A Dynamic Bernstein Graph Recurrent Network for Wireless Cellular Traffic Prediction

Ali Mehrabian, Shahab Bahrami, and Vincent W.S. Wong
Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
email: {alimehrabian619, bahramis, vincentw}@ece.ubc.ca

***Abstract*—Predictive analysis of wireless cellular traffic plays an important role in network resources provisioning. Accurate traffic prediction is a challenging task due to the dynamic spatial-temporal nature of wireless traffic. Most of the existing approaches do not consider spectral domain information for wireless traffic prediction. Some of the approaches cannot capture the spatial dependencies between neighbouring and distant cells. In this paper, we propose a dynamic Bernstein graph recurrent network for traffic prediction in wireless cellular networks. First, we design a spectral dynamic graph construction (SDGC) method to model the spatial dependencies between cells as a dependency graph in a data-driven fashion. A dynamic Bernstein polynomial filtering (DBPF) scheme based on the K-order Bernstein polynomial approximation is then developed to capture the spatial correlations and infer the cell-specific parameters. To predict the spatial-temporal traffic demands, we propose a dynamic Bernstein graph recurrent network (DBGRN), which integrates the proposed DBPF module with a gated recurrent unit (GRU) network. We evaluate the performance of our proposed model using a real-world dataset. Results show that our proposed model outperforms four state-of-the-art baseline schemes, and achieves up to 8% and 10% improvements in terms of the root mean squared error (RMSE) and mean absolute error (MAE), respectively.**

I. Introduction

With the advancement of the fifth generation and beyond (B5G) wireless systems, fine-grained analysis of wireless traffic is a fundamental building block for optimizing network resources and supporting different use cases. As an example, proactive monitoring of wireless traffic enables mobile service providers to partition network resources by using network slicing and network functions virtualization. This results in an improvement of the spectral and energy efficiencies.

The accuracy of traffic prediction depends on several contributing factors. First, the traffic demand of users can vary substantially. As shown in Fig. 1, the Internet traffic in three regions of Milan, Italy, exhibit dynamic patterns in the temporal domain [1]. Other factors such as time periods (e.g., holidays vs. weekdays, peak hour vs. off-peak hour) and usage of different types of mobile applications (e.g., streaming videos, social networks) have an impact on traffic patterns [2]. Second, user mobility introduces spatial dependencies in the traffic data between neighbouring cells. Moreover, geographically distant areas may exhibit similar traffic patterns [3]. This implies that physical proximity may not always provide sufficient information on spatial dependencies between cells.

To address the aforementioned challenges, data-driven approaches have been proposed in the literature. Recurrent neural networks, such as long short-term memory (LSTM) model [4], have been utilized to capture the long-term temporal dependencies. Convolutional neural network (CNN)-based methods have been proposed to capture the local spatial relationships [5]. In [6], a deep 3D residual convolutional network is proposed to capture the local spatial-temporal features. It is followed by an attention-aided convolutional LSTM network to learn the long-term spatial-temporal dependencies. However, the CNN-based models can only extract spatial dependencies between neighbouring cells.

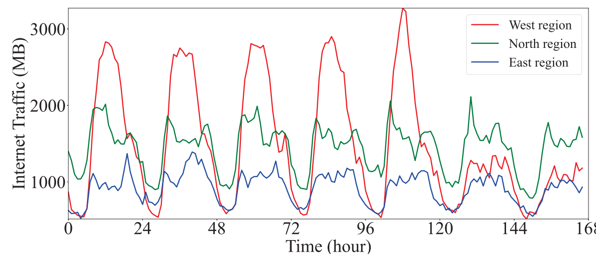


Fig. 1. Illustration of dynamic patterns of Internet traffic service at randomly selected cells in three regions of Milan, Italy, for the duration of one week. The dataset is from Telecom Italia [1].

Recently, a new line of research focuses on graph neural networks (GNNs) with applications to different domains, including traffic prediction [7], [8]. Some recent works have exploited graph convolutional network (GCN) architecture [9], which utilizes the first-order Chebyshev polynomial expansion, to approximate the graph convolutional operation [10], [11]. In [10], the handover data is used to design a pre-defined graph structure. An integration of the GCN architecture with the gated recurrent unit (GRU) is proposed to capture the spatial-temporal dependencies. Moreover, different methods have been proposed to obtain the graph structure. In [3], the dynamic time wrapping algorithm is utilized to construct a weighted graph. An attention-aided recurrent network is proposed to capture the spatial-temporal dependencies. A collaborative global-local learning strategy is then developed to improve the training performance. In [12], a multi-view spatial-temporal graph network is proposed, which captures various local and global spatial-temporal correlations by using multi-head attention mechanism.

