Bayesian Meta-Learning for Adaptive Traffic Prediction in Wireless Networks

Zihuan Wang, *Graduate Student Member, IEEE*, and Vincent W.S. Wong, *Fellow, IEEE*

Abstract—Wireless traffic prediction is indispensable for network planning and resource management. Due to different population distributions and user behavior, there exist strong spatial-temporal variations in wireless traffic across different regions. Most of the conventional traffic prediction approaches can only tackle a particular spatial-temporal pattern and cannot capture such variations in wireless traffic. This motivates us to develop an adaptive approach which can tackle spatial-temporal variations and predict wireless traffic in different regions. In this paper, we formulate an adaptive traffic prediction problem from a probabilistic inference perspective and develop a variational spatial-temporal Bayesian meta-learning (VST-BML) algorithm. We model the traffic prediction in different regions as different prediction tasks. The proposed VST-BML algorithm can learn the common spatial-temporal features shared by all prediction tasks, and adaptively infer the task-specific parameters to tackle spatial-temporal variations. We evaluate the performance of our proposed VST-BML algorithm using a real-world traffic dataset. Experimental results show that the proposed algorithm can quickly adapt to different prediction tasks by using only a small number of data samples and provide accurate traffic prediction in different regions. When compared with five baseline methods, the proposed algorithm can reduce the root-mean-square error (RMSE) and mean absolute error (MAE) by 53.0% and 48.4%, respectively.

Index Terms—Adaptive traffic prediction, Bayesian meta-learning, deep neural networks, spatial-temporal variations.

1 Introduction

1.1 Background

The increasing popularity of smartphones and Internet of things (IoT) devices leads to an explosive growth of wireless data traffic. In order to allocate and utilize network resources efficiently, wireless service providers require accurate results on traffic prediction and forecasting [1]–[3]. By predicting future traffic load, wireless service providers can dynamically allocate network resources and improve the spectral and energy efficiencies. Moreover, proactive measures can be taken to guarantee the diverse quality of service (QoS) requirements of different use cases in the current fifth-generation (5G) and future sixth-generation (6G) wireless networks [4]. Therefore, traffic prediction is important for network planning and optimization, and is becoming an indispensable prerequisite to facilitate the fusion of artificial intelligence (AI) and wireless networking [5]–[7].

Existing approaches for wireless traffic prediction aim to predict the most likely sequences of traffic data in a geographical region given some previous observations. In general, a region is divided into multiple grid cells[1], which have the same size. The traffic prediction is performed on a cell basis [8]–[10]. There exist temporal dependencies in the wireless traffic which can be utilized for prediction. Moreover, user mobility introduces spatial correlations into traffic across neighboring grid cells, which also needs to be taken into consideration when predicting the traffic. Approaches to solving the traffic prediction problem can be classified into two categories: traditional statistical methods and deep learning based algorithms. The algorithms in the first category (e.g., autoregressive integrated moving average (ARIMA) [11]) are usually applied to simple wireless traffic conditions and small datasets. They lack the capability to either tackle high-dimensional time series data or capture the complex spatial-temporal features. On the other hand, deep learning based algorithms have gained increasing attention in recent years and have become state-of-the-art approaches for traffic prediction. The procedures for deep learning based traffic prediction are as follows [8]: (i) Collect sufficient traffic data samples under a certain sampling rate (e.g., every 10 minutes) from the grid cells in a region; (ii) Apply deep learning tools to train a neural network using the dataset obtained in step (i); (iii) Deploy the trained neural network to predict future traffic in this region. In order to obtain accurate predicted results, it is essential for the trained neural network to capture the temporal dependency in traffic data and the spatial correlation among distributed grid cells.

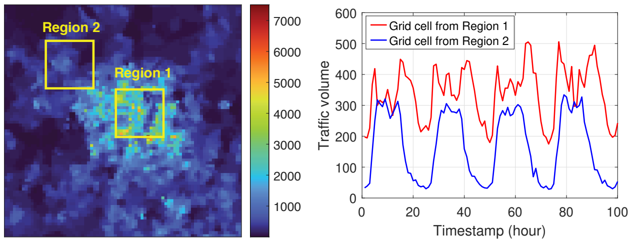
Various deep neural networks have been developed for spatial-temporal modeling and wireless traffic prediction. Recurrent neural networks (RNNs) [12] and long short-term memory (LSTM) networks [8], [13] are proposed for extracting temporal features from time series traffic data. Convolutional neural networks (CNNs) [8], [14], [15] are typically used to capture the spatial dependency of wireless traffic in a region. The convolutional LSTM (ConvLSTM)

Manuscript received on Mar. 31, 2023; revised on Aug. 30, 2023; accepted on Oct. 6, 2023. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and the Digital Research Alliance of Canada (alliancecan.ca). (Corresponding author: Vincent W.S. Wong.)

Z. Wang and V. W.S. Wong are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (e-mail: {zihuanwang, vincentw}@ece.ubc.ca).

Color versions of one or more of the figures in this paper are available online at https://ieeexplore.ieee.org.

[1] In wireless traffic prediction problems, a grid cell may cover multiple cellular base station towers. The wireless traffic in a grid cell is the aggregate traffic from all the cellular base station towers within the cell.



(a)

(b)

Fig. 1. Spatial-temporal patterns of wireless traffic in different regions. (a) Spatial patterns of the traffic; (b) Temporal patterns of the traffic.

network is developed in [16] to analyze both the spatial and temporal dependencies. Graph neural networks (GNNs) can also be applied for traffic prediction. GNNs learn the spatial-temporal dependencies of traffic data via feature propagation and aggregation [17]–[22].

1.2 Motivation

Although the existing works on wireless traffic prediction can capture the spatial-temporal dependencies and predict wireless traffic in a particular region, the prediction strategy learned by those algorithms may not be able to accurately predict traffic in other regions which have different spatial-temporal patterns. We note that understanding the wireless traffic across different regions is important for global network infrastructure planning and deployment, as well as cross-region resource management [3]–[5]. However, it is challenging to predict traffic in different regions due to spatial-temporal variations.

The spatial-temporal pattern can vary significantly in different regions due to different population distributions and user behavior. As an example, in Fig. 1, we use a real-world traffic dataset provided by Telecom Italia [23] to show the spatial-temporal variations across different regions. The data is collected from the city of Milan in Italy. Fig. 1(a) shows the heat map of the traffic volume in Milan. Each pixel corresponds to a grid cell. The brightness of each pixel represents the volume of the wireless traffic in the corresponding grid cell. We select two regions (shown in those two yellow boxes in Fig. 1(a)) to illustrate the spatial variations. It can be observed that the traffic volume in Region 1 (located in the central area) is much higher than that in Region 2 (located in the northwestern area). The traffic has different distributions and spatial patterns in these two regions. The temporal patterns of the wireless traffic are illustrated in Fig. 1(b). We show the traffic volume over 100 hours (i.e., approximately four days) in two grid cells which are selected from Regions 1 and 2, respectively. It can be observed that although the traffic volume exhibits periodicity in the temporal domain, the maximum and minimum values of traffic volume of those two cells in Regions 1 and 2 are different. The grid cell in Region 1 has a higher traffic volume than that in Region 2. During peak hours every day, the traffic curve of the grid cell in Region 1 exhibits more fluctuation than the one in Region 2. The results in Fig. 1(b) indicate that there exist strong temporal variations across different regions.

In order to accurately predict future wireless traffic in different regions that have diverse spatial-temporal patterns, sufficient data samples are required for the training of models, with each model targeting a region with a specific spatial-temporal pattern. However, data collection can be time-consuming and training multiple models requires significant computational resources. Moreover, the distribution of traffic data is unbalanced due to different population distributions and densities. Wireless traffic in urban regions (e.g., downtown) has a large amount of data samples available for training. On the other hand, in rural regions, only limited data samples can be collected and the amount of data samples may not be sufficient for training. This motivates us to develop an adaptive prediction algorithm, which can tackle spatial-temporal variations and adapt to traffic prediction in different regions using only a small number of data samples.

1.3 Contributions

In this paper, we study the adaptive traffic prediction problem and propose a variational spatial-temporal Bayesian meta-learning (VST-BML) algorithm. We model the traffic prediction in different regions as different *prediction tasks*. The proposed VST-BML algorithm learns a set of globally shared parameters, which can extract the common spatial-temporal features shared by all tasks and adaptively determine the task-specific parameters to tackle spatial-temporal variations. The main contributions of this paper are summarized as follows:

- We formulate the adaptive traffic prediction problem from a probabilistic inference perspective based on the latent variable model. We derive the objective function for the optimization of the global parameters by using the evidence lower bound (ELBO). The task-specific parameters are modeled as latent variables conditioned on the global parameters, such that given the optimized global parameters, the task-specific parameters can be adaptively determined.
- We propose a VST-BML algorithm, which learns the global parameters and determines the task-specific parameters through a variational spatial-temporal (VST) network. The VST network consists of an extractor, an amortization network, and a generator. The extractor can capture the shared spatial-temporal features. We propose a dual-attention mechanism to be deployed in the extractor, which enables the network to focus on the most important spatial-temporal features shared by all tasks. The amortization network determines the distribution over task-specific parameters. The generator predicts the traffic given the commonly shared spatial-temporal features and the task-specific parameters.
- We adopt the Bayesian meta-learning (BML) algorithm for the training of the VST network. The proposed VST network is trained based on a distribution of prediction tasks. After training, the VST network can adapt to new prediction tasks which have different spatial-temporal patterns. The use of BML enables the proposed VST network to quickly adapt to different prediction tasks using only a few data samples.
- We evaluate the performance of the proposed VST-BML algorithm based on a real-world wireless traffic dataset and compare it with five baseline meth-

ods. They include ARIMA [11], ConvLSTM network [16], multi-view spatial-temporal graph network (MVSTGN) [18], spatial-temporal cross-domain network (STCNet) [9], and spatial-temporal transformer (ST-Tran) [10]. When compared with the baseline methods, experimental results show that the proposed VST-BML algorithm can reduce the root-mean-square error (RMSE) and mean absolute error (MAE) by 53.0% and 48.4%, respectively.

- For further evaluation of the proposed VST-BML algorithm, we compare the predicted results with the ground truth in different regions which have diverse spatial-temporal patterns in wireless traffic. It is shown that our proposed algorithm can accurately predict wireless traffic under different spatial-temporal patterns by using only five data samples. This demonstrates the fast adaptation capability of our proposed algorithm. We also evaluate the effect of the dual-attention mechanism on the prediction accuracy through a set of ablation experiments. Results demonstrate the capability of the dual-attention mechanism on spatial-temporal feature extraction.

The rest of this paper is organized as follows. In Section 2, we summarize the related work. Section 3 provides a preliminary analysis on spatial-temporal variations in wireless traffic across different regions. Section 4 describes the problem formation for adaptive traffic prediction and presents the proposed VST-BML algorithm. Section 5 illustrates the experimental results for performance evaluation. Finally, conclusions are drawn in Section 6.

2 Related Work

We now summarize some of the recent works on deep learning based wireless traffic prediction algorithms. In [13] and [24], LSTM networks are used to capture the temporal dependency of time series data and predict future traffic load. In [8], an autoencoder is developed for spatial feature extraction and an LSTM network is used to learn the temporal dependency. In [16], a ConvLSTM network is proposed, which replaces the matrix multiplication with convolutional operations in an LSTM cell to extract the spatial and temporal features. In [15], a spatial-temporal network is proposed for traffic prediction. It includes a ConvLSTM and three-dimensional convolutional (Conv3D) layers to capture the spatial-temporal dependencies. In [25], an attention-embedded CNN is developed to learn the local short-term and long-term spatial-temporal dependencies for traffic prediction. In [17], a GNN-based predictive algorithm is proposed which models the spatial-temporal correlations of wireless traffic using a graph representation. In [18], an MVSTGN algorithm is developed which embeds the attention modules into a GNN to capture the global and local spatial-temporal features. In [19], a graph attention spatial-temporal network is proposed. It learns the spatial and temporal features of wireless traffic through a spatial relation graph and an attention-based RNN structure, respectively. In [9], STCNet is proposed which uses cross-domain knowledge (e.g., the information from point of interest distribution) to improve the prediction accuracy. ST-Tran is developed in [10]. It has two transformer blocks for spatial and temporal feature extraction. In [26] and [27], side information (e.g., weather conditions) is used to enhance traffic prediction performance. In [20]–[22], user mobility patterns are adopted to facilitate traffic prediction in wireless networks. Traffic prediction based on distributed training architecture is investigated in [28], where a hierarchical aggregation structure is introduced for local training and central aggregation.

