm is LSTM-based. Fig. 3 shows the block diagram of the proposed prediction algorithm.

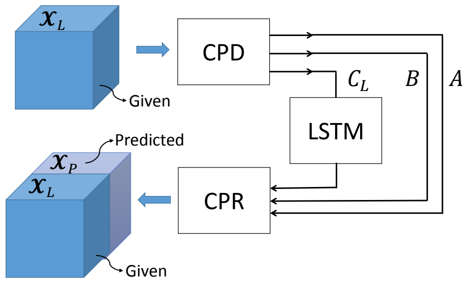


Fig. 3: Block diagram of our tensor-based time-series prediction.

CP decomposition reveals the latent factors of data from different perspective, and LSTM predicts the long temporal factors. The extracted factors using CP and extrapolated factors using LSTM can produce a tensor by CP reconstruction (CPR).

The computational complexity of our proposed approach can be divided into two main terms. The First one is CPD with complexity order of $O(R\max(F,T,N)^3)$. Secondly, the complexity of LSTM per time step is given as $O(W)$ where W is the number of weights [27].

Matrix and tensor completion are state-of-the-art methods for data completion in many applications [29], [30], [31]. The proposed tensor-based scheme can be extended to the joint completion and prediction in a straightforward formulation. Assume a given incomplete tensor, $\mathcal{X}_L^I$ and a mask tensor with the same size of the data tensor, $\mathcal{M} \in \{0,1\}^{F\times T\times N_L}$. The entries corresponding to 0 are not measured. Tensor $\mathcal{X}$ is modeled by a rank-R tensor, while missing entries corrupt the intrinsic structure; thus, the given $\mathcal{X}_L^I$ is not rank-R anymore. However, the incomplete tensor can be completed iteratively such that the completed tensor is low-rank after performing iterations between the following steps,

$$\mathcal{X}_L = \mathcal{M} * \mathcal{X}_L^I + (\underline{\mathbf{1}} - \mathcal{M}) * <\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_L> \tag{5a}$$

$$(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_L) \leftarrow \text{CP decomposition on} \mathcal{X}_L \tag{5b}$$

Here, $\underline{\mathbf{1}}$ is a tensor with all entries equal to 1. The given data in $\mathcal{X}_L^I$ is kept for known entries of the mask and the missed data is estimated via CP factors for unknown entries of the mask. Updating the incomplete tensor enables the algorithm to estimate a more accurate set of factors in the next iteration. Thus, CP factors and missing entries can be updated iteratively. The CP factors are initialized before the first iteration by performing CP decomposition on $\mathcal{M} * \mathcal{X}_L^I$. Alg. 1 shows our proposed method for *joint time series completion and prediction*. The main loop of the algorithm completes data to find a more fitted set of CP factors. Then, LSTM predicts the long-time factors, and the predicted time series are resulted by CP reconstruction. The tensor completion is performed using iterative CP decomposition and data interpolation.

The CP step (in (5b) or Line 3 of Alg. 1) does not use information of the mask, and the mask is used only for data interpolation. A modified version of CP decomposition is presented in Alg. 2 that infuses information of the mask in order to estimate CP factors. The optimized CP for incomplete data can be employed in line 3 of Alg. 1 instead of the plain CP in order to estimate more accurate factors.

Algorithm 1 Time-series completion and prediction via tensor CP decomposition and LSTM prediction.

Input: Incomplete Tensor $\mathcal{X}_L^I$, mask $\mathcal{M}$ and rank R.
Output: Completed and predicted tensor $\hat{\mathcal{X}}$
1: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_L \leftarrow$ CP decomposition on $\mathcal{M} * \mathcal{X}_L^I$ with rank R.
While (The stopping criterion is not met)
2: $\mathcal{X}_L \leftarrow$ using Eq. (5a).
3: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_L \leftarrow$ CPD on $\mathcal{X}_L$ with rank R using Eq. (5b).
End
4: $\boldsymbol{C}_P \leftarrow$LSTM on each column of $\boldsymbol{C}_L$
5: $\boldsymbol{C} \leftarrow$ concatenate $\boldsymbol{C}_L$ and $\boldsymbol{C}_P$
6: $\hat{\mathcal{X}} \leftarrow \ <\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}>$.

Algorithm 2 Optimized CP for incomplete data.