In spite of the favourable results of the aforementioned models, several challenges have yet to be addressed. First, the GCN architecture serves as a low-pass filter by using a polynomial spectral graph filtering scheme. However, the

GCN architecture may not be feasible for large-scale wireless systems. This is because more high-level filter structures, such as narrow-band and comb filters, are required to capture the dynamic traffic patterns. Moreover, pre-determined graph construction methods based on similarity measures undermine the ability to infer the spatial dependencies [7]. To tackle these challenges, in this paper, we propose a graph recurrent neural network for wireless cellular traffic prediction. The proposed model utilizes the information in the spatial, temporal, and spectral domains. The main contributions of this paper are summarized as follows:

- *Construction of the Dependency Graph*: To characterize the spatial dependencies between cells, we propose a spectral dynamic graph construction (SDGC) method to model the network topology as a graph structure. The proposed SDGC module extracts the cell dependencies by leveraging the discrete Fourier transform (DFT) and GRU in the frequency domain. It then uses self-attention mechanism to determine the connections between each pair of cells in a data-driven fashion.
- *Spatial-Temporal Traffic Prediction*: We develop a dynamic Bernstein polynomial filtering (DBPF) module to infer the spatial dependencies and learn the cell-specific parameters dynamically by utilizing the K-order Bernstein polynomial expansion. We propose a dynamic Bernstein graph recurrent network (DBGRN), which integrates the developed DBPF module with a GRU to predict the spatial-temporal traffic demands.
- *Performance Evaluation*: We evaluate the performance of our proposed model by conducting extensive experiments using a real-world dataset. The results show that our proposed model outperforms four state-of-the-art baseline models, and achieves up to 8% and 10% improvements in terms of the root mean squared error (RMSE) and mean absolute error (MAE), respectively. Furthermore, we demonstrate the performance gain of using the K-order Bernstein polynomial expansion over the K-order Chebyshev polynomial expansion. We also show the importance of incorporating spectral domain information in graph dependency construction.

The remainder of this paper is organized as follows. In Section II, we present the problem formulation and our proposed model. In Section III, we provide the performance evaluation of the proposed model and comparisons via experiments. Conclusions are given in Section IV.

Notations: In this paper, we use $\mathbb{R}$ to denote the set of real numbers. We use $(\cdot)^{\mathrm{T}}$ to denote the transpose of a vector or matrix. i represents the imaginary unit satisfying the equation $i^2 = -1$. We use $\odot$ for element-wise product. We use boldface upper-case letters (e.g., $\mathbf{X}$) to denote two-dimensional matrices or multiple-dimensional tensors and boldface lower-case letters (e.g., $\mathbf{x}$) to denote vectors. We use $*$ to denote the graph convolutional operation. $\mathbf{I}_N$ represents the identity matrix of size N. We use $[\cdot]$ to denote the concatenate operation. We use $\otimes$ to denote the tensordot as a multi-dimensional operation.

II. System Model and Proposed Model

In this section, we first present the system model and the wireless traffic prediction problem. Then, we present our proposed model. We describe the construction process of the dependency graph between cells by using the proposed SDGC method. Next, we present the DBPF module, which can capture the spatial dependencies and infer the cell-specific parameters. Finally, we propose the DBGRN framework, which can predict the spatial-temporal wireless traffic.

A. System Model

Consider a geographical area is served by N cells or base stations. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denote the set of cells. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the set of time steps, where T represents the total number of time steps. Let x_t^n denote the traffic volume of cell $v_n \in \mathcal{V}$ in time step $t \in \mathcal{T}$. We denote the traffic vector for cell v_n as $\mathbf{x}^n = (x_1^n, \ldots, x_T^n)$ and the traffic vector observed in time step t as $\mathbf{x}_t = \left(x_t^1, \ldots, x_t^N\right)$. The network topology is modeled as an undirected weighted graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of cells (or nodes) and $\mathcal{E} = \{(v_m, v_n) \,|\, v_m, v_n \in \mathcal{V}\}$ is the set of edges. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ denote the adjacency matrix of graph G, with each element $\mathbf{A}_{m,n}$ corresponding to the weight of an edge between nodes v_m and v_n. The adjacency matrix $\mathbf{A}$ can either be pre-determined or learned in a data-driven fashion. We will describe the latter case in Section II-B.

The traffic prediction problem can be formulated as approximating a function $\mathcal{M}$ which maps P steps of historical traffic data onto the next Q steps of traffic data as follows:

$$\left(\mathbf{x}_{t-P+1}, \ldots, \mathbf{x}_t\right) \xrightarrow{\mathcal{M}(\Theta)} \left(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+Q}\right), \tag{1}$$

where Θ denotes the parameters of the prediction model $\mathcal{M}$. Let C denote the number of samples in the training dataset. We define the temporal traffic tensor $\mathbf{X} \in \mathbb{R}^{C \times N \times P}$ and the output tensor $\mathbf{Y} \in \mathbb{R}^{C \times N \times Q}$, where $\mathbf{X}_{c,n,:}$ denotes the P steps of traffic data of cell v_n from the c-th sample and $\mathbf{Y}_{c,n,:}$ is the associated next Q steps of traffic data.