The aforementioned works study spatial-temporal dependencies for traffic prediction in a particular region. Those works do not consider the spatial-temporal variations and the trained models cannot be generalized to different regions. Although multiple models can be trained with each model targeting a region with a specific spatial-temporal pattern, data collection can be consuming and network training is resource-demanding. To address the above issues, in this paper, we propose a VST-BML algorithm, which can quickly adapt to traffic prediction in different regions using a small number of data samples.

3 Preliminary Analysis on Spatial-Temporal Variations in Wireless Traffic

In Section 1, we briefly explain the spatial-temporal variations in wireless traffic and show the spatial-temporal patterns in different regions. To gain more insights into spatial-temporal variations, in this section, we first provide a more detailed investigation of the spatial-temporal dependencies in wireless traffic and show how this dependency can vary across different regions. The following data analysis of wireless traffic is based on a real-world dataset [23] provided by Telecom Italia. The traffic data was collected from Nov. 1, 2013 to Jan. 1, 2014 in Milan, Italy, with a sampling rate of 10 minutes. The area of Milan city is divided into 100×100 grid cells, with each grid cell covering an area of 235×235 m^2. A region is defined as an area which contains a group of grid cells. The dataset includes the call detail records (CDRs) in Milan. We use the traffic volume of voice call to analyze the spatial-temporal variations across different regions. Similar to the work in [9], we aggregate the traffic data into hourly scale. A timestamp is used to indicate the time when wireless traffic was collected. The interval between two consecutive timestamps is one hour.

Each region has $M \times N$ grid cells. Let $\mathcal{R}$ denote the set of regions. Consider region $i \in \mathcal{R}$, the wireless traffic across the $M \times N$ grid cells at the t-th timestamp can be represented as a matrix $\mathbf{D}_{i,t} \in \mathbb{R}^{M \times N}$:

$$\mathbf{D}_{i,t} = \begin{bmatrix} d_{i,t}^{(1,1)} & d_{i,t}^{(1,2)} & \cdots d_{i,t}^{(1,N)} \\ \vdots & \vdots & \vdots \\ d_{i,t}^{(M,1)} & d_{i,t}^{(M,2)} & \cdots d_{i,t}^{(M,N)} \end{bmatrix}, \tag{1}$$

where $d_{i,t}^{(m,n)}$ denotes the traffic volume in grid cell (m, n) of region i at the t-th timestamp. Let vector $\mathbf{d}_i^{(m,n)} = (d_{i,1}^{(m,n)}, \ldots, d_{i,T}^{(m,n)})$ denote a sequence of traffic data with T timestamps. We use a three-dimensional (3D) tensor $\mathbf{D}_i = [\mathbf{D}_{i,1}, \ldots, \mathbf{D}_{i,T}] \in \mathbb{R}^{T \times M \times N}$ to represent the T-timestamp traffic volume across $M \times N$ cells in region i.

The wireless traffic in each region has a specific spatial-temporal dependency. In particular, there exists temporal

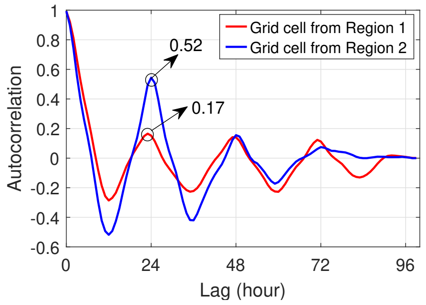


Fig. 2. Temporal autocorrelations of wireless traffic in two grid cells selected from different regions in Milan. Region 1 is in the central area. Region 2 is in the northwestern area.

autocorrelation in the traffic sequence $\mathbf{d}_i^{(m,n)}$ for each grid cell (m,n) in region i. Due to user mobility, there also exists spatial correlation among neighboring grid cells. Moreover, the temporal autocorrelation and spatial correlation may vary significantly in different regions, which indicates strong spatial-temporal variations. In the following subsections, we consider two different regions in the city of Milan, i.e., Region 1 in the central area and Region 2 in the northwestern area, to illustrate the spatial-temporal variations.

3.1 Analysis on Temporal Variations

The sample autocorrelation function [29], as a function of time lag l, is widely used for the evaluation of temporal dependency. The autocorrelation function for a T-sequence of traffic data $\mathbf{d}_i^{(m,n)}$ in grid cell (m,n) of region i is given by:

$$\gamma_i^{(m,n)}(l) = \frac{\sum_{t=1}^{T-l}(d_{i,t}^{(m,n)} - \bar{d}_i^{(m,n)})(d_{i,t+l}^{(m,n)} - \bar{d}_i^{(m,n)})}{\sum_{t=1}^{T}(d_{i,t}^{(m,n)} - \bar{d}_i^{(m,n)})^2}, \quad 0 \le l < T, \tag{2}$$

where $\bar{d}_i^{(m,n)} = \frac{\sum_{t=1}^{T} d_{i,t}^{(m,n)}}{T}$ represents the mean value of the traffic in grid cell (m,n) of region i.

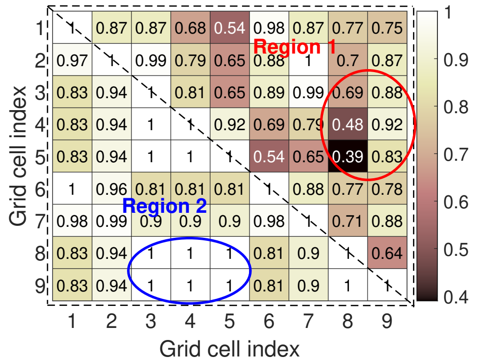
In Fig. 2, we show the temporal autocorrelations of wireless traffic in two grid cells selected from Regions 1 and 2. From Fig. 2, we can observe that the wireless traffic is temporally autocorrelated in both cells. However, the traffic in the cell from Region 1 has a similar autocorrelation with the traffic in the next 24, 48, and 72 hours. That is, when the time lag l is equal to 24, 48, and 72, the autocorrelation values are all between 0.15 and 0.17. On the other hand, the traffic in the cell from Region 2 has a much higher autocorrelation when the time lag l is equal to 24 hours (the autocorrelation is equal to 0.52). This indicates that the wireless traffic in these two cells has different temporal dependencies. The reasons for the temporal variations can be due to heterogeneous user behavior, which influences the trend of wireless traffic in the temporal domain.

3.2 Analysis on Spatial Variations

We use the Pearson correlation coefficient [30] to model the spatial correlation between two neighboring grid cells in a region. A higher Pearson correlation indicates a stronger spatial dependency. The Pearson correlation coefficient for two grid cells (m,n) and (m',n') in region i is defined as

$$\rho_i = \frac{cov(\mathbf{d}_i^{(m,n)}, \mathbf{d}_i^{(m',n')})}{\sigma_{\mathbf{d}_i^{(m,n)}}\sigma_{\mathbf{d}_i^{(m',n')}}}, \tag{3}$$

where $cov(\cdot)$ represents the covariance operation, and σ is the standard deviation. To investigate the variation in spatial correlations, we select two different regions from the dataset. Fig. 3 shows the heat map of the Pearson correlations across the cell groups from Regions 1 and 2, which correspond to the upper and lower triangular parts, respectively. Both cell groups have the same size of coverage area, i.e., 705×705 m^2. The cells in each region are indexed from 1 to 9 for simplicity. Both cells follow the same index order. For example, cells 1 and 2 in Region 1 are adjacent to each other, so are cells 1 and 2 in Region 2. From Fig. 3, we can observe the spatial correlations across the group of grid cells within each region. Although the cell groups in both regions have the same spatial relationships, the spatial correlations across grid cells in Region 2 are higher than those in Region 1. In particular, the Pearson correlation coefficients in Region 2 are all above 0.8. Grid cells 8 and 9 are highly correlated with grid cells $3-5$ (the Pearson correlation coefficient is 1.0). On the other hand, in Region 1, the spatial correlations between grid cells $8, 9$ and $3-5$ are much lower. Grid cells 5 and 8 in Region 1 have the lowest Pearson correlation coefficient, which is 0.39.



	1	2	3	4	5	6	7	8	9
1	1	0.87	0.87	0.68	0.54	0.98	0.87	0.77	0.75
2	0.97	1	0.99	0.79	0.65	0.88	1	0.7	0.87
3	0.83	0.94	1	0.81	0.65	0.89	0.99	0.69	0.88
4	0.83	0.94	1	1	0.92	0.69	0.79	0.48	0.92
5	0.83	0.94	1	1	1	0.54	0.65	0.39	0.83
6	1	0.96	0.81	0.81	0.81	1	0.88	0.77	0.78
7	0.98	0.99	0.9	0.9	0.9	0.98	1	0.71	0.88
8	0.83	0.94	1	1	1	0.81	0.9	1	0.64
9	0.83	0.94	1	1	1	0.81	0.9	1	1

Fig. 3. Spatial correlations of wireless traffic in two different regions in Milan. Region 1 is in the central area. Region 2 is in the northwestern area.

In summary, results from Figs. 2 and 3 show that wireless traffic is highly spatial-temporal dependent and has a specific spatial-temporal pattern in a region. However, the dependency can vary significantly across different regions, which makes accurate traffic prediction in different regions a challenging problem. In the next section, we develop a VST-BML algorithm to tackle the spatial-temporal variations and solve the adaptive traffic prediction problem.

4 Proposed VST-BML Algorithm for Adaptive Traffic Prediction

In this section, we propose a VST-BML algorithm to address the spatial-temporal variations in different regions. We first introduce the adaptive traffic prediction model and formulate the problem from a probabilistic inference perspective. We then present the VST-BML algorithm which can adaptively predict wireless traffic in different regions.

4.1 Adaptive Traffic Prediction Model

The goal of traffic prediction in a particular region is to predict the traffic volume of the next Q timestamps using the previous P traffic observations. This can be achieved by using deep learning techniques, which train a neural network and obtain the network parameters using sufficient pairs of P-timestamp observations (i.e., network input) and Q-timestamp ground truth (i.e., label) in a region. The trained network takes new traffic observations as input and returns the corresponding predictions (for the same region) as output. However, such networks may not be able to provide accurate predictions in other regions due to spatial-temporal variations. When considering traffic prediction in different regions, we aim to develop an adaptive traffic prediction network. Given the traffic observations from any region $i \in \mathcal{R}$, the adaptive network can provide the corresponding traffic prediction results in region i.

