IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 October 2022, accepted 11 December 2022, date of publication 19 December 2022, date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230908

RESEARCH ARTICLE

Network Traffic Modeling and Prediction Using Graph Gaussian Processes

SAJAD MEHRIZI AND SYMEON CHATZINOTAS, (Senior Member, IEEE)
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 1855 Luxembourg, Luxembourg

Corresponding author: Sajad Mehrizi (mehrizis@gmail.com)

This work was supported in part by the Luxembourg National Research Fund (FNR)–Autonomous Network Slicing for Integrated Satellite Terrestrial Transport Networks (ASWELL) Project under Grant 13718904, and in part by the European Research Council (ERC)–Actively Enhanced Cognition based Framework for Design of Complex Systems (AGNOSTIC) Project under Grant 742648.

ABSTRACT Traffic modeling and prediction is a vital task for designing efficient resource allocation strategies in telecommunication networks. This is challenging because network traffic data exhibits complex nonlinear spatiotemporal interactions. Moreover, the data can have missing values when traffic statistic collection is unavailable in certain nodes. In this paper, we introduce a graph Gaussian Process (GP) model for this challenging problem. The GP is a Bayesian non-parametric model and highly flexible in capturing complex patterns in the data. Additionally, it provides uncertainty information which can be exploited for robust resource allocation problems. The developed graph GP model is almost free of hyper-parameter tuning, can accurately capture short-term and long-term temporal patterns and can infer missing values by learning spatiotemporal interactions among the nodes in the network. Subsequently, we approximate the intractable posterior distribution using Variational Bayes (VB) algorithm which can be efficiently implemented. Finally, we evaluate the accuracy of the proposed model for predicting the data traffic using two real-world network datasets. Our simulation results shows that the proposed model can achieve better prediction accuracy with respect to the state-of-the-art approaches.

INDEX TERMS Gaussian process, Bayesian modeling, variational bayes, traffic prediction, graph data structure.

I. INTRODUCTION

Accurate traffic modeling and prediction is essential for efficient proactive resource allocation and traffic engineering in telecommunication networks. It has been widely used to perform different management tasks such as network maintenance, network optimization, routing policy design, load balancing, protocol design, anomaly detection and Virtualized Network Functions (VNF) deployment decisions [1], [2], [3]. Network traffic patterns can be affected by many factors, such as user behavior, network topology and routing strategy, and can have complex nonlinear spatiotemporal interactions. Moreover, even though currently available technologies, such as software-defined networking (SDN), allow for the centralized collection of network statistics, older equipment or failures often make it impossible to have a complete view across all network nodes. Therefore, due to missing values, it is challenging to effectively explore and utilize the data for accurate prediction.

In the literature, various statistical time series models and analysis methods have been developed for traffic prediction. The most commonly-used is autoregressive integrated moving average (ARIMA) model [4], [5]. A limitation of ARIMA is that it can only capture a short-term temporal interaction. Nevertheless, data analysis shows that data traffic can also exhibit long-term interaction [6]. The Seasonal ARIMA (SARIMA) model is an extension of ARMA can improve the prediction accuracy by capturing a long-term interaction and it has been adopted to traffic prediction in [7] and [8]. However, ARIMA and SARIMA are linear models and have limited expressiveness power therefore they cannot capture nonlinear patterns.

More recently, artificial neural networks have attracted major attention in both academia and industry. These models are non-linear and can potentially capture any non-linearity in the data. The authors in [9] applied a general multilayer

The associate editor coordinating the review of this manuscript and approving it for publication was Abdel-Hamid Soliman.

perceptron (MLP) network to predict the base station traffic under different wireless network setups. Recurrent neural networks (RNN) which are specific to sequence data are used in [5], [10], [11], [12], and [13]. In [14], a multiple RNN based learning models along with a multi-task learning framework is proposed to explore spatio-temporal correlations among base stations in cellular networks. Similar works used RNN combined with convolutional neural network (CNN) to capture the spatiotempral structure [15]. However, the aforementioned works assume that the data lie within a regular Euclidean space and do not explicitly exploit the graph structure of telecommunication networks, and therefore the proposed models may not perform satisfactorily.

Artificial graph neural networks [16], [17] which are particularly designed for modeling and predicting time-varying graph-based data are adopted in [18] and [19] for network traffic prediction. However, these developed artificial neural networks have two important limitations. Firstly, they cannot efficiently handle missing values due to their special architectures. Secondly, they cannot provide uncertainty information in the prediction because they are deterministic models.

GP, a Bayesian non-parametric model, which can efficiently capture uncertainty in the prediction has been used, more recently, by researchers for network traffic prediction [20], [21], [22], [23]. The authors in [20] studied a mixture of Gaussian processes using Dirichlet process to improve the scalability of inference and to model data non-stationarity. In [21], a GP model is used to capture a quasi-periodic pattern. In [22], the authors developed an enesemble learning algorithm where each learner is modeled by a GP and the predictions of the GPs are combined to improve the prediction accuracy. In [23] the alternating direction method of multipliers (ADMM) algorithm is used for parallel hyper-parameter optimization to scale up the GP inference. Nevertheless, the developed GP models are specifically designed for univariate time-series and cannot capture complex traffic pattern interactions among different nodes in the network.

In this paper, we aim to introduce a graph-based GP model for traffic prediction. In particular, our contributions are as follows.

- We develop a graph-based GP model which can capture the spatiotemproal interactions in the data. In particular, the GP exploits the structure of the telecommunication network which can be leveraged to infer the missing values.
- Moreover, a structured kernel function is proposed to capture short-term and long-term temporal dependencies to provide accurate prediction.
- Using the VB, we develop an inference algorithm to approximate the posterior distribution of the GP model. The developed inference algorithm is almost free of hyper-parameter tuning and the hyper-parameters are estimated using historical data.
- Finally, throughout our simulations, we show that the introduced graph-based GP model outperforms the state-of-the-art models on two real-world datasets.

This paper is organized as follows. The problem statement is described in Section II. In Section III, we provide an overview to GP. In Section IV, we introduce a GP model for network traffic modeling. In Section V, we develop a VB algorithm for inference and prediction. Finally, Section VI shows the simulation results and Section VII concludes the paper.