The parameters Θ can be determined by minimizing the prediction error over all cells in the training dataset. The problem can be formulated as:

$$\underset{\Theta}{\text{argmin}} \ \frac{1}{CN} \sum_{c=1}^{C} \sum_{n=1}^{N} \mathcal{L}\left(\mathcal{M}\left(\mathbf{X}_{c,n,:}; \Theta\right), \mathbf{Y}_{c,n,:}\right), \tag{2}$$

where $\mathcal{L}$ is a pre-defined loss function. In this paper, we consider the L1 loss function, i.e., $\mathcal{L} = \sum_{q=1}^{Q} |\mathbf{Y}'_{c,n,q} - \mathbf{Y}_{c,n,q}|$, where $\mathbf{Y}'_{c,n,q}$ is the output of prediction model $\mathcal{M}$.

B. Construction of the Dependency Graph

The graph-based approaches require a dependency graph to characterize the spatial dependencies between different cells in the first step. Note that the pre-determined graph structures, which rely on either similarity measures or distance functions, cannot infer the spatial correlations between neighbouring and distant cells effectively. To address this issue, we propose a SDGC method to model the spatial dependencies between

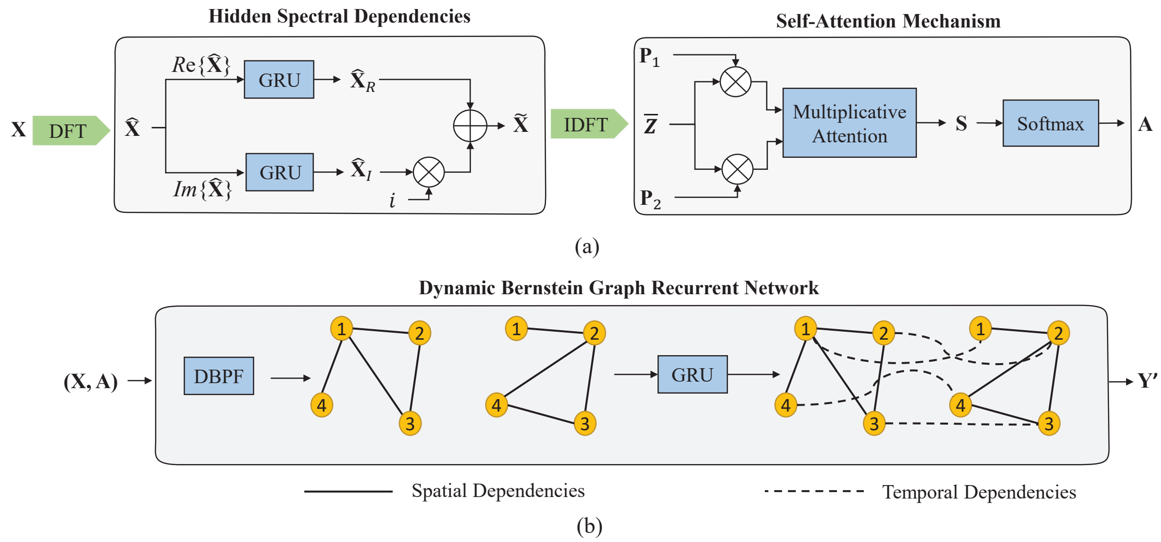


Fig. 2. Illustration of the proposed dynamic Bernstein graph recurrent network (DBGRN). (a) Spectral dynamic graph construction (SDGC): The temporal traffic tensor $\mathbf{X}$ is converted into the frequency domain and the cell dependencies are inferred. The self-attention mechanism determines the weight of edge connections between cells. The output is the adjacency matrix $\mathbf{A}$. (b) Dynamic Bernstein graph recurrent network (DBGRN): Given the temporal traffic tensor $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$, the DBPF module captures the spatial dependencies and learns the cell-specific parameters dynamically. The GRU network extracts the temporal dependencies and predicts the future traffic demands $\mathbf{Y}'$.