4.1.1 Prediction Task, Support Set, and Query Set

We model traffic prediction in different regions with diverse spatial-temporal patterns as different prediction tasks. We use $\tau_i \sim \gamma(\mathcal{T})$ to denote a prediction task in region i, where $\gamma(\mathcal{T})$ represents the distribution of prediction tasks. Let $\mathcal{D}_i$ denote the dataset for task τ_i, which contains multiple input-label pairs, i.e., P-timestamp observations (input) and Q-timestamp ground truth (label) in region i. The dataset $\mathcal{D}_i$ is further partitioned into a support set $\mathcal{D}_i^{\mathrm{s}}$ and a query set $\mathcal{D}_i^{\mathrm{q}}$, where $\mathcal{D}_i^{\mathrm{s}} \cup \mathcal{D}_i^{\mathrm{q}} = \mathcal{D}_i$ and $\mathcal{D}_i^{\mathrm{s}} \cap \mathcal{D}_i^{\mathrm{q}} = \emptyset$. Given an arbitrary timestamp t, we use tensor $\mathbf{X}_i^{(t)} = \{\mathbf{X}_{i,t-P+1}, \ldots, \mathbf{X}_{i,t}\} \in \mathbb{R}^{P \times M \times N}$ to denote the P-timestamp observations and tensor $\mathbf{Y}_i^{(t)} = \{\mathbf{Y}_{i,t+1}, \ldots, \mathbf{Y}_{i,t+Q}\} \in \mathbb{R}^{Q \times M \times N}$ as the Q-timestamp ground truth in support set $\mathcal{D}_i^{\mathrm{s}}$, where $\{\mathbf{X}_{i,p}\}_{p=t-P+1}^{t}$ and $\{\mathbf{Y}_{i,q}\}_{q=t+1}^{t+Q}$ have a similar form as in (1). Similarly, tensors $\tilde{\mathbf{X}}_i^{(t)} = \{\tilde{\mathbf{X}}_{i,t-P+1}, \ldots, \tilde{\mathbf{X}}_{i,t}\} \in \mathbb{R}^{P \times M \times N}$ and $\tilde{\mathbf{Y}}_i^{(t)} = \{\tilde{\mathbf{Y}}_{i,t+1}, \ldots, \tilde{\mathbf{Y}}_{i,t+Q}\} \in \mathbb{R}^{Q \times M \times N}$ denote the observations and ground truth in query set $\mathcal{D}_i^{\mathrm{q}}$, respectively. We consider the support set $\mathcal{D}_i^{\mathrm{s}}$ contains N_{s} pairs of $\{\mathbf{X}_i^{(t_s)}\}_{s=1}^{N_{\mathrm{s}}}$ and $\{\mathbf{Y}_i^{(t_s)}\}_{s=1}^{N_{\mathrm{s}}}$, which correspond to the observations and ground truth of wireless traffic at timestamp t_s, for $s = 1, \ldots, N_{\mathrm{s}}$. The query set $\mathcal{D}_i^{\mathrm{q}}$ contains N_{q} pairs of data samples $\{\tilde{\mathbf{X}}_i^{(t_q)}, \tilde{\mathbf{Y}}_i^{(t_q)}\}_{q=1}^{N_{\mathrm{q}}}$, which are the observations and ground truth of traffic at timestamp t_q (different from those timestamps in support set), for $q = 1, \ldots, N_{\mathrm{q}}$.

We aim to develop an adaptive network parameterized by a set of global parameters θ, such that given the data samples in support set $\mathcal{D}_i^{\mathrm{s}}$, the network parameterized by θ can adaptively determine a set of task-specific parameters ϕ_i which are used for traffic prediction on the disjoint query set $\mathcal{D}_i^{\mathrm{q}}$ for task τ_i. Note that only a small number of data samples are available in the support set [31]. This is due to the fact that in real-world wireless systems, there may be limited traffic data available in some particular regions. Moreover, collecting a large amount of data samples along with data processing and computation is time-consuming and may lower the efficiency of network adaptations to different prediction tasks. In order to effectively utilize the limited data samples in support set $\mathcal{D}_i^{\mathrm{s}}$ and achieve fast adaptation to prediction task τ_i, in the following, we formulate the adaptive traffic prediction problem from a probabilistic inference perspective by using the latent variable model.

4.1.2 Preliminaries of Latent Variable Model

In this subsection, we present the preliminaries of the latent variable model. Let $\mathbf{x}$ denote the observed wireless traffic data. We are interested in obtaining the distribution $p(\mathbf{x})$ of the observed data. By using the latent variable model, the observed data $\mathbf{x}$ is determined by a latent distribution $p(\mathbf{z})$, where $\mathbf{z}$ represents the latent variables. The data $\mathbf{x}$ is generated by a conditional distribution $p(\mathbf{x} \mid \mathbf{z})$. The goal in the latent variable model is to determine the posterior distribution of latent variables $\mathbf{z}$, i.e., $p(\mathbf{z} \mid \mathbf{x})$, given the observed data $\mathbf{x}$, which can be expressed as:

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}}. \tag{4}$$

However, the posterior distribution is difficult to calculate as the integral in the denominator is high dimensional. A general solution is to approximate the posterior distribution by another distribution $q_\xi(\mathbf{z})$ characterized by parameters ξ [32]. The closeness of these two distributions is measured by the evidence lower bound (ELBO). Maximizing the ELBO ensures $q_\xi(\mathbf{z})$ to approach the posterior distribution $p(\mathbf{z} \mid \mathbf{x})$. The ELBO $\mathcal{L}(\xi)$ is defined as [32]:

$$\mathcal{L}(\xi) = \mathbb{E}_{\mathbf{z} \sim q_\xi(\mathbf{z})}[\log p(\mathbf{x} \mid \mathbf{z})] - D_{KL}(q_\xi(\mathbf{z}) \mid\mid p(\mathbf{z})), \tag{5}$$

where D_{KL} corresponds to the Kullback-Leibler (KL) divergence between distributions $q_\xi(\cdot)$ and $p(\cdot)$. Minimizing the KL divergence encourages the distributions $q_\xi(\cdot)$ and $p(\cdot)$ to be similar.

4.1.3 Problem Formulation for Adaptive Traffic Prediction

Based on the latent variable model described above, we now present the problem formulation and derive the objective function for adaptive traffic prediction. Recall that the goal is to learn a set of global parameters θ of the adaptive network. The adaptive network can determine the task-specific parameters ϕ_i for traffic prediction on the query set $\mathcal{D}_i^{\mathrm{q}}$ of task τ_i. We model the task-specific parameters ϕ_i as latent variables and model the query set $\mathcal{D}_i^{\mathrm{q}}$ as the observed data. For each task $\tau_i \sim \gamma(\mathcal{T})$, the posterior distribution $p(\phi_i \mid \mathcal{D}_i^{\mathrm{q}})$ is approximated by another distribution $q_{\xi_i}(\phi_i)$ characterized by the parameters ξ_i. The ELBO for task τ_i can be expressed as follows:

$$\begin{aligned} \mathcal{L}(\xi_i) = \mathbb{E}_{\phi_i \sim q_{\xi_i}(\phi_i)}&[\log p(\mathcal{D}_i^{\mathrm{q}} \mid \phi_i)] \\ &- D_{KL}(q_{\xi_i}(\phi_i) \mid\mid p(\phi_i)),\ \tau_i \sim \gamma(\mathcal{T}). \end{aligned} \tag{6}$$

When taking the global parameters θ into consideration, the generation of the task-specific parameters ϕ_i and the prediction on the query set $\mathcal{D}_i^{\mathrm{q}}$ are conditioned on θ. Therefore, the latent distribution of ϕ_i is determined by $p(\phi_i \mid \theta)$. The posterior distribution over ϕ_i is approximated by the function $q_{\xi_i}(\phi_i \mid \theta)$ for task $\tau_i \sim \gamma(\mathcal{T})$. The conditional distribution of the observed data (i.e., query dataset $\mathcal{D}_i^{\mathrm{q}}$) is given by $p(\mathcal{D}_i^{\mathrm{q}} \mid \phi_i, \theta)$. Note that the query set $\mathcal{D}_i^{\mathrm{q}}$ is constructed by N_{q} input-label pairs, i.e., $\{\tilde{\mathbf{X}}_i^{(t_q)}, \tilde{\mathbf{Y}}_i^{(t_q)}\}_{q=1}^{N_{\mathrm{q}}}$.

We aim to predict the ground truth $\tilde{\mathbf{Y}}_i^{(t_q)}$ given the observations $\tilde{\mathbf{X}}_i^{(t_q)}$ as input. Under this setting, the conditional distribution of the observed data can be rewritten as $p(\tilde{\mathbf{Y}}_i^{(t_q)} \mid \tilde{\mathbf{X}}_i^{(t_q)}, \phi_i, \theta)$. Then, the ELBO for task τ_i can be rewritten as

$$\mathcal{L}(\theta, \xi_i) = \mathbb{E}_{\phi_i \sim q_{\xi_i}(\phi_i \mid \theta)}[\log p(\tilde{\mathbf{Y}}_i^{(t_q)} \mid \tilde{\mathbf{X}}_i^{(t_q)}, \phi_i, \theta)] \\ -D_{KL}(q_{\xi_i}(\phi_i \mid \theta) \mid\mid p(\phi_i \mid \theta)),\ \tau_i \sim \gamma(\mathcal{T}). \quad (7)$$

As can be observed in (7), the generation of task-specific parameters ϕ_i is dependent on a set of parameters ξ_i. The computational complexity grows linearly with the number of tasks. To reduce the complexity, we use the information provided by the support set $\mathcal{D}_i^{\mathrm{s}}$ to facilitate the generation of task-specific parameters ϕ_i. Instead of using ξ_i to approximate the posterior distribution over the task-specific parameters $q_{\xi_i}(\phi_i \mid \theta)$ for each task $\tau_i \sim \gamma(\mathcal{T})$, we apply another function q_λ parameterized by λ to map ϕ_i from the support set $\mathcal{D}_i^{\mathrm{s}}$. We approximate the posterior distribution using $q_\lambda(\phi_i \mid \mathcal{D}_i^{\mathrm{s}}, \theta)$, such that the computational cost of this approximation process can be amortized across tasks. Based on this operation, the ELBO for task τ_i can be expressed as follows:

$$\mathcal{L}(\theta, \lambda) = \mathbb{E}_{\phi_i \sim q_\lambda(\phi_i \mid \mathcal{D}_i^{\mathrm{s}}, \theta)}[\log p(\tilde{\mathbf{Y}}_i^{(t_q)} \mid \tilde{\mathbf{X}}_i^{(t_q)}, \phi_i, \theta)] \\ -D_{KL}(q_\lambda(\phi_i \mid \mathcal{D}_i^{\mathrm{s}}, \theta) \mid\mid p(\phi_i \mid \theta)),\ \tau_i \sim \gamma(\mathcal{T}). \quad (8)$$

In (8), $p(\tilde{\mathbf{Y}}_i^{(t_q)} \mid \tilde{\mathbf{X}}_i^{(t_q)}, \phi_i, \theta)$ is the conditional distribution over $\tilde{\mathbf{Y}}_i^{(t_q)}$, given the input data samples $\tilde{\mathbf{X}}_i^{(t_q)}$, task-specific parameters ϕ_i, and the global parameters θ. $q_\lambda(\phi_i \,|\, \mathcal{D}_i^{\mathrm{s}}, \theta)$ produces the variational distribution of the task-specific parameters ϕ_i, given the support set $\mathcal{D}_i^{\mathrm{s}}$ and global parameters θ. Using the information provided by the global parameters θ, the prior over the task-specific parameters $p(\phi_i \,|\, \theta)$ learns the mean and standard deviation of ϕ_i for any task $\tau_i \sim \gamma(\mathcal{T})$. Note that equation (8) includes an expectation with respect to ϕ_i that is sampled from $q_\lambda(\cdot)$. The derivative of $\mathbb{E}_{\phi_i \sim q_\lambda(\cdot)}$ is difficult to calculate since the process of sampling from a distribution is not differentiable and cannot be backpropagated. This issue can be addressed through reparameterization [33], which represents function $q_\lambda(\cdot)$ in a differentiable form as:

$$q_\lambda(\cdot) = \mu_{q_\lambda}(\cdot) + \sigma_{q_\lambda}(\cdot)\epsilon. \quad (9)$$