Input: Tensor $\mathcal{X}$, mask $\mathcal{M}$, and rank R.
Output: CP factors of $\mathcal{X}$
1: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \leftarrow$ Plain CP decomposition on $\mathcal{X}$ [17].
While (The stopping criterion is not met)
2: $\boldsymbol{A} \leftarrow \underset{\boldsymbol{A}}{\text{minimize}} \ \|\boldsymbol{M}_{(1)} * (\boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^T)\|_F^2$
3: $\boldsymbol{B} \leftarrow \underset{\boldsymbol{B}}{\text{minimize}} \ \|\boldsymbol{M}_{(2)} * (\boldsymbol{X}_{(2)} - \boldsymbol{B}(\boldsymbol{C} \odot \boldsymbol{A})^T)\|_F^2$
4: $\boldsymbol{C} \leftarrow \underset{\boldsymbol{C}}{\text{minimize}} \ \|\boldsymbol{M}_{(3)} * (\boldsymbol{X}_{(3)} - \boldsymbol{C}(\boldsymbol{B} \odot \boldsymbol{A})^T)\|_F^2$
End

SOL can be regarded as a learning-based detection where the problem is to detect whether a channel is occupied or not. Our decision rule for detection is based on the output of our proposed algorithm. Assume $\hat{x}_{ftn}$ is the predicted spectrum value at frequency channel f, time t and day n. Two hypotheses are considered for spectrum occupancy status for this entry as given by

$$S(f,t,n) = \begin{cases} \text{OCCUPIED} & \text{if } \hat{x}_{ftn} \geq \gamma \\ \text{NOT OCCUPIED} & \text{if } \hat{x}_{ftn} < \gamma \end{cases} \tag{6}$$

In which, $S(f,t,n)$ indicates the estimated occupancy status at frequency channel f, time t and day n and γ is a threshold for operating the designed detector. As γ increases, both the probability of detection and the probability of false alarm will decrease. Receiver operating characteristic (ROC) of the proposed detector is able to find the optimum threshold to achieve the desired false alarm rate.

IV. Experimental Results

In the following experiments, we assume that 20 frequency channels are sensed. PSD of each frequency channel is recorded 10 times an hour, i.e., there exist 240 measurements from the spectrum for each mode-2 fiber. Moreover, it is assumed that the recording for 100 days is available. Therefore, $F = 20$, $T = 240$, and $N = 100$.

Synthetic dataset for time t at day n and frequency f follows the joint probability distribution of $P(t,n,f) = P_t(t)P_n(n)P_f(f)$ where each distribution is generated according to the below model,

$$P_t(t) = \sum_{i=1}^{3} \beta_i \mathcal{N}(\tau_i, \sigma_i^2), \tag{7a}$$

$$P_n(n|j) = \mathcal{N}(\mu_j, \lambda_j^2), \text{ for } j = n \bmod 7, \tag{7b}$$

$$P_f(f) = \mathcal{U}[1, 2, ..., F]. \tag{7c}$$

Eq. (7a) is the probability of spectrum occupation in a typical day which is modeled by a Gaussian mixture model (GMM) with three peaks at 3pm, 6pm, and 9pm. Parameters $\{\beta_i, \tau_i, \sigma_i\}$ are designed to satisfy the desired pattern of GMM[1]. The conditional probability of occupancy over days follows (7b). The condition determines that n corresponds to which day of week. The parameters $\{\mu_j, \lambda_j\}$ are designed such that at Mondays, Tuesdays, Wednesdays, and Thursdays, the spectrum is more occupied than Fridays, and Friday is busier than the weekend[2] [8]. In addition, there is no preference for frequency occupation of a user which leads to a uniform distribution with equal probabilities over all frequency bins, which is employed in (7c). This model is inferred from previous work [15].

The employed LSTM architecture has 4 hidden layers. Learning rate is 0.05 and the number of epochs is 300 with ADAM optimizer. Intel Corei7 CPU with 4.20GHz is used for performing simulations on MATLAB 2018b.

The CPD-ALS algorithm determines the factors of the tensor numerically by solving alternating optimization problems. Calculating CP rank of a tensor is an NP-hard problem. However, it is upper bounded by the following inequality [17]

$$\text{Rank}(\mathcal{X}) \leq \min(FT, FN, TN).$$

A practical solution for finding rank is to start with a low number, compute the normalized reconstruction error, and increase it as needed. Normalized error is obtained as a function of rank as follows,

$$e_{cpd}(R) = \frac{||\mathcal{X} - \hat{\mathcal{X}}(R)||_F}{||\mathcal{X}||_F}. \tag{8}$$