II. PROBLEM DEFINITION

We consider a communication network which consists of a set of nodes, e.g., routers, which are connected to each other by communication links. Due to the graph-structured topology, we define the traffic prediction problem on a graph, as depicted in Fig. 1. Mathematically, the network can be represented by un-directed graph $\mathcal{G}_t := (\mathbf{y}_t, \mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges (i.e., communication links). We denote the presence of an edge between node i and node j as $\{i, j\} \in \mathcal{E}$. Moreover, $\mathbf{y}_t \in R^M$ is a vector that contains data traffic at nodes at time t, e.g., in Mb per time unit, where M is the total number of nodes. Set $\mathcal{V}$ is divided into two disjoint sets $\mathcal{V}_o$ and $\mathcal{V}_{\bar{o}}$ such that $\mathcal{V} = \mathcal{V}_o \cup \mathcal{V}_{\bar{o}}$, where $\mathcal{V}_o$ and $\mathcal{V}_{\bar{o}}$ contain nodes that traffic can be measured and nodes that traffic cannot be measured respectively. The number observed nodes and missing nodes are respectively defined by M_o and $M_{\bar{o}}$ such that $M = M_o + M_{\bar{o}}$.

The objective in traffic prediction problem is to predict the traffic over the next H time steps given T historical observations of data traffic. In this paper, we advocate a probabilistic approach. This requires to compute the predictive distribution $p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+H} | \mathbf{y}_{o,t}, \ldots, \mathbf{y}_{o,t-T+1})$, where $\mathbf{y}_{ot} \in R^{M_o}$ is the observation vector of M_o nodes a time t, each element records historical traffic observations for a specific node in set $\mathcal{V}_o$. The traffic at nodes are not independent but related by pairwise relationships. In other words, it is expected that neighboring nodes have similar traffic patterns over time. Moreover, in practice, some of the nodes may degenerate, e.g., due to packet loss, or generate, e.g., edge nodes, the data traffic. Therefore, the flow balance equations cannot be used to predict the traffic at the missing nodes. In particular, the relationship among the nodes in the graph is not deterministic and is stochastic. Therefore, it is essential to effectively capture the network structure for improved prediction accuracy. In the next section, we review the basic of GP since it is used as the main tool to construct our graph-based probabilistic model for traffic prediction.

III. OVERVIEW TO GAUSSIAN PROCESS

GPs are powerful non-parametric Bayesian tools suitable for modeling real-world problems. A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution. Using a GP, we can define a distribution over non-parametric functions $f(\mathbf{x})$:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), K\left(\mathbf{x}, \mathbf{x}'\right)\right), \tag{1}$$

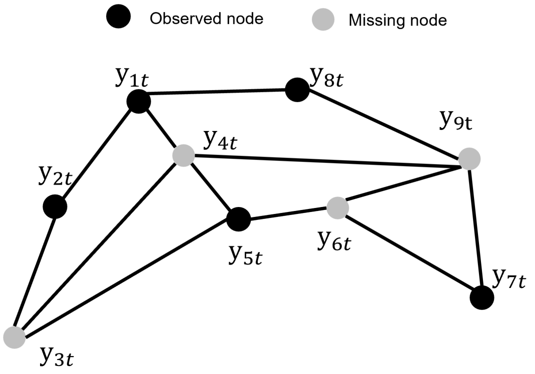

FIGURE 1. **Illustration of network traffic data.**

where $\mathbf{x}$ is an arbitrary input variable with Q dimensions, and the mean function, $m(\mathbf{x})$, and the Kernel function, $K(\mathbf{x}, \mathbf{x}')$, are respectively defined as:

$$m(\mathbf{x}) := \mathbb{E}[f(\mathbf{x})], \tag{2}$$

$$K(\mathbf{x}, \mathbf{x}') := \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))\left(f(\mathbf{x}') - m(\mathbf{x}')\right)\right]. \tag{3}$$

This means that a collection of N function value samples has a joint Gaussian distribution:

$$[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^T \sim \mathcal{N}(\boldsymbol{m}, \mathbf{K}), \tag{4}$$

where $\boldsymbol{m} := [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_N)]^T$ and the covariance matrix $\mathbf{K}$ has entries $[\mathbf{K}]_{i,j} := K(\mathbf{x}_i, \mathbf{x}_j)$. The kernel function specifies the main characteristics of the function that we wish to model and the basic assumption is that variables $\mathbf{x}$ which are close are likely to be correlated. Constructing a good kernel function for a learning task depends on intuition and experience. More details about GP can be found in [24].

IV. GAUSSIAN PROCESS FOR NETWORK TRAFFIC DATA

In this section, we introduce a Bayesian model based on GP for data traffic. The developed model includes both the missing and observed nodes. In Section V, we explain how the model is used for inference and prediction in presence of missing nodes in more details. Our model assumes that the traffic is e generated based on a factor analysis model with Gaussian likelihood as:

$$\mathbf{y}_t = \mathbf{W}\mathbf{f}_t + \mathbf{b} + \mathbf{n}_t, \tag{5}$$

where $\mathbf{W} := [\mathbf{w}_1^T, \ldots, \mathbf{w}_M^T]^T \in R^{M \times D}$ is factor loading matrix which row $\mathbf{w}_m$ captures node m specific patterns and $\mathbf{f}_t := [f_{11}, \ldots, f_{Dt}]^T \in R^D$ is a temporal latent variable vector which captures common temporal patterns among the nodes. Vector $\mathbf{b} \in R^M$ is the bias term which captures the overall traffic volume at each node. Moreover, $\mathbf{n}_t := [n_1(t), \ldots, n_M(t)]$ is the additive white noise assumed $n_m(t) \sim \mathcal{N}(0, \sigma)$, $\forall m = 1, \ldots, M$. The expressiveness power of the model is determined by the number of latent factors D. In particular, the model can capture more diverse and more complex patterns in the data as D increases. However, we should note that increasing D can also increase the risk of overfittng. Therefore, choosing the right value for D is important for achieving accurate prediction.

In order to effectively capture the graph structure of the network and temporal patterns in the traffic, it is important to model variables $\mathbf{W}$, $\mathbf{f}_t$ and $\mathbf{b}$ with flexible priors. In the following, we focus on this problem.