In (9), the output of $q_\lambda(\cdot)$ is reparameterized by the mean $\mu_{q_\lambda}(\cdot)$ and standard deviation $\sigma_{q_\lambda}(\cdot)$. Random variable ϵ denotes the Gaussian distributed noise with zero mean and unit variance. By using reparameterization, equation (8) can be rewritten as:

$$\tilde{\mathcal{L}}(\theta, \lambda) = \\ \frac{1}{K}\sum_{k=1}^{K} \log p\Big(\tilde{\mathbf{Y}}_i^{(t_q)} \,|\, \tilde{\mathbf{X}}_i^{(t_q)}, \mu_{q_\lambda}(\mathcal{D}_i^{\mathrm{s}}, \theta) + \sigma_{q_\lambda}(\mathcal{D}_i^{\mathrm{s}}, \theta)\epsilon^{(k)}, \theta\Big) \\ -D_{KL}(q_\lambda(\phi_i \,|\, \mathcal{D}_i^{\mathrm{s}}, \theta) \,||\, p(\phi_i \,|\, \theta)),\ \tau_i \sim \gamma(\mathcal{T}), \quad (10)$$

where K is the number of Monte Carlo samples. The first term on the right-hand side in (10) measures the accuracy of the prediction results compared with the ground truth $\tilde{\mathbf{Y}}_i^{(t_q)}$. The second term serves as a KL regularization, which ensures the approximation of the posterior distribution to be close to the true distribution. In this work, we aim to maximize ELBO in (10) across all prediction tasks:

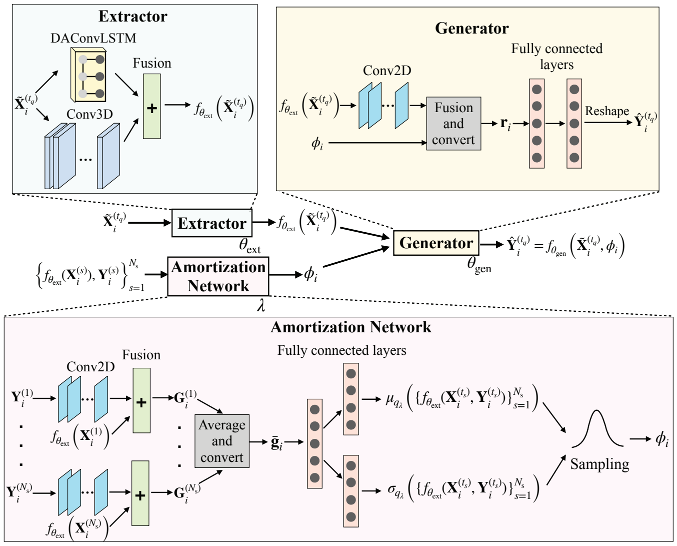

Extractor
DAConvLSTM
Fusion
Conv3D
$\tilde{\mathbf{X}}_i^{(t_q)}$
$f_{\theta_{\mathrm{ext}}}\left(\tilde{\mathbf{X}}_i^{(t_q)}\right)$
Generator
Conv2D
Fully connected layers
Fusion and convert
$\mathbf{r}_i$
ϕ_i
Reshape
$\hat{\mathbf{Y}}_i^{(t_q)}$
Extractor
θ_{ext}
Generator
θ_{gen}
$\hat{\mathbf{Y}}_i^{(t_q)} = f_{\theta_{\mathrm{gen}}}\left(\tilde{\mathbf{X}}_i^{(t_q)}, \phi_i\right)$
$\left\{f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(s)}), \mathbf{Y}_i^{(s)}\right\}_{s=1}^{N_{\mathrm{s}}}$
Amortization Network
λ
Amortization Network
Conv2D
Fusion
Fully connected layers
$\mathbf{Y}_i^{(1)}$
$f_{\theta_{\mathrm{ext}}}\left(\mathbf{X}_i^{(1)}\right)$
$\mathbf{G}_i^{(1)}$
$\mathbf{Y}_i^{(N_{\mathrm{s}})}$
$f_{\theta_{\mathrm{ext}}}\left(\mathbf{X}_i^{(N_{\mathrm{s}})}\right)$
$\mathbf{G}_i^{(N_{\mathrm{s}})}$
Average and convert
$\bar{\mathbf{g}}_i$
$\mu_{q_\lambda}\left(\{f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}, \mathbf{Y}_i^{(t_s)})\}_{s=1}^{N_{\mathrm{s}}}\right)$
$\sigma_{q_\lambda}\left(\{f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}, \mathbf{Y}_i^{(t_s)})\}_{s=1}^{N_{\mathrm{s}}}\right)$
Sampling
ϕ_i

Fig. 4. The proposed VST network. The inputs in support set $\{\mathbf{X}_i^{(t_s)}\}_{s=1}^{N_{\mathrm{s}}}$ and query set $\{\tilde{\mathbf{X}}_i^{(t_q)}\}_{q=1}^{N_{\mathrm{q}}}$ are first processed by the extractor to generate the feature maps. Then, data samples in support set $\{f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\}_{s=1}^{N_{\mathrm{s}}}$ are used to determine the distribution of task-specific parameters ϕ_i through the amortization network. After that, the sampled task-specific parameters ϕ_i and the features provided by the extractor $f_{\theta_{\mathrm{ext}}}(\tilde{\mathbf{X}}_i^{(t_q)})$ are sent to the generator. The predicted results $\hat{\mathbf{Y}}_i^{(t_q)}$ are determined by the generator based on the task-specific parameters and the extracted features.

$$\underset{\theta, \lambda}{\mathrm{maximize}}\ \mathbb{E}_{\tau_i \sim \gamma(\mathcal{T})}[\tilde{\mathcal{L}}(\theta, \lambda)]. \quad (11)$$

In summary, given the optimized θ and λ as well as the support set $\mathcal{D}_i^{\mathrm{s}}$ of task τ_i, the distribution of task-specific parameters ϕ_i can be determined. Then, the underlying distribution of the spatial-temporal pattern can be inferred from a small number of data samples in the support set without encountering the issue of overfitting. In the next subsection, we solve problem (11) by using a VST network. We train the developed VST network by using the BML algorithm to obtain the parameters θ and λ.

4.2 Variational Spatial-Temporal Network

We now present the proposed VST network, where θ and λ are the learnable parameters of the VST network. The proposed VST network extracts the common spatial-temporal dependencies shared by all tasks and adaptively infers the task-specific parameters ϕ_i. The structure of the proposed VST network is shown in Fig. 4. The proposed VST network includes three modules: an extractor which is parameterized by θ_{ext}, an amortization network which is parameterized by λ, and a generator which is parameterized by θ_{gen}. We define $\theta = \{\theta_{\mathrm{ext}}, \theta_{\mathrm{gen}}\}$ as the global parameters, which capture the shared spatial-temporal features of all tasks. The amortization network determines the task-specific parameters. In the following, we will explain these three modules in detail.

4.2.1 *Extractor*

The extractor is common to all prediction tasks. It is developed to pre-process the input from both the support and query sets to extract the spatial-temporal features shared by all tasks. In order to accurately predict the wireless traffic, it is important for the extractor to capture both the local short-term and long-term spatial-temporal features. The Conv3D network can extract dependencies of the traffic data in the spatial and temporal domains by using a 3D kernel. In the extractor, we apply the Conv3D operation to extract the local short-term spatial-temporal dependencies. We consider the kernel sizes of the Conv3D as $\kappa^{\text{ext}}_{\text{3D},1}$, $\kappa^{\text{ext}}_{\text{3D},2}$, and $\kappa^{\text{ext}}_{\text{3D},3}$. The number of channels is denoted as $H^{\text{ext}}_{\text{3D}}$. Moreover, let θ_{C3D} denote the learnable parameters in the Conv3D operation. We use the rectified linear unit (ReLU) as the activation function.

For the shared long-term spatial-temporal dependencies, we propose a dual-attention embedded ConvLSTM (DAConvLSTM) network. The DAConvLSTM network preserves the capabilities of the ConvLSTM network to learn long-term spatial-temporal dependencies through the LSTM cells and convolutional operations. Moreover, the DAConvLSTM network can extract the most important spatial-temporal features in the long term by using the dual-attention mechanism. In the following, the conventional ConvLSTM network is presented. Then, we introduce the proposed dual-attention mechanism. Given a P-timestamp input $\mathbf{X}^{(t)}_i = \{\mathbf{X}_{i,t-P+1}, \ldots, \mathbf{X}_{i,t}\}$ in the support set of task τ_i, the ConvLSTM operation on each element $\mathbf{X}_{i,p}$, where $p = t-P+1, \ldots, t$, can be expressed as

$$\begin{aligned}
\mathbf{i}_{i,p} &= \sigma(\mathbf{W}_{\text{xi}} * \mathbf{X}_{i,p} + \mathbf{W}_{\text{hi}} * \mathbf{H}_{i,p-1} + \mathbf{W}_{\text{ci}} \odot \mathbf{C}_{i,p-1} + \mathbf{b}_{\text{i}}),\\
\mathbf{f}_{i,p} &= \sigma(\mathbf{W}_{\text{xf}} * \mathbf{X}_{i,p} + \mathbf{W}_{\text{hf}} * \mathbf{H}_{i,p-1} + \mathbf{W}_{\text{cf}} \odot \mathbf{C}_{i,p-1} + \mathbf{b}_{\text{f}}),\\
\mathbf{C}_{i,p} &= \mathbf{f}_{i,p} \odot \mathbf{C}_{i,p-1}\\
&\quad + \mathbf{i}_{i,p} \odot \tanh(\mathbf{W}_{\text{xc}} * \mathbf{X}_{i,p} + \mathbf{W}_{\text{hc}} * \mathbf{H}_{i,p-1} + \mathbf{b}_{\text{c}}),\\
\mathbf{o}_{i,p} &= \sigma(\mathbf{W}_{\text{xo}} * \mathbf{X}_{i,p} + \mathbf{W}_{\text{ho}} * \mathbf{H}_{i,p-1} + \mathbf{W}_{\text{co}} \odot \mathbf{C}_{i,p} + \mathbf{b}_{\text{o}}),\\
\mathbf{H}_{i,p} &= \mathbf{o}_{i,p} \odot \tanh(\mathbf{C}_{i,p}),
\end{aligned}$$

where $*$ and $\odot$ denote the two-dimensional convolution (Conv2D) operator and Hadamard product, respectively. The kernel sizes of the Conv2D module are denoted as $\kappa^{\text{ext}}_{\text{2D},1}$ and $\kappa^{\text{ext}}_{\text{2D},2}$. We use $H^{\text{ext}}_{\text{2D}}$ to denote the number of channels of the Conv2D module. $\sigma(\cdot)$ is the sigmoid function. $\mathbf{i}_{i,p}$, $\mathbf{f}_{i,p}$, $\mathbf{C}_{i,p}$, $\mathbf{o}_{i,p}$, and $\mathbf{H}_{i,p}$ denote the input gate, forget gate, cell state, output gate, and hidden state, respectively. Note that the gates and states are all 3D tensors. $\mathbf{W}_{\text{xi}}$, $\mathbf{W}_{\text{hi}}$, $\mathbf{W}_{\text{ci}}$, and $\mathbf{b}_{\text{i}}$ are the weights and bias for the input gate, which need to be learned through network training. Similarly, $\mathbf{W}_{\text{xf}}$, $\mathbf{W}_{\text{hf}}$, $\mathbf{W}_{\text{cf}}$, and $\mathbf{b}_{\text{f}}$ are the weights and bias associated with the forget gate. $\mathbf{W}_{\text{xc}}$, $\mathbf{W}_{\text{hc}}$, and $\mathbf{b}_{\text{c}}$ are the weights and bias related to the cell state. $\mathbf{W}_{\text{xo}}$, $\mathbf{W}_{\text{ho}}$, $\mathbf{W}_{\text{co}}$, and $\mathbf{b}_{\text{o}}$ are the weights and bias for the output gate. Note that the weights and biases are shared across all tasks. In addition, $\tanh(\cdot)$ is the hyperbolic tangent function. The input-to-state, cell-to-state, and cell-to-cell transitions are element-wise controlled by each gate $\mathbf{i}_{i,p}$, $\mathbf{f}_{i,p}$, and $\mathbf{o}_{i,p}$. This facilitates the model to keep the historical information and learn to forget unimportant information in the spatial-temporal domain. To further improve the capability of the network to capture the most important long-term spatial and temporal trends shared by all tasks, we propose to embed two attention mechanisms in the ConvLSTM network.