In this equation, $||.||_F$ denotes matrix Frobinious norm and R is an integer number between 1 and the maximum rank. Tensor $\mathcal{X}\hat{(R)}$ is the rank-R approximation of $\mathcal{X}$ optimized by a tensor decomposition algorithm. The goal is to select the lowest rank that approximates $\mathcal{X}$. The effect of rank for training the basic time series will be investigated later. In this experiment, results of the proposed method is exhibited. The synthesized data is organized into an $F \times T \times N$ tensor. In which $F = 20$ (20 frequency bins), $T = 240$ (240 measurements per days), and $N = 100$ (100 days). With rank 10, CP decomposition provides $\mathbf{A} \in \mathbb{R}^{20\times10}$, $\mathbf{B} \in \mathbb{R}^{240\times10}$, and $\mathbf{C} \in \mathbb{R}^{100\times10}$. In order to evaluate prediction performance, the underlying tensor is broken into two tensors, (i) the learning tensor, $\mathcal{X}_L$, and (ii) the test tensor, $\mathcal{X}_P$, that is the subject of prediction. In this experiment $N_L = 80$ days are used for learning and $N_P = 20$ days are considered for prediction.

[1] We set β_1=0.5, β_2=0.3, β_3=0.2, τ_1=150 (3PM), τ_2=180 (6PM), τ_3=210 (9PM), and σ_i=20 (2hours).

[2] We set $\mu_0 = \mu_1 = \mu_2 = \mu_3 = 1$, $\mu_4 = 0.5$, $\mu_5 = \mu_6 = 0.2$, and $\lambda_j = 0.1\mu_j$.

The obtained long-time CP factors, $\boldsymbol{C}_L \in \mathbb{R}^{80\times 10}$ are exploited to predict $\boldsymbol{C}_P \in \mathbb{R}^{20\times 10}$. Each column of $\boldsymbol{C}_L$ is a pseudo-time series that is employed for prediction of $\boldsymbol{C}_P$ independently. Predicted values from AR, SVM, CNN, and LSTM training networks are computed. We also calculated the prediction of the matrix-based data using the aforementioned training methods to demonstrate the impact of utilizing CPD. Numerical comparison with other methods is presented in Table I. Tensor-based methods improve prediction accuracy as well as save the computational burden.

TABLE I: Normalized Prediction Error and Processing Time (sec)

Method	CPD time	Learning time	Total time	Error%
AR [32]	N/A	55.12	55.12	33.55
AR+CPD	3.71	4.23	7.94	21.83
SVM [26]	N/A	1202.21	1202.21	23.78
SVM+CPD	3.71	20.52	24.23	16.94
CNN [8]	N/A	496.44	496.44	22.40
CNN+CPD	3.71	15.87	19.58	17.81
LSTM [33]	N/A	2389.96	2389.96	23.71
LSTM+CPD	3.71	12.01	15.72	**15.26**

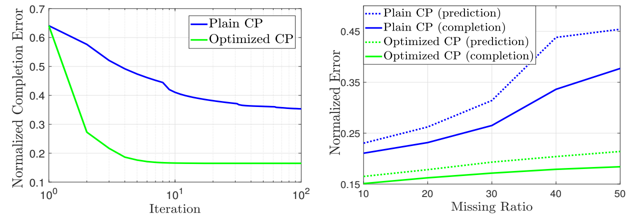
Employing LSTM for prediction of CP factors exhibits the best results, and it decreases computation cost in comparison to the plain LSTM on the set of raw time series. The normalized error is computed using the following rule,

$$e_p = \frac{\sum (x_i - \hat{x}_i)^2}{\sum x_i^2},$$

where x_i and $\hat{x}_i$ are the actual and the predicted values in the time series.

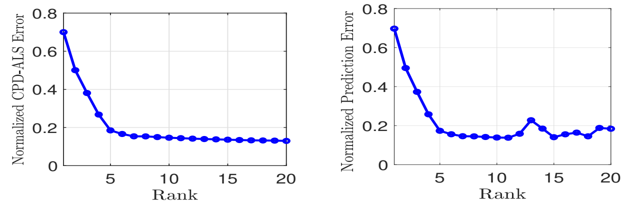
It can be observed that each prediction technique is improved by employing CPD. Our proposed method, LSTM+CPD, returns the best performance in terms of the normalized prediction error. In general, LSTM outperforms methods based on AR, SVM, or CNN [32], [26], [8]. It is worthwhile to notice that our proposed method predicts spectrum occupancy more accurately than performing LSTM on raw time series data [33]. On top of the enhanced prediction error, CPD achieves a massive data reduction. Table I demonstrates the processing time for each method and illustrates that exploiting CPD is able to diminish the total running time of prediction rigorously.