A. PRIOR FOR $\mathbf{f}_t$

In order to capture the time evolution of the data traffic, we assume that the latent variables $\mathbf{f}_t$ follows an additive model with four components given by:

$$f_{dt} := f_d(t) := f_{1d}(t) + f_{2d}(t) + f_{3d}(t) + f_{4d}(t), \forall d = 1, \ldots, D. \tag{6}$$

The additive model in (6) is used to capture the disjunction of four main temporal characteristics of the data. Function $f_{1d}(t)$ models a short-term trend and is assumed to be a GP with radial basis function (RBF) kernel as:

$$\begin{aligned} f_{1d}(t) &\sim \mathcal{GP}(0, k_{1d}(t, t')), \\ k_{1d}(t, t') &= \alpha_{1d} \exp\left(-\beta_{1d}\|t - t'\|^2\right). \end{aligned} \tag{7}$$

Parameters α_{1d} and β_{1d} model the important behavior of function $f_{1d}(t)$. In particular, α_{1d} captures the horizontal variation and β_{1d} captures the vertical variation of function $f_{1d}(t)$ in time domain. Function $f_{2d}(t)$ models a daily quasi periodic pattern and is modeled by a GP as:

$$\begin{aligned} f_{2d}(t) &\sim \mathcal{GP}(0, k_{2d}(t, t')), \\ k_{2d}(t, t') &= \alpha_{2d} \exp(-\beta_{2d}\|t - t'\|^2) \\ &\quad \times \exp(-\gamma_{1d} sin^2 \frac{\pi}{\lambda_1}(t - t')). \end{aligned} \tag{8}$$

Kernel function $k_{2d}(t, t')$ is the product of a RBF and a purely periodic RBF kernel which can capture the conjunction of both kernels [25], [26]. The overall kernel is a quasi periodic. As in 7, parameters α_{2d}, β_{2d} and γ_{1d} capture the horizontal and the vertical variations of function $f_{2d}(t)$. Moreover, parameter λ_1 should be set as the number of observations per day. In a similar way, function $f_{3d}(t)$ models a weekly quasi periodic pattern and is modeled by a GP as:

$$\begin{aligned} f_{3d}(t) &\sim \mathcal{GP}(0, k_{3d}(t, t')), \\ k_{3d}(t, t') &= \alpha_{3d} \exp(-\beta_{3d}\|t - t'\|^2) \\ &\quad \exp(-\gamma_{2d} sin^2 \frac{\pi}{\lambda_2}(t - t')), \end{aligned} \tag{9}$$

where λ_2 should be set as the number of observations per week. Similarly, parameters α_{3d}, β_{3d} and γ_{2d} capture the horizontal and the vertical variations of function $f_{3d}(t)$. Furthermore, function $f_{4d}(t)$ models the unstructured patterns and is given by:

$$\begin{aligned} f_{4d}(t) &\sim \mathcal{GP}(0, k_{4d}(t, t')), \\ k_{4d}(t, t') &= \alpha_{nd}\delta(t - t'). \end{aligned} \tag{10}$$

where $\delta(.)$ is the Dirac delta function. We note that, using the additive property of GP, function $f_d(t)$ can be rewritten as the following GP model:

$$f_d(t) \sim \mathcal{GP}(0, k_d(t,t')),$$
$$k_d(t,t') = k_{1d}(t,t') + k_{2d}(t,t') + k_{3d}(t,t') + k_{4d}(t,t'). \tag{11}$$

B. PRIOR FOR W

Now we define prior destitution for $\mathbf{W}$ such that to explicitly exploit the graph topology of the network. To do so, we use a Gaussian random field (GRF) model as:

$$p(\mathbf{w}_{:,d}) \propto e^{-\frac{\theta_{d1}}{2}\sum_{\{i,j\}\in\mathcal{E}}(w_{id}-w_{jd})^2 - \frac{\theta_{d2}}{2}\sum_i w_{id}^2}, \quad \forall d = 1,\ldots,D, \tag{12}$$

or equivalently:

$$p(\mathbf{w}_{:,d}) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_d^{-1}), \tag{13}$$

where $\mathbf{Q}_d = \theta_{d1}\mathbf{L} + \theta_{d2}\mathbf{I}_M$ and $\mathbf{L}$ is the graph Laplacian matrix. The graph Laplacian matrix is defined as:

$$\mathbf{L} = \mathbf{I}_M - \mathbf{A}, \tag{14}$$

In 14, $\mathbf{A}$ is the adjacency matrix corresponding to the graph where $\mathbf{A}_{ij}$ is 1 if node i is connected to node j otherwise is 0. GRF is a sparse approximation of a GP defined over discrete input set. The connection between GRF and GP has been studied in [27]. The assumption by using the GRF is that nodes that are connected to each other should have similar data traffic patterns. The GRF encourages the values $w_{i,d}$ and $w_{j,d}$ to be similar if nodes i and j are neighbors and therefore acts as a regularizer which can improve the prediction accuracy.

C. PRIOR FOR b

Similarly, for the bias term, $\mathbf{b}$, we use a GRF as prior given by:

$$p(\mathbf{b}) = \mathcal{N}(\mathbf{Za}, \mathbf{Q}_{D+1}^{-1}), \tag{15}$$

where $\mathbf{Q}_{D+1} = \theta_{D+1,1}\mathbf{L} + \theta_{D+1,2}\mathbf{I}$. Matrix $\mathbf{Z}$ contains some nodes specific features such as the degree of nodes. In particular, by using this prior we encourage neighbor nodes to have similar bias if they have similar features.

Since the coefficient $\mathbf{a}$ in (15) is unknown, we put the following non-informative prior over its values:

$$p(a_i) = \mathcal{N}(0, 10^{-3}), \forall i. \tag{16}$$

Finally, the complete probabilistic graph-based GP model is depicted in Fig. 2. A summary of the model's parameters is also shown in Tab. 1.

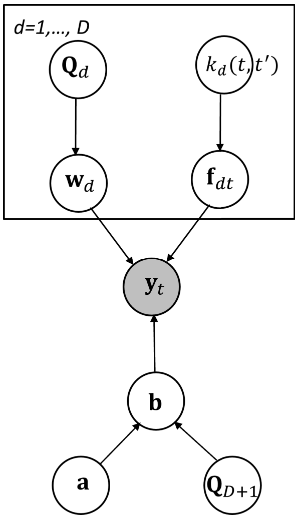

FIGURE 2. Probabilistic graph-based GP model.

TABLE 1. A summary of parameters.