Spatial attention (S-ATT) mechanism: We propose to embed an S-ATT mechanism in the ConvLSTM network to capture the important spatial correlation. We develop an attention embedded ConvLSTM (ACLSTM) cell by reconstructing the input and output gates of ConvLSTM with the S-ATT mechanism [25]. In particular, by using the S-ATT mechanism, the input gate is reconstructed as follows:

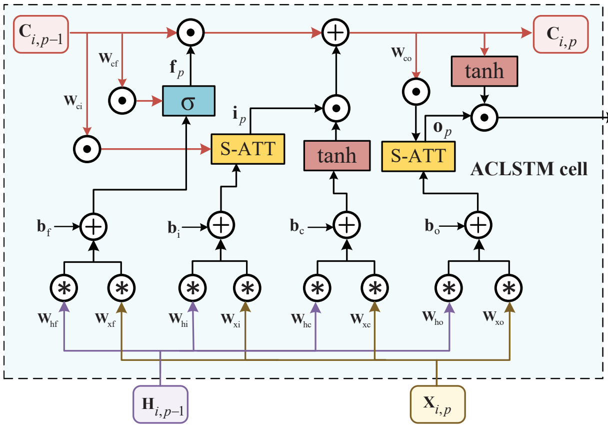


Fig. 5. The structure of an ACLSTM cell.

$$\begin{aligned}
\mathbf{Z}_{i,p} = \mathbf{W}_{\text{i}} * \tanh(&\mathbf{W}_{\text{xi}} * \mathbf{X}_{i,p} + \mathbf{W}_{\text{hi}} * \mathbf{H}_{i,p-1} \\
&+ \mathbf{W}_{\text{ci}} \odot \mathbf{C}_{i,p-1} + \mathbf{b}_{\text{i}}), \qquad (12)
\end{aligned}$$

$$\mathbf{A}^{jk}_{i,p}(h) = \frac{\exp(\mathbf{Z}^{jk}_{i,p}(h))}{\max\limits_{\hat{j},\hat{k}} \exp(\mathbf{Z}^{\hat{j}\hat{k}}_{i,p}(h))}, \qquad (13)$$

$$\begin{aligned}
\mathbf{i}_{i,p} = \{\mathbf{A}^{jk}_{i,p}(h) \mid h = 1, \ldots, H,& \qquad (14)\\
j = 1, \ldots, M,\ k = 1, \ldots, N\},&
\end{aligned}$$

where $\mathbf{W}_{\text{i}}$ is a Conv2D kernel with size $\kappa^{\text{ext}}_{\text{ATT},1}$ and $\kappa^{\text{ext}}_{\text{ATT},2}$. The number of channels is equal to $H^{\text{ext}}_{\text{ATT}}$. The term $\max_{\hat{j},\hat{k}} \exp(\mathbf{Z}^{\hat{j}\hat{k}}_{i,p}(h))$ corresponds to the maximum element chosen within channel h of $\mathbf{Z}_{i,p}$, for $h = 1, \ldots, H^{\text{ext}}_{\text{ATT}}$. The division by the maximum value ensures that the attention scores are distributed in the range between zero and one. The output gate of the ConvLSTM cell can be reconstructed in a similar manner as the input gate shown in (12)−(14). By embedding the S-ATT mechanism into the ConvLSTM network, the ACLSTM cell can focus on the most important long-term spatial features shared by all tasks. The structure of an ACLSTM cell is shown in Fig. 5.

Temporal attention (T-ATT) mechanism: Given an input sequence $\mathbf{X}^{(t)}_i = \{\mathbf{X}_{i,t-P+1}, \ldots, \mathbf{X}_{i,t}\}$ with length P, the final hidden state of an ACLSTM cell $\mathbf{H}_{i,p}$ contains information for the entire input sequence. However, using a single variable $\mathbf{H}_{i,p}$ to represent the information extracted from the sequence $\{\mathbf{X}_{i,t-P+1}, \ldots, \mathbf{X}_{i,t}\}$ may lead to information loss. To tackle this issue, we propose a T-ATT mechanism, which determines the weights for hidden states $\{\mathbf{H}_{i,t-P+1}, \ldots, \mathbf{H}_{i,t}\}$, such that the hidden states with more information in the temporal domain have larger weights in the output state. In particular, we reshape the hidden states $\mathbf{H}_{i,p}$ into a vector $\mathbf{h}_{i,p}$, for $p = t-P+1, \ldots, t$. We concatenate the final state vector $\mathbf{h}_{i,t}$ with $\mathbf{h}_{i,p}$, and form the vector $\bar{\mathbf{h}}_{i,t,p} = [\mathbf{h}^{\text{T}}_{i,t}\ \mathbf{h}^{\text{T}}_{i,p}]^{\text{T}}$, for $p = t-P+1, \ldots, t$. By using the T-ATT mechanism, the attention weights $a_{i,t,p}$ can

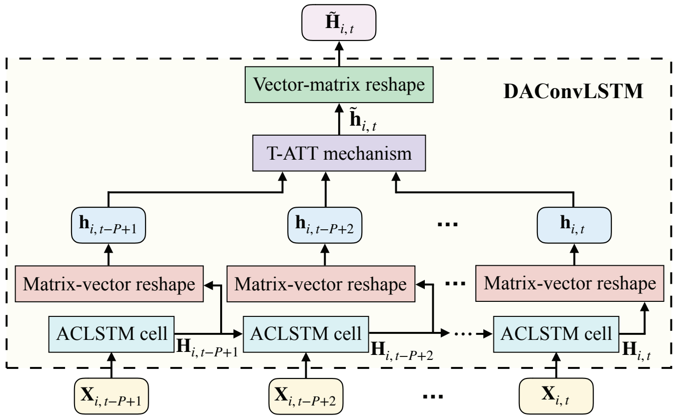


Fig. 6. The structure of the DAConvLSTM network.

be determined by the following softmax operation:

$$a_{i,t,p} = \frac{\exp\left\{\mathbf{v}^{\mathrm{T}} \tanh\left(\mathbf{W}_a \bar{\mathbf{h}}_{i,t,p}\right)\right\}}{\sum_{k=t-P+1}^{t} \exp\left\{\mathbf{v}^{\mathrm{T}} \tanh\left(\mathbf{W}_a \bar{\mathbf{h}}_{i,t,k}\right)\right\}}, \quad p = t-P+1, \ldots, t, \tag{15}$$

where $\mathbf{W}_a$ and $\mathbf{v}$ are the parameters of the T-ATT mechanism which are shared by all the prediction tasks. Then, the output of the T-ATT mechanism is given by the weighted hidden state vector $\tilde{\mathbf{h}}_{i,t}$:

$$\tilde{\mathbf{h}}_{i,t} = \sum_{p=t-P+1}^{t} a_{i,t,p} \mathbf{h}_{i,p}. \tag{16}$$

The weighted hidden state vector $\tilde{\mathbf{h}}_{i,t}$ is transformed back to a matrix, denoted as $\tilde{\mathbf{H}}_{i,t}$. By embedding the S-ATT and T-ATT mechanisms into the ConvLSTM network, we construct the DAConvLSTM network. The structure of the DAConvLSTM network is shown in Fig. 6. We use $\theta_{\mathrm{DACL}} = \{\mathbf{W}_{\mathrm{i}}, \mathbf{W}_{\mathrm{o}}, \mathbf{W}_{\mathrm{xk}}, \mathbf{W}_{\mathrm{hk}}, \mathbf{W}_{\mathrm{ck}}, \mathbf{b}_{\mathrm{k}}, \mathbf{W}_{\mathrm{a}}, \mathbf{v}\}$ to denote the learnable parameters in the DAConvLSTM network, where $\mathrm{k} \in \{\mathrm{i}, \mathrm{f}, \mathrm{c}, \mathrm{o}\}$ represents the gate or cell state in the ConvLSTM network.

To leverage the capability of both Conv3D and DAConvLSTM networks to learn spatial-temporal dependencies, we fuse the output of the two networks and obtain an ensembling result. Through fusion operation, the extractor can exploit the advantages of both Conv3D (to capture local spatial-temporal fluctuations) and DAConvLSTM (to extract long-term trends). This leads to an improved prediction performance compared with employing only one of the two models. The network parameters of the extractor are given by $\theta_{\mathrm{ext}} = \{\theta_{\mathrm{C3D}}, \theta_{\mathrm{DACM}}\}$ and the extractor network is denoted as $f_{\theta_{\mathrm{ext}}}(\cdot)$. The overall structure of the proposed extractor is shown in the top-left part of Fig. 4.

4.2.2 Amortization Network

To tackle the spatial-temporal variations, we develop an amortization network parameterized by λ to approximate the posterior distribution of the task-specific parameters. The structure of the amortization network is shown in the lower part of Fig. 4. The amortization network determines the mean and standard deviation of ϕ_i given the support set $\mathcal{D}_i^{\mathrm{s}}$ and the common knowledge provided by the extractor. The amortization network has three phases. In the first phase, the labels from the support set $\{\mathbf{Y}_i^{(t_s)}\}_{s=1}^{N_{\mathrm{s}}}$ are sent to a Conv2D network for pre-processing. The kernel sizes of the Conv2D network are κ_1^{amo} and κ_2^{amo}, and the number of channels is denoted as $H_{\mathrm{2D}}^{\mathrm{amo}}$. In the second phase, the output from the Conv2D network and the features provided by the extractor $f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)})$ are fused together. The results after fusion are denoted as $\{\mathbf{G}_i^{(1)}, \ldots, \mathbf{G}_i^{(N_{\mathrm{s}})}\}$. Then, the results $\{\mathbf{G}_i^{(1)}, \ldots, \mathbf{G}_i^{(N_{\mathrm{s}})}\}$ are averaged and converted to a vector $\bar{\mathbf{g}}_i$ of dimension D_{g}. In the third phase, the averaged results $\bar{\mathbf{g}}_i$ are sent to fully connected layers to determine the mean and standard deviation of the distribution over the task-specific parameters. The mean and standard deviation are denoted as $\mu_{q_\lambda}\left(f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\right)$ and $\sigma_{q_\lambda}\left(f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\right)$, respectively. Given the mean and standard deviation of task τ_i, we can sample the task-specific parameters $\phi_i = \mu_{q_\lambda}\left(f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\right) + \sigma_{q_\lambda}\left(f_{\theta_{\mathrm{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\right) \epsilon$ for task $\tau_i \sim \gamma(\mathcal{T})$. The dimension of the sampled task-specific parameters is denoted by D_ϕ.

4.2.3 Generator

The generator is used to produce the predicted results to approach the ground truth $\{\tilde{\mathbf{Y}}_i^{(t_q)}\}_{t_q=1}^{N_{\mathrm{q}}}$ in the query set for task τ_i. Two pieces of information need to be sent to the generator. They are the input from the query set and the task-specific parameters. Each input in the query set, which corresponds to $\tilde{\mathbf{X}}_i^{(t_q)}$, is successively processed by the extractor and the Conv2D module. Let κ_1^{gen} and κ_2^{gen} denote the kernel sizes, and H_{2D}^{gen} denote the number of channels of the Conv2D module. The output of the Conv2D module is then fused with the sampled task-specific parameters ϕ_i. The result after fusion is converted to a vector, which is denoted as $\mathbf{r}_i$ with dimension D_{r}. Then, $\mathbf{r}_i$ is fed into two fully connected layers successively. The output of the fully connected layers is reshaped to a 3D tensor with dimension $Q \times M \times N$, which corresponds to the predicted traffic values of the next Q timestamps for task τ_i. We denote θ_{gen} as the parameters of the generator. The final predicted results are expressed as $\hat{\mathbf{Y}}_i^{(t_q)} = f_{\theta_{\mathrm{gen}}}(\tilde{\mathbf{X}}_i^{(t_q)}, \phi_i)$, where $f_{\theta_{\mathrm{gen}}}$ represents the generator.