neighbouring and distant cells as a graph structure in a data-driven fashion. We use the auto-correlation between different traffic data observations in the spectral domain to construct the dependency graph. First, DFT is used to convert the temporal traffic tensor $\mathbf{X}$ into the frequency domain, where $\hat{\mathbf{X}} \in \mathbb{R}^{C\times N\times P}$ represents the spectral traffic tensor. Given the c-th sample for cell v_n, the DFT operation transforms a vector $\mathbf{X}_{c,n,:}$ with P time steps into another vector $\hat{\mathbf{X}}_{c,n,:}$ with P values as follows:

$$\hat{\mathbf{X}}_{c,n,k} = \sum_{p=1}^{P} \mathbf{X}_{c,n,p} e^{-i\frac{2\pi}{P}kp}, \quad k = 1, \dots, P, \tag{3}$$

where $\hat{\mathbf{X}}_{c,n,k}$ denotes the k-th transformed traffic data volume of the c-th sample for cell $v_n \in \mathcal{V}$ in the frequency domain. The spectral traffic tensor $\hat{\mathbf{X}}_{c,n,k}$ consists of the real and imaginary parts. We have

$$\hat{\mathbf{X}} = Re\{\hat{\mathbf{X}}\} + i\, Im\{\hat{\mathbf{X}}\}. \tag{4}$$

As shown in Fig. 2(a), each part of the spectral traffic tensor $\hat{\mathbf{X}}_{c,n,k}$ is fed into an individual GRU network to learn the hidden spectral dependencies. The outputs $\hat{\mathbf{X}}_R$ and $\hat{\mathbf{X}}_I \in \mathbb{R}^{C\times N\times H}$ represent the hidden state corresponding to each converted data sample, where H denotes the dimension of the hidden state. Subsequently, we construct a new tensor $\tilde{\mathbf{X}} = \hat{\mathbf{X}}_R + i\,\hat{\mathbf{X}}_I$. We then use the inverse discrete Fourier transform (IDFT) to transform $\tilde{\mathbf{X}}$ back to the time domain, where tensor $\mathbf{Z} \in \mathbb{R}^{C\times N\times H}$ represents the transformed tensor. Similar to (3), given the c-th sample for cell v_n, the IDFT operation transforms vector $\tilde{\mathbf{X}}_{c,n,:}$ with H time steps into another vector $\mathbf{Z}_{c,n,:}$ with H values as follows:

$$\mathbf{Z}_{c,n,t} = \frac{1}{H}\sum_{h=1}^{H} \tilde{\mathbf{X}}_{c,n,h} e^{i\frac{2\pi}{H}ht}, \quad t = 1, \dots, H, \tag{5}$$

where $\mathbf{Z}_{c,n,t}$ represents the t-th transformed hidden state value of the c-th sample for cell $v_n \in \mathcal{V}$ in the time domain. We then take the average along the first dimension to determine the hidden state traffic matrix $\bar{\mathbf{Z}} \in \mathbb{R}^{N\times H}$ as follows:

$$\bar{\mathbf{Z}} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{Z}_{c,:,:}. \tag{6}$$

In the next step, to determine the weights of graph edges between cells, we use the self-attention mechanism by initializing two learnable projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2 \in \mathbb{R}^{N\times H}$. The linear projection of the hidden state traffic matrix is defined as follows:

$$\mathbf{Z}_1 = \bar{\mathbf{Z}}\mathbf{P}_1^{\mathrm{T}}, \tag{7a}$$

$$\mathbf{Z}_2 = \bar{\mathbf{Z}}\mathbf{P}_2^{\mathrm{T}}. \tag{7b}$$

We further compute the importance score between each pair of cells by the multiplicative attention as follows:

$$\mathbf{S} = RLReLU\left(\frac{\mathbf{Z}_1\mathbf{Z}_2^{\mathrm{T}}}{\sqrt{H}}\right), \tag{8}$$

where the element $\mathbf{S}_{m,n}$ in matrix $\mathbf{S}$ corresponds to the importance score of cell v_m with respect to cell v_n. The randomized leaky rectified linear unit (RLReLU) function in (8) is defined as follows:

$$RLReLU(u) = \begin{cases} u, & u \geq 0 \\ au, & u < 0 \end{cases}, \tag{9}$$

where a is randomly sampled from uniform distribution $U[\alpha, \beta]$ and $\alpha, \beta \in [0, 1)$. Unlike the rectified linear unit (ReLU) function that keeps only nonnegative values, RLReLU considers both strong and weak connections between cells. This enables the creation of a dense graph with more information. Finally, we normalize the importance score matrix $\mathbf{S}$

by applying a softmax function to obtain the adjacency matrix $\mathbf{A}$ of graph G:

$$\mathbf{A}_{m,n} = softmax(\mathbf{S}_{m,n}) = \frac{e^{\mathbf{S}_{m,n}}}{\sum_{j=1}^{N} e^{\mathbf{S}_{m,j}}}. \quad (10)$$

The projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ as well as the weights of the GRU networks are updated dynamically through time via backpropagation algorithm during the learning process. Thus, we can obtain the adjacency matrix of graph G without any prior knowledge.