In the next experiment, the proposed method, Alg. 1, is employed for missing spectrum recovery when a portion of spectrum measurements is missing. To this aim, the whole tensor is assumed to be incomplete. Therefore, random measurements from a $F \times T \times N$ tensor are available to recover the whole tensor. The proposed spectrum completion algorithm requires performing CPD in each iteration of completion. It is shown that employing the modified CP for incomplete data, Alg. 2, is more effective for missing spectrum recovery. Each iteration of data completion using the optimized CP needs more computations. However, the number of needed iterations for the modified CP is much less than the plain CP. Fig. 4 (a) shows the performance of our proposed time series completion method using the plain CP and the introduced CP versus iteration of data completion in Alg. 1.



(a) (b)

Fig. 4: Normalized completion error using the proposed method in Alg. 1 (a) Over iterations. (b) For different missing ratios.



(a) (b)

Fig. 5: Normalized error of prediction vs. the assumed rank for the underlying tensor. a) for learning tensor and b) for the test tensor. As the rank increases the learning error decreases. However, increasing rank causes over-learning for prediction. Thus, prediction error is not necessarily decreasing.

The performance of our proposed joint completion and prediction problem is presented for different missing ratios. The plain CP algorithm and the modified CP are compared for performing Alg. 1 to solve the joint problem. In this experiment, time series of 80 days are considered for learning and 20 days for prediction. The learning tensor, $\mathcal{X}_L$, is assumed to have missing entries. As it can be seen in Fig. 4 (b), our proposed algorithm successfully completes data in terms of the normalized error and predicts time series using LSTM. As previously stated, the modified CP outperforms plain CP in the presence of missing entries. The prediction error is close to that of exploiting all data for learning that is presented in Table I. For example, in presence of 10% missing entries for learning, the prediction error is 16.53%. This number is close to 15.26% which is obtained by learning using the full tensor.

Each component of CP decomposition learns some patterns of data. Selection of the rank equal to R provides R set of factors that reconstruct the learning tensor. As the assumed rank increases, more details about learning tensor are captured and the reconstruction error decreases. Fig. 5 (a) shows the reconstruction error of the learning tensor versus the assumed rank. However, learning fine details does not help prediction. Thus, the imposed rank cannot be a large number. For example, in Fig. 5 assuming rank as a number between 5 and 10 is reasonable. Fig. 5 (b) shows the performance of prediction using LSTM versus the selected rank of CP for decomposition of the learning tensor. As it is shown, beyond rank 10, the normalized error of prediction is not decreasing monotonically.

The last experiment of this paper shows the performance of spectrum occupancy detection. Two hypotheses are considered based on (6). The detection performance is determined using a ground truth of spectrum occupancy from the synthesized data. Our proposed spectrum prediction results in a value for spectrum in each channel over time. The value turns into a decision rule by (6). Probability of detection, P_D, vs.

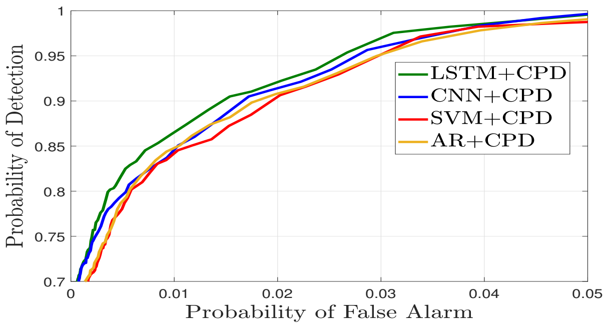


Fig. 6: ROC of the proposed detector in Eq. 6.

probability of false alarm, P_F, are plotted by applying different values for the threshold. Utilization of AR, SVM, CNN, and LSTM on the tensor-based prediction is compared by their ROC graph in Fig. 6. LSTM exhibits better performance for detection of free channels. It means that with a fixed false alarm rate, the probability of detection using the proposed LSTM-based method is higher than that of the other methods.

V. Conclusion

A combination of tensor decomposition and LSTM time-series prediction is proposed as a new paradigm for large-scale spectrum occupancy prediction. The measured spectrum data is organized into a 3-way tensor. The CPD-ALS algorithm is performed to obtain CP factors for big data reduction and learning reliable patterns of data. The LSTM network is then utilized to predict CP factors in order to estimate future spectrum occupancy patterns over time and for all frequency channels. Employing LSTM as the core predictor of CP factors outperforms other schemes such as AR, SVM, and CNN.

References

[1] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5g: When cellular networks meet artificial intelligence," *IEEE Wireless communications*, vol. 24, no. 5, pp. 175–183, 2017.

[2] J. Zhang, G. Ding, Y. Xu, and F. Song, "On the usefulness of spectrum prediction for dynamic spectrum access," in *Wireless Communications & Signal Processing (WCSP), 2016 8th International Conference on*. IEEE, 2016, pp. 1–4.