4.3 BML-based Training and Testing

We apply the BML algorithm for the training and testing of the VST network, such that the VST network can obtain the common knowledge shared by different prediction tasks and quickly adapt to different prediction tasks using the data samples in the support set. The BML-based training procedure is shown in Algorithm 1. For each training iteration, we sample a batch of tasks. For each batch of tasks, we partition the dataset into the support and query sets accordingly (Line 5) and determine the task-specific parameters based on the support set (Line 6). Given the task-specific parameters and the query set, we compute the ELBO in (10) (Line 7). Then, the parameters θ and λ are updated using the Adam optimizer [34] (Line 9).

In the testing stage, as shown in Algorithm 2, we sample a new task $\tau_j \sim \gamma(\mathcal{T})$ for testing and partition the dataset into the support and query sets (Line 2). The trained VST network generates the task-specific parameters ϕ_j based on the data samples in the support set $\mathcal{D}_j^{\mathrm{s}}$ (Line 4). Then, given

Algorithm 1 BML-based Training Procedure

```
1:  Input: Distribution of tasks $\gamma(\mathcal{T})$, initialize $\theta$ and $\lambda$, learning rate $\gamma$ of Adam optimizer [34], total number of iterations $N_{\max}$. $N_{\text{iter}} := 0$.
2:  while $N_{\text{iter}} < N_{\max}$ do
3:      Sample a batch of tasks from $\gamma(\mathcal{T})$.
4:      for each sampled task $\tau_i \sim \gamma(\mathcal{T})$ do
5:          Partition the dataset into the support set $\mathcal{D}_i^{\text{s}}$ and query set $\mathcal{D}_i^{\text{q}}$.
6:          Sample the task-specific parameters for task $\tau_i$. That is, $\phi_i = \mu_{q_\lambda}\left(f_{\theta_{\text{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\right) + \sigma_{q_\lambda}\left(f_{\theta_{\text{ext}}}(\mathbf{X}_i^{(t_s)}), \mathbf{Y}_i^{(t_s)}\right)\epsilon$.
7:          Compute the ELBO in (10).
8:      end for
9:      Solve problem (11) and update $\{\theta, \lambda\}$ based on Adam optimizer.
10:     $N_{\text{iter}} := N_{\text{iter}} + 1$.
11: end while
12: Output: Trained global parameters $\theta$ and the amortization network parameters $\lambda$.
```

Algorithm 2 BML-based Testing Procedure

```
1: Input: New traffic prediction task $\tau_j$ sampled from $\gamma(\mathcal{T})$, the trained $\theta$ and $\lambda$.
2: Partition the dataset into the support set $\mathcal{D}_j^{\text{s}}$ and query set $\mathcal{D}_j^{\text{q}}$.
3: for each $\tilde{\mathbf{X}}_j^{(t_q)}$ in query set $\mathcal{D}_j^{\text{q}}$ do
4:     Sample the task-specific parameters for task $\tau_j$. That is, $\phi_j = \mu_{q_\lambda}\left(f_{\theta_{\text{ext}}}(\mathbf{X}_j^{(t_s)}), \mathbf{Y}_j^{(t_s)}\right) + \sigma_{q_\lambda}\left(f_{\theta_{\text{ext}}}(\mathbf{X}_j^{(t_s)}), \mathbf{Y}_j^{(t_s)}\right)\epsilon$.
5:     Obtain $\hat{\mathbf{Y}}_j^{(t_q)} = f_{\theta_{\text{gen}}}(\tilde{\mathbf{X}}_j^{(t_q)}, \phi_j)$.
6: end for
7: Output: Predicted results $\hat{\mathbf{Y}}_j^{(t_q)}$.
```

the input $\tilde{\mathbf{X}}_j^{(t_q)}$ in the query set and task-specific parameters ϕ_j, the generator determines the predicted traffic $\hat{\mathbf{Y}}_j^{(t_q)}$ in the query set for task ϕ_j (Line 5). By using the BML algorithm, the trained VST network can quickly adapt to the testing task by using the data samples in the support set.

4.4 Computational Complexity Analysis

In this subsection, we provide a computational complexity analysis of the proposed VST-BML algorithm. For the BML-based training procedure, the computational complexity includes the computation required in the extractor, amortization network, and generator. The extractor contains DAConvLSTM and Conv3D networks. The computational complexity of the DAConvLSTM network is given by

$$\mathcal{O}_{\text{DAConvLSTM}} = \mathcal{O}\bigg(P\Big(H_{\text{2D}}^{\text{ext}} MN \kappa_{\text{2D},1}^{\text{ext}} \kappa_{\text{2D},2}^{\text{ext}} + MN(H_{\text{2D}}^{\text{ext}} \kappa_{\text{2D},1}^{\text{ext}} \kappa_{\text{2D},2}^{\text{ext}} + H_{\text{ATT}}^{\text{ext}} \kappa_{\text{ATT},1}^{\text{ext}} \kappa_{\text{ATT},2}^{\text{ext}})\Big)\bigg). \quad (17)$$

The computational complexity of the Conv3D network is

$$\mathcal{O}_{\text{Conv3D}} = \mathcal{O}\left(H_{\text{3D}}^{\text{ext}} MNP \kappa_{\text{3D},1}^{\text{ext}} \kappa_{\text{3D},2}^{\text{ext}} \kappa_{\text{3D},3}^{\text{ext}}\right). \quad (18)$$

The computational complexity of the extractor can be expressed as

$$\mathcal{O}_{\text{ext}} = \mathcal{O}_{\text{DAConvLSTM}} + \mathcal{O}_{\text{Conv3D}}. \quad (19)$$

For the amortization network, the computational complexity of the Conv2D and fully connected networks are given by $\mathcal{O}(H_{2D}^{\text{amo}} MN \kappa_1^{\text{amo}} \kappa_2^{\text{amo}})$ and $\mathcal{O}(D_{\text{g}} D_\phi)$, respectively. The computational complexity of the amortization network is

$$\mathcal{O}_{\text{amo}} = \mathcal{O}\left(H_{2D}^{\text{amo}} MN \kappa_1^{\text{amo}} \kappa_2^{\text{amo}} + D_{\text{g}} D_\phi\right). \quad (20)$$

Finally, the computational complexity of the generator is given by

$$\mathcal{O}_{\text{gen}} = \left(H_{2D}^{\text{gen}} MN \kappa_1^{\text{gen}} \kappa_2^{\text{gen}} + D_{\text{r}} QMN\right). \quad (21)$$

During network training, the data samples from the support set are processed by the extractor and amortization network. The data samples from the query set are processed by the extractor and generator. The overall computational complexity of BML-based training is given by

$$\mathcal{O}_{\text{train}} = \left(N_{\max} N_{\text{B}} \left(N_{\text{s}} \left(\mathcal{O}_{\text{ext}} + \mathcal{O}_{\text{amo}}\right) + N_{\text{q}} \left(\mathcal{O}_{\text{ext}} + \mathcal{O}_{\text{gen}}\right)\right)\right), \quad (22)$$

where $N_{\max}$ is the total number of iterations during training and N_{B} is the number of sampled tasks in each batch.

For online testing, given a new task, the computational complexity for adaptive traffic prediction is

$$\mathcal{O}_{\text{test}} = \left(N_{\text{s}} \left(\mathcal{O}_{\text{ext}} + \mathcal{O}_{\text{amo}}\right) + N_{\text{q}} \left(\mathcal{O}_{\text{ext}} + \mathcal{O}_{\text{gen}}\right)\right). \quad (23)$$

We can observe that the proposed VST-BML algorithm has a linear complexity with the sizes of a region, i.e., M and N. The number of data samples in the support and query sets, i.e., N_{s} and N_{q}, also affects the computational complexity, In the next section, we provide a runtime evaluation for both training and testing procedures.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed VST-BML algorithm on a real-world wireless traffic dataset [23], which is provided by Telecom Italia. Similar to some recent works (e.g., [9], [10], [18]), we show the prediction performance on the CDRs provided by this dataset, i.e., voice call, short message service (SMS), and Internet in the city of Milan in Italy. We use two metrics to evaluate the prediction performance. The first metric is the RMSE, which measures the difference between the predicted results and ground truth. The second metric is the MAE, which measures the average of the absolute difference between the predicted results and ground truth. We compare the RMSE and MAE of the proposed VST-BML algorithm with five baseline methods. We then present the runtime of offline training and online testing for different methods. After that, comparisons between the predicted results and ground truth in different regions are provided to further demonstrate the prediction performance of the proposed VST-BML algorithm. We also conduct ablation experiments to evaluate the effect of the dual-attention mechanism in the extractor. Finally, we evaluate the effect of the number of data samples in the support and query sets on the prediction accuracy. In the following, we first introduce the considered baseline methods and experimental settings. Then, the experimental results are presented.

TABLE 1
Prediction performance comparisons among different methods in terms of RMSE and MAE.

Methods	Voice call		SMS		Internet	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
ARIMA [11]	17.5877	15.8873	26.2275	22.5377	130.9993	113.4887
ConvLSTM [16]	12.5828	9.3775	18.8622	11.3792	92.7453	59.8861
MVSTGN [18]	13.3364	8.2637	14.9361	9.3586	91.1871	50.6461
STCNet [9]	9.8973	7.2803	16.0960	10.2739	89.5647	58.2248
ST-Tran [10]	7.3830	4.1749	13.8399	9.6982	88.2059	57.6397
VST-BML	**3.2544**	**2.1535**	**8.5369**	**6.8602**	**56.1430**	**39.7242**
Improvement	53.0% ↑	48.4% ↑	38.3% ↑	26.7% ↑	36.3% ↑	21.6% ↑

5.1 Baseline Methods and Experimental Settings

We compare the performance of our proposed VST-BML algorithm with that of the following baseline methods.

- ARIMA [11]: ARIMA is a statistical analysis model that learns the temporal dependency from the time series data and predicts wireless traffic. It has limited capability in capturing complex spatial-temporal dependency of the traffic data in a region.
- ConvLSTM [16]: ConvLSTM applies convolutional operations in both the input-to-state and state-to-state transitions in traditional LSTM. ConvLSTM network can extract both spatial and temporal dependencies.
- MVSTGN [18]: MVSTGN uses a GNN for wireless traffic prediction. The attention modules are embedded in the GNN to extract the global spatial-temporal correlation. Densely connected convolutional layers are employed to extract the local spatial-temporal dependencies of the nodes.
- STCNet [9]: STCNet captures spatial-temporal dependencies using the ConvLSTM network. After feature extraction, STCNet predicts wireless traffic based on CNNs.
- ST-Tran [10]: ST-Tran includes a spatial and temporal transformer block which can learn the spatial-temporal features. The learned features are fused together to make the final prediction.