C. Spatial Dependencies Between Cells

To capture the spatial dependencies between cells, graph convolutional operation can be used within the context of graph-based approaches. Let $\mathbf{D}$ denote the diagonal degree matrix with the (m, m)-th element equals to the sum of the m-th row of the adjacency matrix $\mathbf{A}$. That is, $\mathbf{D}_{m,m} = \sum_n \mathbf{A}_{m,n}$. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The normalized graph Laplacian matrix can be obtained as $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \in \mathbb{R}^{N\times N}$. To perform the graph convolutional operation, the eigenvectors of the normalized graph Laplacian matrix $\tilde{\mathbf{L}}$ are used as a set of bases to transform the graph data into the frequency domain. However, performing eigenvalue decomposition on the normalized graph Laplacian matrix $\tilde{\mathbf{L}}$ to obtain the eigenvectors has a computational complexity of $O(N^3)$ [9]. This high cost limits the use of graph convolutional operation for traffic prediction in practical wireless systems with a large number of cells.

Some recent works have considered the use of graph spectral filtering with polynomials, namely Chebyshev polynomial, in traffic forecasting problem. This approach can reduce the computational complexity to $O(N^2)$. The GCN architecture in [9] exploits the first-order Chebyshev polynomial, which approximates the graph convolutional operation as a polynomial filtering scheme to capture the spatial dependencies. However, the GCN architecture only performs as a low-pass filter. It may not be able to capture complex spatial dependencies. Due to the dynamic nature of wireless traffic patterns, other filter structures, such as narrow-band and comb filters, are required. To tackle this issue, we use the K-order Bernstein polynomial approximation to capture the spatial dependencies. The K-order approximation of the Bernstein polynomial is capable of learning arbitrary spectral filters, including the band-rejection and comb filters. Based on the K-order approximation of Bernstein polynomial in [13], we propose a DBPF module to capture the spatial dependencies and infer the cell-specific parameters dynamically. Let $\mathbf{O} \in \mathbb{R}^{C\times N\times Q}$ denote the output of the DBPF module. The output of the DBPF module can be determined as follows:

$$\mathbf{O} = \mathbf{B}(\mathbf{W}, K) * \mathbf{X} = \sum_{k=0}^{K} \frac{1}{2^K} \binom{K}{k} (\mathbf{E}_1 \otimes \mathbf{E}_2), \quad (11)$$

where $\mathbf{E}_1 = (2\mathbf{I}_N - \mathbf{L})^{K-k}\mathbf{L}^k$ and $\mathbf{E}_2 = \mathbf{X} \otimes \mathbf{W}_{:,k+1,:,:}$. The elements in tensor $\mathbf{W} \in \mathbb{R}^{N\times(K+1)\times P\times Q}$ are the learnable filter weights, and $\mathbf{W}_{:,k+1,:,:} \in \mathbb{R}^{N\times P\times Q}$ denotes the $(k+1)$-th tensor along the second dimension.

To learn the filter weights, we assign a particular set of parameters $\mathbf{W}_{n,:,:,:} \in \mathbb{R}^{(K+1)\times P\times Q}$ to each cell v_n. The dynamic patterns of all traffic data can be learned by sharing these parameters among all cells. This process enables each cell to learn its specific patterns from a set of shared parameters dynamically over time. However, the number of cells N can be large in practical wireless systems. Thus, $\mathbf{W} \in \mathbb{R}^{N\times(K+1)\times P\times Q}$ would have a large number of parameters to train, which may lead to the overfitting problem. To address this issue, we use the matrix factorization technique, which yields a smaller set of parameters to learn. In particular, we use the same projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ in (7a) and (7b) to generate tensor $\mathbf{W}$ using tensordot operation. That is, $\mathbf{W} = (\mathbf{P}_1 \odot \mathbf{P}_2) \otimes \mathbf{F}$, where $\mathbf{F} \in \mathbb{R}^{H\times(K+1)\times P\times Q}$ is a learnable tensor and H is the same hidden dimension as in (5). Since $H << N$, tensor $\mathbf{F}$ with a smaller set of parameters can be trained during the learning process. Compared with the approach in [7], which uses the enhanced first-order Chebyshev polynomial approximation, our proposed DBPF module captures the complex spatial dependencies and learns the dynamic cell-specific patterns by fine-tuning the approximation order.