[3] H. Eltom, S. Kandeepan, Y. C. Liang, B. Moran, and R. J. Evans, "HMM based cooperative spectrum occupancy prediction using hard fusion," in *Communications Workshops (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 669–675.

[4] J.-W. Wang and R. Adriman, "Analysis of opportunistic spectrum access in cognitive radio networks using hidden markov model with state prediction," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 10, 2015.

[5] S. Bai, X. Zhou, and F. Xu, "Spectrum prediction based on improved-back-propagation neural networks," in *Natural Computation (ICNC), 2015 11th International Conference on*. IEEE, 2015, pp. 1006–1011.

[6] A. A. Eltholth, "Spectrum prediction in cognitive radio systems using a wavelet neural network," in *Software, Telecommunications and Computer Networks (SoftCOM), 2016 24th International Conference on*.

[7] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "A distribution-based approach to anomaly detection and application to 3g mobile traffic," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–8.

[8] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Communications Letters*, 2018.

[9] L. Yu, J. Chen, and G. Ding, "Spectrum prediction via long short term memory," in *Computer and Communications (ICCC), 2017 3rd IEEE International Conference on*. IEEE, 2017, pp. 643–647.

[10] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and discriminative text classification with recurrent neural networks," *arXiv preprint arXiv:1703.01898*, 2017.

[11] Y. Fan, X. Lu, D. Li, and Y. Liu, "Video-based emotion recognition using cnn-rnn and c3d hybrid networks," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. ACM.

[12] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, "Mimetic: Mobile encrypted traffic classification using multimodal deep learning," *Computer Networks*, vol. 165, p. 106944, 2019.

[13] V. Jithesh, M. J. Sagayaraj, and K. Srinivasa, "LSTM recurrent neural networks for high resolution range profile based radar target classification," in *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*. IEEE, 2017.

[14] M. Taherisadr, M. Joneidi, and N. Rahnavard, "EEG signal dimensionality reduction and classification using tensor decomposition and deep convolutional neural networks," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*.

[15] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, "Primary users in cellular networks: A large-scale measurement study," in *New frontiers in dynamic spectrum access networks, 2008. DySPAN*.

[16] B. Li, X. Zhang, X. Li, and H. Lu, "Tensor completion from one-bit observations," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 170–180, 2019.

[17] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[18] F. Hejazi, M. Joneidi, and N. Rahnavard, "A tensor-based localization framework exploiting phase interferometry measurements," in *2020 IEEE Radar Conference (RadarConf)*.

[19] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[20] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 14, no. 3, pp. 229–239, 2000.

[21] M. Joneidi and N. Rahnavard, "Primary user localization and online radio cartography via structured tensor decomposition," *arXiv preprint arXiv:1905.04284*, 2019.

[22] C. Qian, X. Fu, N. D. Sidiropoulos, and Y. Yang, "Tensor-based channel estimation for dual-polarized massive MIMO systems," *arXiv preprint arXiv:1805.02223*, 2018.

[23] M. N. da Costa, G. Favier, and J. M. T. Romano, "Tensor modelling of MIMO communication systems with performance analysis and kronecker receivers," *Signal Processing*, vol. 145, pp. 304–316, 2018.

[24] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[25] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis," 1970.

[26] F. Azmat, Y. Chen, and N. Stocks, "Analysis of spectrum occupancy using machine learning algorithms," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 6853–6860, 2016.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] A. Wang, D. Wei, B. Wang, and Z. Jin, "Noisy low-tubal-rank tensor completion through iterative singular tube thresholding," *IEEE Access*, vol. 6, pp. 35 112–35 128, 2018.

[29] A. Esmaeili, K. Behdin, M. A. Fakharian, and F. Marvasti, "Transductive multi-label learning from missing data using smoothed rank function," *Pattern Analysis and Applications*, pp. 1–9, 2020.

[30] A. Montanari and N. Sun, "Spectral algorithms for tensor completion," *Communications on Pure and Applied Mathematics*, vol. 71, no. 11, pp. 2381–2425, 2018.

[31] J. Yang, Y. Zhu, K. Li, J. Yang, and C. Hou, "Tensor completion from structurally-missing entries by low-tt-rankness and fiber-wise sparsity," *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[32] A. Eltholth, "Forward backward autoregressive spectrum prediction scheme in cognitive radio systems," in *Signal Processing and Communication Systems (ICSPCS), 2015 9th International Conference on*. IEEE, 2015, pp. 1–5.

[33] L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun, "Spectrum prediction based on taguchi method in deep learning with long short-term memory," *IEEE Access*, vol. 6, pp. 45 923–45 933, 2018.