We consider the traffic prediction on an hourly basis, where the collected raw traffic data is grouped into hourly scale, i.e., the duration between two consecutive timestamps is set to one hour. After aggregation, there are 1,488 hours in total. We consider the size of a region to be $M \times N = 10 \times 10$. In each prediction task, we choose $P = 5$ and $Q = 1$. That is, we aim to predict the wireless traffic of the next timestamp based on the previous five observations. We consider the number of data samples in the support set and query set to be $N_{\mathrm{s}} = 5$ and $N_{\mathrm{q}} = 1$, respectively. The proposed VST-BML algorithm can quickly adapt to traffic prediction on the query set by using only five data samples in the support set. Without loss of generality, we generate a task set which contains 5,000 prediction tasks (i.e., traffic prediction in randomly selected 5,000 regions) in total. 80% of the tasks are used for training, and the remaining 20% of the tasks are used for testing. The learning rate of the Adam optimizer [34] is set to 10^{-5}. Note that all the methods are trained using the same data samples and evaluated using the same testing dataset. BML-based training and testing methods are used in our proposed VST-BML algorithm. For the other five baseline methods, the training dataset is constructed by aggregating the data samples in the support and query sets of all training tasks. During training, the traffic volume is normalized to be between zero and one by using max-min normalization. After traffic prediction in the testing stage, the predicted results are rescaled back to their nominal values.

5.2 Experimental Results

We evaluate the prediction performance of the proposed VST-BML algorithm on the testing tasks. In the testing tasks, we perform wireless traffic prediction in new regions, which are different from the regions in the training tasks.

5.2.1 *Performance Comparison*

Table 1 summarizes the RMSE and MAE performance of different methods. It can be observed from Table 1 that the proposed VST-BML algorithm outperforms the other five baseline methods for all types of wireless traffic. In particular, the traditional statistical model ARIMA has the highest RMSE and MAE. This is because ARIMA can only capture simple temporal dependency of the time series data and has limited capability in tackling high-dimensional and complex spatial-temporal correlations. The deep learning based methods, i.e., ConvLSTM, MVSTGN, STCNet, and ST-Tran, have better performance than ARIMA since they can extract the spatial-temporal features. Those deep learning based methods can learn the spatial-temporal dependencies in a particular region based on convolutional operations, attention mechanism, LSTM cells, and graph representation. However, they have limited capability to learn the spatial-temporal variations across different regions. Even given the information provided by the support set, those methods are not able to accurately capture different spatial-temporal patterns in different regions. This is due to the fact that a larger number of data samples are required for those algorithms to learn a particular spatial-temporal pattern, while the support set contains only a limited number of data samples. Under this condition, retraining of the networks using sufficient data samples is needed for those algorithms to perform traffic prediction in a new region. On the other hand, the proposed VST-BML algorithm can tackle the spatial-temporal variations and provide accurate predictions in different regions by using only $N_{\mathrm{s}} = 5$ data

TABLE 2
Runtime for offline training and online testing

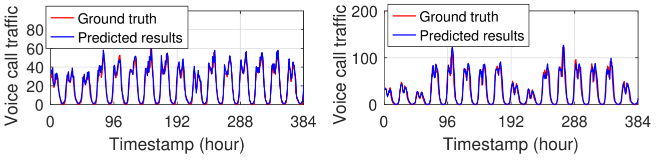
Methods	Offline Training	Online Testing
ARIMA	1.4084 (s)	0.0031 (s)
ConvLSTM	5.0812 (s/epoch)	0.1917 (s)
MVSTGN	15.4796 (s/epoch)	1.9231 (s)
STCNet	12.2590 (s/epoch)	1.3863 (s)
ST-Tran	14.3721 (s/epoch)	1.7868 (s)
VST-BML	**32.3419 (s/epoch)**	**2.2758 (s)**

samples from the support set without retraining the VST network. This demonstrates the fast adaptation capability of the proposed algorithm. For voice call traffic, compared with the ST-Tran model which has the best performance in all baselines, the proposed VST-BML algorithm can reduce the RMSE and MAE values by 53.0% and 48.4%, respectively. For SMS and Internet traffic, the proposed algorithm can provide 38.3% and 36.3% reduction in terms of RMSE, and 29.2% and 31.0% reduction in terms of MAE, respectively. Given the results in Table 1, we can summarize the advantages of the proposed VST-BML algorithm over the other five baseline methods as follows:

- The proposed VST network can well capture the common short-term and long-term spatial-temporal features shared across different regions through the extractor. The use of the dual-attention mechanism in the extractor enables the VST network to focus on the most important spatial-temporal information. The generated task-specific parameters by the amortization network have the representative capability to capture the particular spatial-temporal pattern in the target region.
- The BML-based training algorithm enables the VST network to effectively learn the underlying distribution of spatial-temporal patterns by using only a small number of data samples in the support set without encountering the issue of overfitting. Given a few data samples (e.g., five samples) in the support set in a region, the proposed VST-BML algorithm has the capability to quickly extract the complex spatial-temporal pattern in that region.

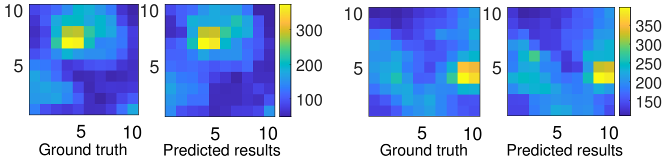
5.2.2 Runtime Comparison

In this subsection, we compare the execution time of offline training and online testing for different methods. We conduct the experiments using a computing server with an Intel Core i7-10700 @ 3.80 GHz CPU and an NVIDIA GeForce RTX 2070 GPU. The results are shown in Table 2. The results show that the proposed VST-BML algorithm requires a longer training time than the other baseline methods since the proposed algorithm requires an additional iteration loop for the computation of task-specific parameters for each task during training. For online testing, we can observe that the VST-BML algorithm has a comparable runtime as the baseline methods. We note that although the proposed algorithm has a longer offline training time, the trained VST network has the adaptation capability for traffic prediction in different regions and guarantees a high prediction accuracy in terms of RMSE and MAE. The other baseline methods are not able to provide accurate predictions. For the baseline methods, network retraining may be required before performing traffic prediction in a different region, which incurs additional computational overhead.



(a) A grid cell in southern part (b) A grid cell in northern part

Fig. 7. Comparisons of predicted results and ground truth over a time period of 16 days for voice call traffic. (a) and (b) show the comparison results of two grid cells randomly selected in two different regions.



(a) A region in southwestern part (b) A region in western part

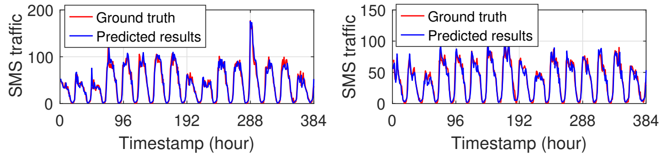
Fig. 8. Comparisons of predicted results and ground truth in a region for voice call traffic. (a) and (b) show the comparison results in two randomly selected regions.

5.2.3 Prediction Performance of the VST-BML Algorithm

To further illustrate the spatial-temporal variations and evaluate the adaptive prediction performance of the proposed VST-BML algorithm, we show the predicted results versus the ground truth. The prediction performance on voice call, SMS, and Internet traffic are presented in the following. For each type of traffic, we compare the predicted results and ground truth from both the temporal and spatial perspectives. The experiments are conducted in different regions which are randomly selected in different geographical locations in Milan.

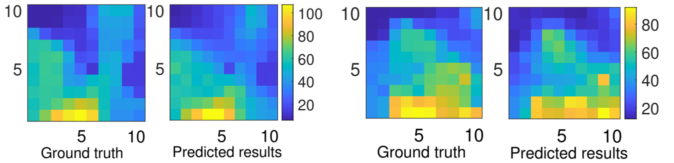
In Fig. 7, we plot the predicted results by the VST-BML algorithm and the ground truth over a randomly selected 16-day (384 hours) period for voice call traffic. Figs. 7(a) and (b) show the results of two randomly selected grid cells from two different regions in the southern and northern parts of Milan, respectively. The results show that the wireless traffic in both grid cells changes periodically, with peaks and valleys appearing every 24 hours. However, the temporal patterns in these two cells vary a lot. Specifically, in Fig. 7(a), the daily peak values of wireless traffic are between 30 and 60. On the other hand, in Fig. 7(b), the daily peak values fall within a larger range, i.e., between 25 and 125. While there exist temporal variations, we can observe that the proposed VST-BML algorithm can consistently provide accurate traffic predictions for both grid cells over the selected 16 days. The results in Fig. 7 demonstrate that the proposed VST-BML algorithm has a strong adaptation capability to tackle temporal variations across different regions.

To evaluate the adaptation capability of the proposed algorithm on spatial variations, in Fig. 8, we show the predicted results by the VST-BML algorithm and the ground truth in different regions, where each region contains a group of 10×10 grid cells. The results are obtained from an



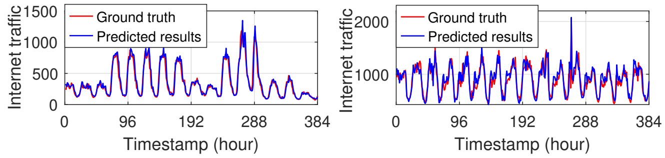
(a) A grid cell in eastern part (b) A grid cell in southern part

Fig. 9. Comparisons of predicted results and ground truth over a time period of 16 days for SMS traffic. (a) and (b) show the comparison results of two grid cells randomly selected in two different regions.



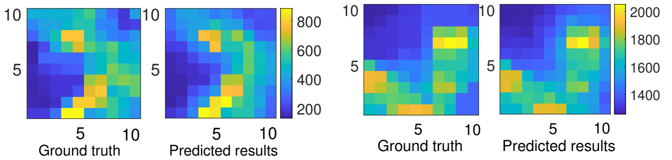
(a) A region in northwestern part (b) A region in southwestern part

Fig. 10. Comparisons of predicted results and ground truth in a region for SMS traffic. (a) and (b) show the comparison results in two randomly selected regions.



(a) A grid cell in northeastern part (b) A grid cell in central part

Fig. 11. Comparisons of predicted results and ground truth over a time period of 16 days for Internet traffic. (a) and (b) show the comparison results of two grid cells randomly selected in two different regions.



(a) A region in southeastern part (b) A region in eastern part

Fig. 12. Comparisons of predicted results and ground truth in a region for Internet traffic. (a) and (b) show the comparison results in two randomly selected regions.

arbitrary timestamp. Figs. 8(a) and (b) present the heat maps of the predicted results and ground truth in two randomly selected regions from the southwestern and western parts of Milan, respectively. Each pixel represents a grid cell in a region and the brightness of each pixel represents the corresponding traffic volume of voice call. The results in Fig. 8 show that the traffic in these two regions has different spatial patterns, which are reflected by different traffic volumes and distributions. While the spatial patterns are highly diverse, the proposed VST-BML algorithm can provide accurate prediction results that well match the spatial patterns in the target regions. This indicates that the proposed VST-BML algorithm has the fast adaptation capability to capture spatial variations in different regions by using $N_{\mathrm{s}} = 5$ data samples.

In Figs. 9 and 10, we show the predicted results versus the ground truth of SMS traffic. Fig. 9 shows the comparison between the predicted results and ground truth over 16 days in two randomly selected grid cells from two different regions in the eastern and southern parts, respectively. It can be observed that the predicted results by the proposed VST-BML algorithm match the ground truth for both regions. In Fig. 9(a), we can observe a sharp increase in traffic volume around timestamp 288, and the proposed VST-BML algorithm can still provide accurate predicted result which approaches the ground truth. By comparing the results shown in Figs. 9(a) and (b) which have diverse temporal patterns, we can conclude that the proposed VST-BML algorithm can quickly adapt to different traffic prediction tasks using five data samples. Fig. 10 compares the predicted results by the VST-BML algorithm for SMS traffic with the ground truth from two regions in the northwestern and southwestern parts, respectively. The results show that the predicted results are close to the ground truth in both regions, which demonstrates the capability of the proposed algorithm to tackle spatial variations across different regions.