D. Spatial-Temporal Traffic Prediction

To further extract the temporal dependencies, we use GRU which can learn the long-term dependencies among sequential data. As shown in Fig. 2(b), we integrate the proposed DBPF module with the GRU to obtain the DBGRN framework, which can capture both the spatial and temporal dependencies simultaneously. In time step $t \in \{1, \ldots, P\}$, we have:

$$\mathbf{P} = \mathbf{P}_1 \odot \mathbf{P}_2, \quad (12a)$$

$$\mathbf{z}_t = \sigma\left(\mathbf{B}\left(\mathbf{P} \otimes \mathbf{F}_z, K\right) * [\mathbf{X}_{c,:,t}, \mathbf{h}_{t-1}] + \mathbf{P}\mathbf{V}_z\right), \quad (12b)$$

$$\mathbf{r}_t = \sigma\left(\mathbf{B}\left(\mathbf{P} \otimes \mathbf{F}_r, K\right) * [\mathbf{X}_{c,:,t}, \mathbf{h}_{t-1}] + \mathbf{P}\mathbf{V}_r\right), \quad (12c)$$

$$\hat{\mathbf{h}}_t = \tanh\left(\mathbf{B}\left(\mathbf{P} \otimes \mathbf{F}_{\hat{h}}, K\right) * [\mathbf{X}_{c,:,t}, \mathbf{r}_t \odot \mathbf{h}_{t-1}] + \mathbf{P}\mathbf{V}_{\hat{h}}\right), \quad (12d)$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \hat{\mathbf{h}}_t, \quad (12e)$$

where $\mathbf{X}_{c,:,t}$ and $\mathbf{h}_t$ are the input and output, respectively. The variables $\mathbf{r}_t$ and $\mathbf{z}_t$ are the reset gate and update gate, respectively. $\sigma(.)$ is the sigmoid function and $\tanh(.)$ is the hyperbolic tangent function. $\mathbf{P}_1, \mathbf{P}_2, \mathbf{F}_z, \mathbf{F}_r, \mathbf{F}_{\hat{h}}$, as well as $\mathbf{V}_z, \mathbf{V}_r$, and $\mathbf{V}_{\hat{h}} \in \mathbb{R}^{H\times Q}$ are the learnable parameters of the model, which can be trained via backpropagation algorithm through time. For the purpose of multi-step prediction (i.e, $Q > 1$), we stack multiple layers to learn the long-term spatial-temporal patterns, and use linear projection to predict the next Q steps of wireless traffic data.

III. Performance Evaluation and Comparison

We apply our proposed model to a real-world dataset representing the mobile traffic volume provided by Telecom Italia [1] in the city of Milan. The Milan area is divided into $10,000$ cells. In the dataset, three types of cellular traffic services, namely: short message service (SMS), call service, and Internet service, are collected from Nov. 1, 2013 to Jan.

1, 2014 in 10-minute intervals. Since many cells have zero traffic volume in the 10-minute time intervals, we aggregate the traffic into hourly intervals. We use the mean absolute error (MAE) and root mean squared error (RMSE) as metrics to evaluate the performance of the prediction models.

We compare the performance of our proposed prediction model with the following baseline models:

- LSTM [4]: The LSTM model in [4] uses multiple LSTM units for mobile traffic prediction.
- Spectral-temporal graph neural network (StemGNN) [8]: The StemGNN model is designed for multivariate time-series forecasting. It uses the graph Fourier transform, DFT and deep neural networks to capture inter-series correlations and intra-series dependencies jointly in the spectral domain.
- Multi-view spatial-temporal graph network (MVSTGN) [12]: The MVSTGN combines multi-head attention and convolution mechanisms, to learn the spatial-temporal characteristics of the wireless data.
- Adaptive graph convolutional recurrent network (AGCRN) [7]: The AGCRN model combines the enhanced GCN architecture with GRU to capture the spatial-temporal dependencies.

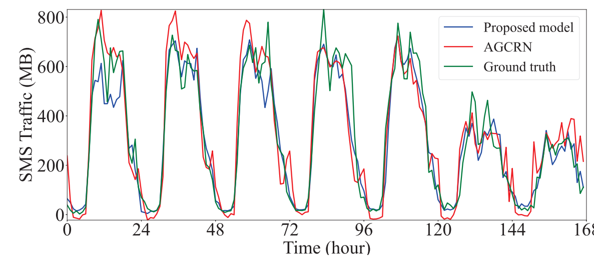
Without loss of generality, we randomly select 700 cells and conduct experiments on three types of traffic services. The proposed DBGRN framework has two layers. Each layer has 128 hidden neurons. We choose $\alpha = 0.1$ and $\beta = 0.4$ in the RLReLU function. The hidden state dimension H is set to 2. We use Adam optimizer to update our model with a learning rate of 0.025. We choose $P = 5$ and $Q = 1$. We use min-max normalization technique to scale the data to be within the range of $[0, 1]$. After performing all the operations, the values are scaled back to the original ones. For each service type, we use the first six weeks data to train the prediction models. The data in the seventh week is used as the validation set. We use the traffic data from the last week for testing.