Parameters	Description
$\mathbf{A}$	Adjacency matrix
$\mathbf{b}$	Bias term
σ	Data traffic noise variance
$\mathbf{y}_t$	Data traffic vector
$\mathbf{W}$	Factor loading matrix
$\mathbf{Q}_d$	GRF precision matrix
$k_d(t,t')$	Kernel function
D	Latent factor dimensions
$\mathbf{Z}$	Nodes' feature matrix
$M_{\bar{o}}$	Number of missing nodes
M_o	Number of observed nodes
$\mathbf{a}$	Regression coefficient of nodes' features
$\mathbf{f}_t$	Time latent factor
M	Total number of nodes

D. THE COVARIANCE STRUCTURE

To have a better understanding about the underlying data pattern that the model in (5) can capture, we compute the overall covariance structure as:

$$\mathbb{E}[y_{mt}y_{m't'}] = \sum_{d=1}^{D}\mathbb{E}[w_{md}, w_{m'd}]\mathbb{E}[f_d(t), f_d(t')] + \sigma \tag{17}$$

where for simplicity of the analysis we ignore the bias term. We can express (17) in a matrix form given by:

$$\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \sum_{d=1}^{D}\mathbf{Q}_d^{-1}\otimes\mathbf{K}_d + \sigma\mathbf{I}_{MT}, \tag{18}$$

where $\mathbf{y} = vec(\mathbf{Y}^T)$, $\mathbf{Y} = [\mathbf{y}_1,\ldots,\mathbf{y}_T]$. It can be seen that the covariance matrix is sum of D covariance matrices which each has a Kronecker structure. In a special case, the covariance matrix has a simple Kronecker structure for $D = 1$. Designing covariance matrices with Kronecker structure is a common approach to model multi-variate output GP for graph structured data, e.g., in [28]. The limitation with the simple Kronecker structure is that it cannot capture complex patterns in the data. On the other hand, the sum Kronecker structure introduces much more expressiveness power for complex and non-linear patterns.

V. MODEL INFERENCE AND PREDICTION

In this section, we focus on model inference and prediction in the light of the data traffic measurements at the observed nodes in set $\mathcal{V}_o$. We note that the likelihood function in (5) depends on the parameters of both missing and observed nodes which are highly correlated through the GRF prior distributions in (13) and (15). However, the traffic measurements are unavailable for the missing nodes and therefore the inference is not trivial. To tackle the issue, our approach is to integrate out the parameters of missing nodes from the model to obtain a marginal model which includes only the observed nodes. Using the marginal model, we can perform the inference. In Section V-A, we focus on this problem. After the inference, we define posterior predictive distributions to make predictions about both the observed and the missing nodes by exploiting their interaction structures encoded in the GRF priors. This problem is explained in Section V-B.

A. INFERENCE

As we mentioned previously, the inference problem requires to compute the marginal model of the observed nodes. Let divide matrix $\mathbf{W}$ into two submatrices $\mathbf{W}_o \in R^{M_o \times D}$ and $\mathbf{W}_{\bar{o}} \in R^{M_{\bar{o}} \times D}$, where $\mathbf{W}_o$ and $\mathbf{W}_{\bar{o}}$ are the latent factors for the observed nodes and missing nodes respectively. Similarly, we divide vector $\mathbf{b}$ into two subvectors $\mathbf{b}_o \in R^{M_o}$ and $\mathbf{b}_{\bar{o}} \in R^{M_{\bar{o}}}$, where $\mathbf{b}_o$ and $\mathbf{b}_{\bar{o}}$ are the bias terms for the observed nodes and the missing nodes respectively. The marginal prior distributions of the parameters of observed nodes can be computed as:

$$p(\mathbf{W}_o)=\int p(\mathbf{W}_o, \mathbf{W}_{\bar{o}})d\mathbf{W}_{\bar{o}}, \;\; p(\mathbf{b}_o)=\int p(\mathbf{b}_o, \mathbf{b}_{\bar{o}})d\mathbf{b}_{\bar{o}}, \tag{19}$$

where the joint distributions $p(\mathbf{W}_o, \mathbf{W}_{\bar{o}})$ and $p(\mathbf{b}_o, \mathbf{b}_{\bar{o}})$ are given by the GRFs in (12) and (15) respectively. By re-arranging $\mathbf{W}$ and $\mathbf{b}$, the GRF models can be rewritten as:

$$\begin{bmatrix}\mathbf{w}_{o,:,d}\\ \mathbf{w}_{\bar{o},:,d}\end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_d^{-1}), \begin{bmatrix}\mathbf{b}_o\\ \mathbf{b}_{\bar{o}}\end{bmatrix} \sim \mathcal{N}(\begin{bmatrix}\mathbf{Z}_o\mathbf{a}\\ \mathbf{Z}_{\bar{o}}\mathbf{a}\end{bmatrix}, \mathbf{Q}_{D+1}^{-1}), \tag{20}$$

$$\mathbf{Q}_d = \begin{bmatrix}\mathbf{Q}_{11d} & \mathbf{Q}_{12d},\\ \mathbf{Q}_{21d} & \mathbf{Q}_{22d}\end{bmatrix}, \;\; \forall d = 1, \ldots, D+1$$

where $\mathbf{Q}_{11d} \in R^{M_o \times M_o}$, $\mathbf{Q}_{12d} = \mathbf{Q}_{21d}^T \in R^{M_o \times M_{\bar{o}}}$ and $\mathbf{Q}_{11d} \in R^{M_{\bar{o}} \times M_{\bar{o}}}$. Using Schur complement lemma [29], we can compute the marginal distributions of $\mathbf{W}_o$ and $\mathbf{b}_o$ as:

$$p(\mathbf{w}_{o,:,d}) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_{o,d}^{-1}), \;\; p(\mathbf{b}_o) = \mathcal{N}(\mathbf{Z}_o\mathbf{a}, \mathbf{Q}_{o,D+1}^{-1}), \tag{21}$$

$$\mathbf{Q}_{o,d} = \mathbf{Q}_{11d} - \mathbf{Q}_{12d}\mathbf{Q}_{22d}^{-1}\mathbf{Q}_{21d}, \;\; \forall d = 1, \ldots, D+1.$$

Given the marginal priors in (21), the inference problem focuses on the following Gaussian marginal model:

$$\mathbf{y}_{ot} = \mathbf{W}_o\mathbf{f}_t + \mathbf{b}_o + \mathbf{n}_t, \tag{22}$$

where $\mathbf{y}_{ot}$ contains the data traffic measurements of the observed nodes.