We then show the predicted results versus ground truth for Internet traffic. We present the experimental results from temporal and spatial perspectives in Figs. 11 and 12, respectively. Similar to the previous sets of experiments, these results are from different regions which have diverse spatial-temporal patterns. We can observe that Internet traffic changes more dynamically when compared with voice call or SMS traffic. In particular, the peak Internet traffic in each day is much higher, and the ratio between peak traffic and off-peak traffic is larger. For Internet traffic, experimental results show that the proposed algorithm can still quickly capture various spatial-temporal patterns in wireless traffic by using five data samples in the support set from a region. The reasons can be attributed to the strong capability of the proposed algorithm in extracting shared common features and adaptively capturing spatial-temporal patterns in the target regions.

5.2.4 Ablation Study

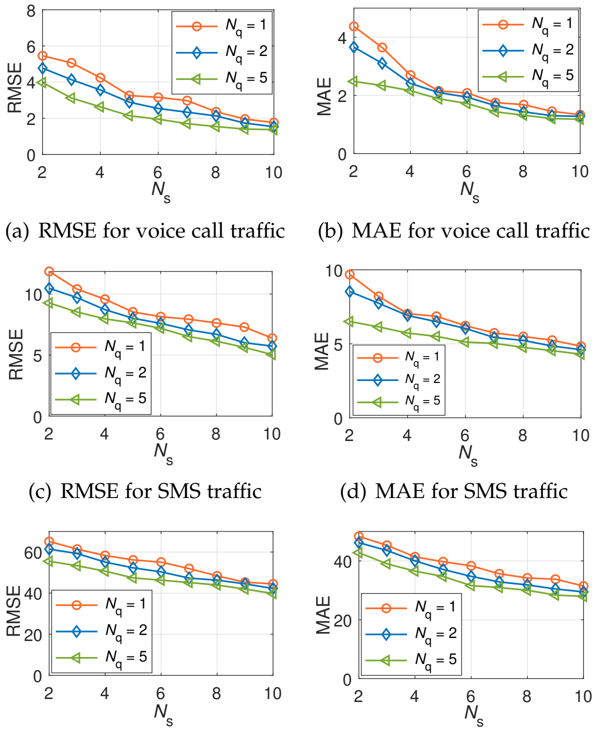
In this subsection, we evaluate the effect of the dual-attention mechanism in the extractor, which is used to capture the most important long-term spatial-temporal features. We conduct a set of ablation experiments on all three types of wireless traffic in the dataset. We consider the following cases: (a) without attention mechanism (denoted by “w.o. ATT”), (b) with T-ATT mechanism only (denoted by “with T-ATT”), (c) with S-ATT mechanism only (denoted by “with S-ATT”), and (d) with dual-attention mechanism (denoted by “with DA”), The RSME and MAE results of the proposed VST-BML algorithm are shown in Table 3. The results show that the prediction accuracy can considerably be improved by using the dual-attention mechanism, which can extract the most important long-term spatial-temporal dependencies. In addition, it is shown that using the S-ATT mechanism brings more performance gains than using the T-ATT mechanism for all three types of traffic. Since the traffic patterns have more significant variations in the spatial domain across different regions, effectively capturing the spatial correlations in a region is important.

5.2.5 Effect of the Number of Data Samples

Finally, we evaluate the effect of N_{s} and N_{q}, which are the number of data samples in the support and query sets,

TABLE 3
The effect of the dual-attention module on the prediction performance.

Type	Module	RMSE	MAE
Voice call traffic	w.o. ATT	5.9735	4.7470
	with T-ATT	5.2958	4.1185
	with S-ATT	4.1732	3.0039
	with DA	3.2544	2.1535
SMS traffic	w.o. ATT	11.2014	8.7244
	with T-ATT	10.1654	7.2441
	with S-ATT	9.6615	7.0478
	with DA	8.5369	6.8602
Internet traffic	w.o. ATT	67.2338	49.3350
	with T-ATT	64.1340	44.1206
	with S-ATT	60.2456	41.3326
	with DA	56.1430	39.7242



(a) RMSE for voice call traffic (b) MAE for voice call traffic

(c) RMSE for SMS traffic (d) MAE for SMS traffic

(e) RMSE for Internet traffic (f) MAE for Internet traffic

Fig. 13. Evaluation of the effect of N_s and N_q on prediction accuracy. Figures (a)−(f) illustrate the RMSE and MAE performance for three types of wireless traffic.

respectively. In Fig. 13, we show the RMSE and MAE performance for all three types of wireless traffic with different values of N_s and N_q. We consider N_q to be equal to 1, 2, and 5 and N_s varies from 2 to 10. It can be observed from the figures that the proposed VST-BML algorithm can provide more accurate predicted results with larger N_s and N_q. Both RMSE and MAE decrease with an increasing number of data samples in the support and query sets. The reasons are as follows. When more data samples are available in the support set (i.e., N_s increases), the proposed algorithm can obtain more information and extract more spatial-temporal features. In addition, we note that the goal is to adaptively provide accurate predictions for the traffic in the query set of each region. When N_q increases, more data samples in the query set can be used for the calculation of the objective function (10) in step 7 of Algorithm 1. This leads to more accurate averaged objective values and better prediction performance. The proposed algorithm can learn a better strategy for adaptive traffic prediction and improve prediction performance. Therefore, by using more data samples in the support and query sets, the proposed algorithm can tackle spatial-temporal variations more effectively and provide more accurate predicted results.

6 Conclusion

In this paper, we investigated the adaptive traffic prediction problem in wireless networks, where there exist strong spatial-temporal variations in wireless traffic across different regions. We proposed a VST-BML algorithm to tackle spatial-temporal variations and predict traffic in different regions. We evaluated the performance of the proposed VST-BML algorithm on a real-world dataset which contains three types of traffic, i.e., voice call, SMS, and Internet. The results showed that the proposed VST-BML algorithm can provide more accurate predicted results when compared with five baseline methods. We also compared the predicted results with the ground truth in different regions. Results showed that our proposed algorithm can consistently provide accurate predicted results and has a fast adaptation capability. Moreover, experimental results showed that when increasing the number of data samples in the support and query sets, the prediction accuracy can further be improved.

For future work, we are interested in developing a more flexible framework for adaptive traffic prediction. We will consider the size of each region being different and design an algorithm that is applicable for traffic prediction in regions with different sizes. Furthermore, in this work, we predicted future traffic on a fixed time scale (i.e., on an hourly basis). It would be beneficial to design a flexible scheme that enables the prediction of future traffic on various time scales (e.g., on an hourly, daily, and weekly basis), which can facilitate diverse resource management requirements (e.g., dynamic resource allocation, network infrastructure planning and deployment).

References

[1] Q. Wu, X. Chen, Z. Zhou, L. Chen, and J. Zhang, "Deep reinforcement learning with spatio-temporal traffic forecasting for data-driven base station sleep control," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 935–948, Apr. 2021.

[2] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, Jun. 2019.

[3] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. of IEEE Int. Conf. Computer Commun. (INFOCOM)*, Atlanta, GA, May 2017.

[4] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. Zhang, "Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 212–217, Apr. 2020.

[5] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[6] 3GPP TR 22.874, "Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS (Release 18)," Dec. 2021.

[7] ITU, "Enter the ITU challenge to optimize 5G networks with AI," *ITU Hub*, Jul. 2021.

[8] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. of IEEE Int. Conf. Computer Commun. (INFOCOM)*, Atlanta, GA, May 2017.

[9] 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.

[10] Q. Liu, J. Li, and Z. Lu, "ST-Tran: Spatial-temporal transformer for cellular traffic prediction," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3325–3329, Oct. 2021.

[11] Y. Shu, M. Yu, J. Liu, and O. Yang, "Wireless traffic modeling and prediction using seasonal ARIMA models," in *Proc. of IEEE Int. Conf. Commun. (ICC)*, Anchorage, AK, May 2003.

[12] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.

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

[14] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, Apr. 2017.

[15] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. and Comput. (MobiHoc)*, Los Angeles, CA, Jun. 2018.

[16] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. of Adv. Neural Inf. Process. Syst (NeurIPS)*, Montreal, Canada, Dec. 2015.

[17] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, Sept. 2019.

[18] Y. Yao, B. Gu, Z. Su, and M. Guizani, "MVSTGH: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2837–2849, May 2023.

[19] 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.

[20] 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.

[21] F. Sun, P. Wang, J. Zhao, N. Xu, J. Zeng, J. Tao, K. Song, C. Deng, J. Lui, and X. Guan, "Mobile data traffic prediction by exploiting time-evolving user mobility patterns," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4456–4470, Dec. 2022.

[22] 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.

[23] G. Barlacchi, M. D. 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," *Sci. Data*, vol. 2, no. 150055, Oct. 2015.

[24] W. Wang, C. Zhou, H. He, W. Wu, W. Zhuang, and X. Shen, "Cellular traffic load prediction with LSTM and Gaussian process regression," in *Proc. of IEEE Int. Conf. Commun. (ICC)*, Jun. 2020.

[25] 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.

[26] S. Fang, X. Pan, S. Xiang, and C. Pan, "Meta-MSNet: Meta-learning based multi-source data fusion for traffic flow prediction," *IEEE Signal Process. Lett.*, vol. 28, pp. 6–10, 2021.

[27] T. Kuber, I. Seskar, and N. Mandayam, "Traffic prediction by augmenting cellular data with non-cellular attributes," in *Proc. of IEEE Wireless Commun. and Netw. Conf. (WCNC)*, Nanjing, China, Mar./Apr. 2021.

[28] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *Proc. of IEEE Int. Conf. Computer Commun. (INFOCOM)*, May 2021.

[29] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting, Third Edition*. Springer, 2016.

[30] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1033–1046, Jun. 2012.

[31] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sept. 2022.

[32] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008–2026, Aug. 2019.

[33] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of Int. Conf. Learning Representations (ICLR)*, Banff, Canada, Apr. 2014.

[34] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. of Int. Conf. Learning Representations (ICLR)*, San Diego, CA, May 2015.

Zihuan Wang (Graduate Student Member, IEEE) received the B.Sc. and M.A.Sc. degrees from Dalian University of Technology, Dalian, China, in 2017 and 2020, respectively. She is currently a Ph.D. Candidate in the Department of Electrical and Computer Engineering, The University of British Columbia (UBC), Vancouver, Canada. Her research interests include machine learning and artificial intelligence for wireless networks. She is the Assistant to Editor-in-Chief of IEEE Transactions on Wireless Communications. She received UBC's Four Year Fellowship in 2020, the Li Tze Fong Memorial Fellowship in 2023, and the Graduate Support Initiative Award in 2021-2023. She received the Best Paper Award at the IEEE ICC 2022.

Vincent W.S. Wong (S'94, M'00, SM'07, F'16) received the B.Sc. degree from the University of Manitoba, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. (now Microchip Technology Inc.). He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to 5G/6G wireless networks, Internet of things, mobile edge computing, smart grid, and energy systems. Dr. Wong is the Editor-in-Chief of *IEEE Transactions on Wireless Communications*. He has served as an Area Editor of *IEEE Transactions on Communications* and *IEEE Open Journal of the Communications Society*, an Associate Editor of *IEEE Transactions on Mobile Computing* and *IEEE Transactions on Vehicular Technology*, and a Guest Editor of *IEEE Journal on Selected Areas in Communications*, *IEEE Internet of Things Journal*, and *IEEE Wireless Communications*. Dr. Wong is the General Co-chair of *IEEE INFOCOM* 2024. He was a Tutorial Co-Chair of *IEEE GLOBECOM*'18, a Technical Program Co-chair of *IEEE VTC*2020-*Fall* and *IEEE SmartGridComm*'14, and a Symposium Co-chair of *IEEE ICC*'18, *IEEE SmartGridComm* ('13, '17) and *IEEE GLOBECOM*'13. He received the Best Paper Award at the *IEEE ICC* 2022 and *IEEE GLOBECOM* 2020. He is the Chair of the IEEE Vancouver Joint Communications Chapter and has served as the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications. Dr. Wong is an IEEE Vehicular Technology Society Distinguished Lecturer (2023−2025) and was an IEEE Communications Society Distinguished Lecturer (2019−2020).