1) Performance Comparison: Table I presents the experimental results of different models for three types of traffic services. The results show that our proposed model outperforms all four baseline models for both metrics with a significant margin. The AGCRN baseline [7] is the second best-performing model. In particular, our proposed model can achieve 9.6% (SMS), 7.29% (call) and 9.16% (Internet) improvements in terms of the MAE when compared with AGCRN. Moreover, our proposed model provides 5.55%, 8.33%, and 6.5% improvements on RMSE in SMS, call and Internet services, respectively, when compared with AGCRN.

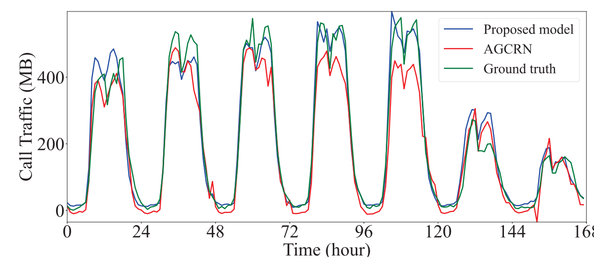
In Fig. 3, we plot the ground truth and predicted values obtained by our proposed model and AGCRN [7] at a randomly selected cell for the duration of one week. The results show that the dynamics of the ground truth traffic curve is well-captured by our proposed model, especially at the peaks and troughs. Our proposed model considers the information in the spatial, temporal, and spectral domains. In particular, the proposed SDGC module uses the spectral information of traffic data to model the spatial dependencies between neighbouring and distant cells as a graph structure. Furthermore, our proposed DBGRN framework is capable of learning both complex spatial correlations and cell-specific traffic patterns.

TABLE I
PREDICTION PERFORMANCE COMPARISONS AMONG DIFFERENT MODELS.

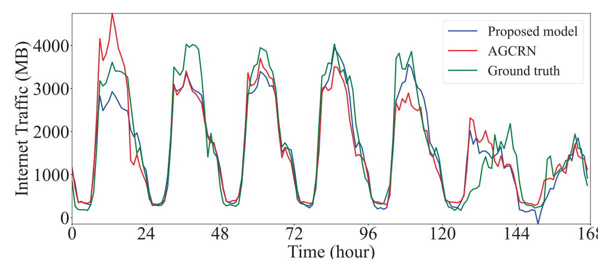
	SMS		Call		Internet	
Models	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	14.57	50.51	15.93	52.53	84.40	222.28
StemGNN	11.03	30.27	11.44	31.50	55.19	162.32
MVSTGN	10.51	32.16	9.47	28.37	46.81	135.97
AGCRN	8.95	25.06	5.90	18.83	43.33	117.0
Our model	**8.09**	**23.67**	**5.47**	**17.26**	**39.36**	**109.39**
$\uparrow (+,-)\%$	+9.60	+5.55	+7.29	+8.33	+9.16	+6.50



(a)



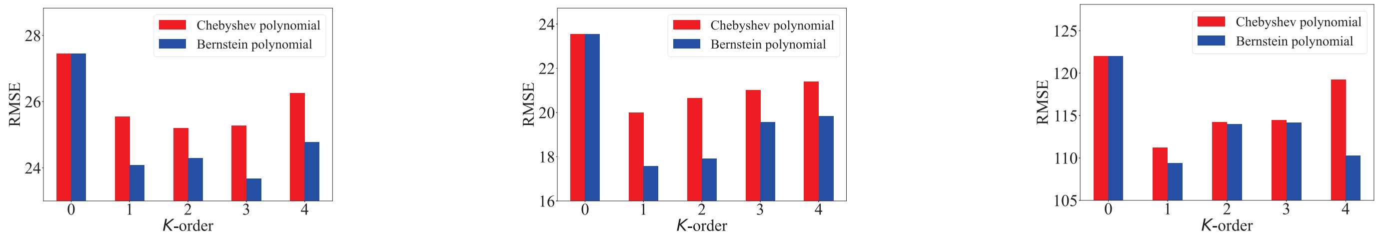
(b)



(c)

Fig. 3. Comparisons of predicted results of our proposed model, AGCRN [7] and the ground truth for (a) SMS service, (b) call service and (c) Internet service at a randomly selected cell for the duration of one week.

2) Effect of the K-order Bernstein Polynomial: The Bernstein polynomial filtering scheme is able to learn arbitrary interpretable spectral filters. It can capture the complex spatial dependencies and learn cell-specific parameters. We now compare the performance of the K-order Bernstein polynomial expansion with the K-order Chebyshev polynomial expansion, which has been used in traffic prediction problems [7], [10]. We use the dynamic filter weights learning approach proposed in our framework for fair comparison. We present the predic-



(a) SMS Service (b) Call Service (c) Internet Service

Fig. 4. The impact of the K-order approximation of the polynomials on the prediction performance of (a) SMS, (b) call and (c) Internet services.