In particular, the inference goal is to compute the posterior distribution of random variables $\mathbf{h} := \{\mathbf{F}, \mathbf{W}_o, \mathbf{b}_o, \mathbf{a}\}$, where $\mathbf{F} := [\mathbf{f}_1, \ldots, \mathbf{f}_T] \in R^{D \times T}$; and find the best values for the hyperparameters of GPs $\boldsymbol{\psi}^{GP} := \{\boldsymbol{\psi}_d^{GP}\}_{d=1}^{D}$, $\boldsymbol{\psi}_d^{GP} := \{\alpha_{d1}, \alpha_{d2}, \alpha_{d3}, \beta_{d1}, \beta_{d1}, \beta_{d1}, , \beta_{d2}, \gamma_{d1}, \gamma_{d2}\}$, GRFs $\boldsymbol{\psi}^{GRF} := \{\boldsymbol{\psi}_d^{GRF}\}_{d=1}^{D}$, $\boldsymbol{\psi}_d^{GRF} := \{\theta_{d1}, \theta_{d2}\}$ and noise variance σ given the traffic observations over T samples in time, i.e., $\mathcal{Y} := \{\mathbf{y}_{ot}\}_{t=1}^{T}$.

The posterior of $\mathbf{h}$ has the following form:

$$p(\mathbf{h}|\mathcal{Y}; \boldsymbol{\psi}) = \frac{1}{Z(\boldsymbol{\psi})}p(\mathbf{b}_o)p(\mathbf{a})\prod_{t=1}^{T} p(\mathbf{y}_{ot}|\mathbf{f}_t, \mathbf{W}_o, \mathbf{b}) \prod_{d=1}^{D} p(\mathbf{w}_{o,:d})p(\mathbf{f}_{:d}) \tag{23}$$

where $\boldsymbol{\psi} := \{\boldsymbol{\psi}^{GP}, \boldsymbol{\psi}^{GRF}, \sigma\}$ and $Z(\boldsymbol{\psi})$ is the marginal likelihood which only depends on the hyperparameters. The common approach for finding the best values for the hyperparameters is to maximize the log marginal likelihood as:

$$\boldsymbol{\psi}_{opt} = \arg\max_{\boldsymbol{\psi}} \log Z(\boldsymbol{\psi}). \tag{24}$$

However, neither of the problems in (23) and (24) are trivial to solve because computing the marginal likelihood involves multiple high dimensional integrals which are difficult to obtain in a closed form. To tackle the issue, we use the VB learning algorithm [30]. The VB is an iterative approach where each iteration consists of two main steps. In one step, we approximate the posterior distribution of $\mathbf{h}$ given the hyperparameters. In the next step, we optimize the hyperparameters given the approximate posterior distribution. In particular, at iteration i, we solve the following subproblems:

Subproblem 1) We approximate the posterior distribution by $q^{(i)}(\mathbf{h})$ given $\boldsymbol{\psi}^{(i-1)}$ as:

$$q^{(i)}(\mathbf{h}) \approx p(\mathbf{h}|\mathcal{Y}; \boldsymbol{\psi}^{(i-1)}). \tag{25}$$

The approximate distribution $q^{(i)}(\mathbf{h})$ is determined such that to have minimum dissimilarity with the true posterior. Using the KL divergence to measure this dissimilarity, $q^{(i)}(\mathbf{h})$ can be found by solving the following optimization problem:

$$\min_{q^{(i)}(\mathbf{h})} \text{KL}\left(q^{(i)}\left(\mathbf{h}\right) || \, p\left(\mathbf{h}|\mathcal{Y}; \boldsymbol{\psi}^{(i-1)}\right)\right), \quad s.t: \int q^{(i)}\left(\mathbf{h}\right) d\mathbf{h} = 1, \tag{26}$$

where $\text{KL}\left(q\left(.\right) || \, p\left(.\right)\right) := \mathbb{E}_{q(.)}\left\{log\frac{q(.)}{p(.)}\right\}$.

Subproblem 2) We optimize the hyperparameters by maximizing a lower bound of the log marginal likelihood. It can be shown that the negative of KL divergence in (26) is an upper bound for the log marginal likelihood [31].

Theretofore, the KL divergence in (26) can be used to optimize the hyperparameters as:

$$\boldsymbol{\psi}^{(i)} = \arg\max_{\boldsymbol{\psi}} -\mathrm{KL}\left(q^{(i)}\left(\mathbf{h}\right) \,||\, p\left(\mathbf{h}|\mathcal{Y};\boldsymbol{\psi}\right)\right). \tag{27}$$

We now explain the VB learning algorithm in more detail. Hereafter, we ignore superscript i for notational simplicity.

1) POSTERIOR APPROXIMATION

To simplify the optimization problem in (26), the assumption will be that the probability density function $q(\mathbf{h})$ is factorized with respect to each variable in $\mathbf{h}$ as:

$$\begin{aligned} q(\mathbf{h}) &= q(\mathbf{b}_o)q(\mathbf{a})q(\mathbf{F})q(\mathbf{W}_o) \\ &= q(\mathbf{b}_o)q(\mathbf{a})\prod_d q(\mathbf{f}_{:,d})q(\mathbf{w}_{o,:,d}). \end{aligned} \tag{28}$$

Note that the second equality is obtained without any further assumption. In particular, when we assume $\mathbf{W}_o$ and $\mathbf{F}$ are independent, they are automatically factorized over latent dimensions d due to the factorized form of their prior. Using the Karush–Kuhn–Tucker (KKT) conditions, it can be shown that the optimized form of j factor based on the minimization of (26) is given by [30]:

$$q\left(\mathbf{h}_j\right) \propto \exp\left(\mathbb{E}_{\sim q(\mathbf{h}_j)}\left[\log\left(p\left(\mathbf{h},\mathcal{Y}\right)\right)\right]\right), \tag{29}$$

where the notation $\mathbb{E}_{\sim q(\mathbf{h}_j)}[.]$ means to take the expectation with respect to all the variables except $\mathbf{h}_j$. Each optimal variational distribution can be obtained as in the following.

- Compute $q(\mathbf{f}_{:,d})$: The optimal un-normalized log variational density can be written as:

$$\begin{aligned} &\log q(\mathbf{f}_{:,d}) \\ &\propto \mathbb{E}_{\sim q(\mathbf{f}_{:,d})}\left[\sum_{t=1}^{T}\log p(\mathbf{y}_{ot}|\mathbf{W}_o, f_{dt}, \mathbf{b}_o) + \log p(\mathbf{f}_{:,d})\right] \\ &\propto -\frac{1}{2}\mathbf{f}_{:,d}^T\mathbf{C}_{w_d}\mathbf{f}_{:,d} - \frac{1}{2}\mathbf{f}_{:,d}^T\mathbf{K}_d^{-1}\mathbf{f}_{:,d} + \mathbf{c}_{w_d}\mathbf{f}_{:,d}, \end{aligned} \tag{30}$$

where we define $\mathbf{C}_{w_d} := \mathbb{E}_{\sim q(\mathbf{f}_{:,d})}\left[\frac{1}{\sigma}\mathbf{w}_{o,:,d}^T\mathbf{w}_{o,:,d}\right]\mathbf{I}_T$, $\mathbf{c}_{w_d} := \mathbb{E}_{\sim q(\mathbf{f}_{:,d})}\left[\frac{1}{\sigma}\mathbf{w}_{o,:,d}^T\mathbf{R}_d\right]$, $\mathbf{R}_d := [\mathbf{r}_{d1}, \ldots, \mathbf{r}_{dT}]$ and $\mathbf{r}_{dt} := \mathbf{y}_{ot} - \mathbf{b}_o - \sum_{d'\neq d}\mathbf{w}_{o,:,d'}f_{d't}$. It can be seen that (30) is in a form of a normal density given by:

$$q(\mathbf{f}_{:d}) = \mathcal{N}(\boldsymbol{\mu}_{f_d}, \Sigma_{f_d}), \tag{31}$$

where $\Sigma_{f_d} := (\mathbf{C}_{w_d} + \mathbf{K}_d^{-1})^{-1}$ and $\boldsymbol{\mu}_{f_d} := \Sigma_{f_d}\mathbf{c}_{w_d}^T$.

- Compute $q(\mathbf{w}_{o,:d})$: The un-normalized log variational density can be written as:

$$\begin{aligned} &\log q(\mathbf{w}_{o,:d}) \propto \\ &\quad \mathbb{E}_{\sim q(\mathbf{w}_{o,:d})}\left[\sum_{t=1}^{T}\log p(\mathbf{y}_{ot}|\mathbf{W}_o, \mathbf{f}_t, \mathbf{b}_o) + \log p(\mathbf{w}_{o,:d})\right] \\ &\propto -\frac{1}{2}\mathbf{w}_{o,:d}^T\mathbf{C}_d^f\mathbf{w}_{o,:d} - \frac{1}{2}\mathbf{w}_{o,:d}^T\mathbf{Q}_{o,d}\mathbf{w}_{o,:d} + \mathbf{c}_d^f\mathbf{w}_{o,:d}, \end{aligned} \tag{32}$$

where $\mathbf{C}_{f_d} := \mathbb{E}_{\sim q(\mathbf{w}_{o,:d})}\left[\frac{1}{\sigma}\mathbf{f}_{:d}^T\mathbf{f}_{:d}\right]\mathbf{I}_{M_o}$, $\mathbf{c}_{f_d} := \mathbb{E}_{\sim q(\mathbf{w}_{o,:d})}\left[\frac{1}{\sigma}\mathbf{f}_{:d}^T\mathbf{E}_d\right]$, $\mathbf{E}_d := \left[\mathbf{e}_{d1}, \ldots, \mathbf{e}_{dM_o}\right]$, $\mathbf{e}_{dm} := \bar{\mathbf{y}}_m - b_m - \sum_{d'\neq d}w_{d'm}\mathbf{f}_{:d'}$, and $\bar{\mathbf{y}}_m := [y_{om1}, \ldots, y_{omT}]^T$. It can be seen that (32) is in a form of a normal density given by:

$$q(\mathbf{w}_{o,:d}) = \mathcal{N}(\boldsymbol{\mu}_{w_d}, \Sigma_{w_d}), \tag{33}$$

where $\Sigma_{w_d} := (\mathbf{C}_{f_d} + \mathbf{Q}_{o,d})^{-1}$ and $\boldsymbol{\mu}_{w_d} := \Sigma_{w_d}\mathbf{c}_{f_d}^T$.

- Compute $q(\mathbf{b}_o)$: The un-normalized log variational density can be written as

$$\log q(\mathbf{b}_o) \propto -\frac{1}{2}\mathbf{b}_o^T\mathbf{C}_b\mathbf{b}_o + \mathbf{b}_o^T\mathbf{c}_b - \frac{1}{2}\mathbf{b}_o^T\mathbf{Q}_{o,D+1}\mathbf{b}_o, \tag{34}$$

where $\mathbf{C}_b = \frac{1}{\sigma}\mathbf{I}_{M_o}$, $\mathbf{c}_b = \frac{1}{\sigma}(\sum_{t=1}^{T}\mathbf{y}_{ot} - \mathbb{E}_{\sim q(\mathbf{b}_o)}[\mathbf{W}_o\mathbf{f}_t]) + \mathbf{Q}_{o,D+1}\mathbf{Z}_o\mathbb{E}_{q(\mathbf{a})}[\mathbf{a}]$. It can be seen that (34) is in a form of a normal density given by:

$$q(\mathbf{b}_o) = \mathcal{N}(\boldsymbol{\mu}_b, \Sigma_b), \tag{35}$$

where $\Sigma_b = (\mathbf{C}_b + \mathbf{Q}_{o,D+1})^{-1}$ and $\boldsymbol{\mu}_b = \Sigma_b\mathbf{c}_b$.

- Compute $q(\mathbf{a})$: The un-normalized log variational density can be written as

$$\begin{aligned} \log q(\mathbf{a}) \propto &-\frac{1}{2}\mathbf{a}^T\mathbf{Z}_o^T\mathbf{Q}_{o,D+1}\mathbf{Z}_o\mathbf{a} \\ &+\mathbf{a}^T\mathbf{Z}_o^T\mathbf{Q}_{o,D+1}\mathbb{E}_{q(\mathbf{b}_o)}[\mathbf{b}_o] - \frac{1}{2}\mathbf{a}^T\mathbf{a}, \end{aligned} \tag{36}$$

which is in a form of a normal density given by:

$$q(\mathbf{a}) = \mathcal{N}(\boldsymbol{\mu}_a, \Sigma_a), \tag{37}$$

where $\Sigma_a = (\mathbf{Z}_o^T\mathbf{Q}_{o,D+1}\mathbf{Z}_o + \mathbf{I})^{-1}$ and $\boldsymbol{\mu}_a = \Sigma_a(\mathbf{Z}_o^T\mathbf{Q}_{o,D+1}\mathbb{E}_{q(\mathbf{b}_o)}[\mathbf{b}_o])$.