TABLE II
PERFORMANCE COMPARISONS AMONG DIFFERENT GRAPH DEPENDENCY CONSTRUCTION APPROACHES.

Modules	SMS		Call		Internet	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
ATT	8.56	24.32	6.28	18.8	42.43	112.2
Time+ATT	8.23	23.98	5.73	17.98	40.87	110.91
SDGC	**8.09**	**23.67**	**5.47**	**17.26**	**39.36**	**109.39**

tion performance of the Bernstein and Chebyshev polynomial filtering schemes in terms of RMSE in Fig. 4. The results show that the Bernstein polynomial achieves a lower RMSE for all values of K among all three service types when compared with the Chebyshev polynomial. Furthermore, considering the Bernstein polynomial approximation, the results show that the best-performing models for call and Internet traffic services are obtained by the first-order approximation (i.e., $K = 1$). The best-performing model for SMS service is achieved when K is equal to three. This illustrates that the first-order polynomial approximation may not always result in the best-performing model and fine-tuning the polynomial order can improve the prediction performance.

3) Effect of Inferring Spectral Dependencies: Finally, we evaluate the effectiveness of our proposed SDGC method, which uses the traffic data information in the spectral domain for graph dependency construction. For comparison, we consider two other approaches to obtain the dependency graph. In the first approach, we only use an individual self-attention mechanism module. We refer to this as the ATT module. In the second approach, we first use a GRU network to capture the dependencies of the input data $\mathbf{X}$ in the time-domain. It is followed by a self-attention mechanism module. We refer to this as the Time+ATT module. The results in the Table II show that our proposed SDGC module achieves lower RMSE and MAE than the ATT and Time+ATT modules. This illustrates that incorporating the spectral information of traffic data in graph dependency construction can improve the prediction accuracy.

IV. CONCLUSION

In this paper, we investigated the problem of wireless cellular traffic prediction and proposed a dynamic Bernstein graph recurrent network that considers the information in the spatial, temporal, and spectral domains. Our proposed model is able to learn the spatial relationships between neighbouring and distant cells as a dependency graph in a data-driven fashion. It can effectively predict the spatial-temporal traffic demands. We evaluated the performance of our proposed model using a real-world dataset with three service types. The experimental results showed that the proposed model outperforms four state-of-the-art baseline models, and achieves 8% and 10% improvements in terms of the RMSE and MAE, respectively. For future work, we will consider data augmentation techniques to further improve the prediction accuracy.

REFERENCES

[1] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, “A multi-source dataset of urban life in the city of Milan and the Province of Trentino,” *Scientific Data*, vol. 2, p. 150055, Oct. 2015.

[2] L. Yu, M. Li, W. Jin, Y. Guo, Q. Wang, F. Yan, and P. Li, “STEP: A spatio-temporal fine-granular user traffic prediction system for cellular networks,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3453–3466, Dec. 2021.

[3] K. He, X. Chen, Q. Wu, S. Yu, and Z. Zhou, “Graph attention spatial-temporal network with collaborative global-local learning for citywide mobile traffic prediction,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 4, pp. 1244–1256, Apr. 2022.

[4] H. D. Trinh, L. Giupponi, and P. Dini, “Mobile traffic prediction from raw data using LSTM networks,” in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, Sept. 2018.

[5] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, “Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[6] Z. Wang and V. W.S. Wong, “Cellular traffic prediction using deep convolutional neural network with attention mechanism,” in *Proc. of IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2022.

[7] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive graph convolutional recurrent network for traffic forecasting,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Virtual, Dec. 2020.

[8] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, “Spectral temporal graph neural network for multivariate time-series forecasting,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Virtual, Dec. 2020.

[9] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. on Learning Representations (ICLR)*, Toulon, France, Apr. 2017.

[10] S. Zhao *et al.*, “Cellular network traffic prediction incorporating handover: A graph convolutional approach,” in *Proc. IEEE Int. Conf. on Sensing, Commun., and Netw. (SECON)*, Virtual, Jun. 2020.

[11] Y. Fang, S. Ergüt, and P. Patras, “SDGNet: A handover-aware spatiotemporal graph neural network for mobile traffic forecasting,” *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 582–586, Mar. 2022.

[12] Y. Yao, B. Gu, Z. Su, and M. Guizani, “MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction,” accepted for publication in *IEEE Trans. Mobile Comput.*, Nov. 2021.

[13] M. He, Z. Wei, Z. Huang, and H. Xu, “BernNet: Learning arbitrary graph spectral filters via Bernstein approximation,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Virtual, Dec. 2021.