2) HYPERPARAMETER OPTIMIZATION

In this step, the goal is to optimize the hyper-parameters given the posterior distribution. Using the optimization problem in (27), the objective functions for the GP, GRF, parameters and σ to be minimized can be written as:

$$\begin{aligned} &L_1(\boldsymbol{\psi}_d^{GP}) \\ &= \frac{1}{2}\log|\mathbf{K}_d| - \frac{1}{2}tr(\mathbf{K}_d^{-1}\mathbb{E}[\mathbf{f}_{:d}^T\mathbf{f}_{:d}]), \forall d = 1, \ldots \end{aligned} \tag{38}$$

$$\begin{aligned} D \quad &L_2(\boldsymbol{\psi}_d^{GRF}) \\ &= -\frac{1}{2}\log|\mathbf{Q}_{o,d}| - \frac{1}{2}tr(\mathbf{Q}_{o,d}\mathbb{E}[\mathbf{w}_{o:d}^T\mathbf{w}_{o:d}]), \forall d = 1, \ldots D \end{aligned} \tag{39}$$

$$L_3(\boldsymbol{\psi}_{D+1}^{GRF}) = -\frac{1}{2}\log|\mathbf{Q}_{o,D+1}| - \frac{1}{2}tr(\mathbf{Q}_{o,D+1}\mathbb{E}[\mathbf{b}_o^T\mathbf{b}_o]), \tag{40}$$

$$L_4(\sigma) = -\frac{M_oT}{2}\log\sigma - \frac{1}{2\sigma}\sum_{t=1}^{T}\mathbb{E}[\mathbf{r}_t^T\mathbf{r}_t], \tag{41}$$

where $\mathbf{r}_t = \mathbf{y}_{ot} - \mathbf{W}_o \mathbf{f}_t - \mathbf{b}_o$. The maximal value of the objective (41) can be obtained by taking its gradient equal to zero which is given by:

$$\sigma = \frac{1}{M_o T} \sum_{t=1}^{T} \mathbb{E}[\mathbf{r}_t^T \mathbf{r}_t]. \tag{42}$$

Due to non-linear dependency of GP and GRF objective functions to the hyperparameters, there are no-closed form solutions for (39), (39) and (40). However, the functions are differentiable with respect the hyper-parameters and gradient-based methods (e.g., Newton method) can be used in order to minimize the objective functions.

Overall, the inference algorithm for posterior approximation and hyperparameter optimization takes the form as in Alg. 1.

B. PREDICTION

For the H-step ahead prediction, we need to compute two types of posterior predictive distributions (PPDs), one for the observed nodes and one for the missing nodes.

1) OBSERVED NODES

The PPD for the observed node is given by:

$$\begin{aligned} p(\mathbf{Y}_o^* | \mathcal{Y}) = &\int p(\mathbf{Y}_o^* | \mathbf{W}_o, \mathbf{F}^*, \mathbf{b}_o) p(\mathbf{F}^* | \mathbf{F}) q(\mathbf{F}) \\ & q(\mathbf{W}_o) q(\mathbf{b}_o) d\mathbf{F}^* d\mathbf{W}_o d\mathbf{F} d\mathbf{b}_o, \end{aligned} \tag{43}$$

where $\mathbf{Y}_o^* = \left[\mathbf{y}_{o,T+1}, \ldots, \mathbf{y}_{o,T+H}\right]$ and $\mathbf{F}^* = \left[\mathbf{f}_{T+1}, \ldots, \mathbf{f}_{T+H}\right]$. The conditional $p(\mathbf{F}_* | \mathbf{F})$ density can by computing by noting that the joint $p(\mathbf{F}, \mathbf{F}_*)$ is a normal as:

$$p(\mathbf{f}_{:d}, \mathbf{f}_{:d}^*) = \mathcal{N}(\mathbf{0}, [\begin{matrix} \mathbf{K}_d & \mathbf{K}_d^* \\ \mathbf{K}_d^{*T} & \mathbf{K}_d^{**} \end{matrix}]), \;\; \forall d = 1, \ldots, D, \tag{44}$$

where $[\mathbf{K}_d^*]_{ij} = K_d(i,j), \forall i \in \{1, \ldots, T\}, j \in \{T+1, \ldots, T+H\}$, and $[\mathbf{K}_d^{**}]_{ij} = K_d(i,j), \forall i,j \in \{T+1, \ldots, T+H\}$. Using the conditional property of normal [29], we obtain:

$$\begin{aligned} p(\mathbf{f}_{:d}^* | \mathbf{f}_{:d}) &= \mathcal{N}(\mathbf{M}_d \mathbf{f}_{:d}, \mathbf{P}_d), \\ \mathbf{M}_d &= \mathbf{K}_d^{*T} \mathbf{K}_d^{-1}, \mathbf{P}_d = \mathbf{K}_d^{**} - \mathbf{K}_d^{*T} \mathbf{K}_d^{-1} \mathbf{K}_d^*. \end{aligned} \tag{45}$$

We can integrate out the latent variables $\mathbf{f}_{:d}$ and $\mathbf{b}_o$ to simplify (43) as:

$$\begin{aligned} p(\mathbf{Y}^* | \mathcal{Y}) = &\int \prod_{t=T+1}^{T+H} \mathcal{N}(\mathbf{y}_t | \mathbf{W}_o \mathbf{f}_t^* + \boldsymbol{\mu}_b, \sigma \mathbf{I} + \Sigma_b) p(\mathbf{f}_{:d}^*) \\ & \times q(\mathbf{W}_o) d\mathbf{f}_{:d}^* d\mathbf{W}_o, \end{aligned} \tag{46}$$

where

$$\begin{aligned} p(\mathbf{f}_{:d}^*) &= \mathcal{N}(\boldsymbol{\mu}_{f_d^*}, \Sigma_{f_d^*}), \\ \boldsymbol{\mu}_{f_d^*} &= \mathbf{M}_d \boldsymbol{\mu}_{f_d}, \Sigma_{f_d^*} = \mathbf{M}_d \mathbf{P}_d \mathbf{M}_d^T + \Sigma_{f_d}. \end{aligned} \tag{47}$$

It is difficult to obtain a closed-form expression of (46). However, the mean and the variance can be computed analytically as:

$$\mathbb{E}(\mathbf{Y}^* | \mathcal{Y}) = \sum_{d=1}^{D} \boldsymbol{\mu}_{w_d} \boldsymbol{\mu}_{f_d^*}^T + \boldsymbol{\mu}_b, \tag{48}$$

$$\begin{aligned} var(y_{mt} | \mathcal{Y}) = \sigma + [\Sigma_b]_{m,m} + &\sum_{d=1}^{D} [\boldsymbol{\mu}_{w_d}]_m^2 [\Sigma_{f_d^*}]_{t,t}. \\ &+ [\boldsymbol{\mu}_{f_d^*}]_t^2 [\Sigma_{w_d}]_{m,m} + [\Sigma_{w_d}]_{m,m} [\Sigma_{f_d^*}]_{t,t}. \end{aligned} \tag{49}$$

2) MISSING NODES

The PPD for the missing nodes is given by:

$$\begin{aligned} p(\mathbf{Y}_{\bar{o}}^* | \mathcal{Y}) = & \\ &\int p(\mathbf{Y}_{\bar{o}}^* | \mathbf{W}_{\bar{o}}, \mathbf{F}^*, \mathbf{b}_{\bar{o}}) p(\mathbf{F}^*) p(\mathbf{W}_{\bar{o}} | \mathbf{W}_o) q(\mathbf{a}) \\ & p(\mathbf{b}_{\bar{o}} | \mathbf{b}_o, \mathbf{a}, \mathbf{Z}_{\bar{o}}) q(\mathbf{b}_o) q(\mathbf{W}_o) d\mathbf{f}^* \\ & \times d\mathbf{W}_o d\mathbf{b}_o d\mathbf{b}_{\bar{o}} d\mathbf{W}_{\bar{o}}, \end{aligned} \tag{50}$$

where $\mathbf{Y}_{\bar{o}}^* = \left[\mathbf{y}_{\bar{o},T+1}, \ldots, \mathbf{y}_{\bar{o},T+H}\right]$ is the future traffic of the missing nodes and $p(\mathbf{F}^*)$ is the marginal predictive distribution of the time latent variables defined in (47). To compute (50), we need to obtain the distributions of the missing nodes latent factors $\mathbf{W}_{\bar{o}}$ and biases $\mathbf{b}_{\bar{o}}$ conditioned on the inferred parameters, which are respectively represented by $p(\mathbf{W}_{\bar{o}} | \mathbf{W}_o)$ and $p(\mathbf{b}_{\bar{o}} | \mathbf{b}_o, \mathbf{a}, \mathbf{Z}_{\bar{o}})$. The latent variables $\mathbf{W}_{\bar{o}}$, $\mathbf{b}_{\bar{o}}$ and $\mathbf{F}^*$ are subsequently used as the parameters of the distribution of the missing nodes $p(\mathbf{Y}_{\bar{o}}^* | \mathbf{W}_{\bar{o}}, \mathbf{F}^*, \mathbf{b}_{\bar{o}})$ to generate the data traffic. Note the this distribution is similar to (28) and is given by:

$$p(\mathbf{Y}_{\bar{o}}^* | \mathbf{W}_{\bar{o}}, \mathbf{F}^*, \mathbf{b}_{\bar{o}}) = \prod_{t=T+1}^{T+H} \mathcal{N}(\mathbf{y}_{\bar{o}t} | \mathbf{W}_{\bar{o}} \mathbf{f}_t^* + \mathbf{b}_{\bar{o}}, \sigma \mathbf{I}) \tag{51}$$

In order to simplify the PPD in (50), we compute the marginal predictive distribution of the missing nodes latent variables by integrating out all the inferred latent variables of the observed nodes. In the following, we derive these marginal distributions.

Using the GRF model in (20), the conditional density $p(\mathbf{W}_{\bar{o}} | \mathbf{W}_o)$ can be computed as:

$$p(\mathbf{w}_{\bar{o},:d} | \mathbf{w}_{o,:d}) = \mathcal{N}(\mathbf{M}_{w_d} \mathbf{w}_{o,:d}, \mathbf{P}_{w_d}), \forall d = 1, \ldots, D, \tag{52}$$

where $\mathbf{M}_{w_d} = \mathbf{Q}_{o\bar{o}d}^T \mathbf{Q}_{od}^{-1}$, $\mathbf{P}_{w_d} = \mathbf{Q}_{o\bar{o}d}^T \mathbf{Q}_{od}^{-1} \mathbf{Q}_{o\bar{o}d}$ and $\mathbf{Q}_{o\bar{o}d} = -\mathbf{Q}_{11d}^{-1} \mathbf{Q}_{12d} \mathbf{Q}_{\bar{o}d}^{-1}$. The marginal predictive $\mathbf{W}_{\bar{o}}$ can be computed by integrating out $\mathbf{W}_o$ as:

$$\begin{aligned} p(\mathbf{w}_{\bar{o},:d}) &= \mathcal{N}(\boldsymbol{\mu}_{w_{\bar{o}d}}, \Sigma_{w_{\bar{o}d}}), \\ \boldsymbol{\mu}_{w_{\bar{o}d}} &= \mathbf{M}_{w_d} \boldsymbol{\mu}_{w_d}, \Sigma_{w_d} = \mathbf{M}_{w_d} \mathbf{P}_{w_d} \mathbf{M}_{w_d}^T + \Sigma_{w_d}. \end{aligned} \tag{53}$$

Similarly, we can compute the marginal predictive distribution of $\mathbf{b}_{\bar{o}}$ as:

$$p(\mathbf{b}_{\bar{o}}) = \mathcal{N}(\boldsymbol{\mu}_{b_{\bar{o}}}, \Sigma_{b_{\bar{o}}}), \tag{54}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{b_{\bar{o}}} &= \mathbf{Z}_{\bar{o}} \boldsymbol{\mu}_a + \mathbf{M}_b (\boldsymbol{\mu}_{b_o} - \mathbf{Z}_o \boldsymbol{\mu}_a) \\ \Sigma_{b_{\bar{o}}} &= \mathbf{M}_a \Sigma_a \mathbf{M}_a + \bar{\mathbf{P}}_b \\ \bar{\mathbf{P}}_b &= \mathbf{M}_b \Sigma_{b_o} \mathbf{M}_b + \mathbf{P}_b \end{aligned}$$

Algorithm 1 The VB Algorithm

```
Initialize hyperparameters $\boldsymbol{\psi}^{(0)}$ and posterior parameters $\boldsymbol{\mu}_\ell^{(0)}, \Sigma_\ell^{(0)}, \forall \ell = a, b, f_1, \ldots, f_D, w_1, \ldots, w_D$;
Set $i = 1$;
repeat
    · Approximate the posterior distribution:
    for $d \leftarrow 1, D$ do                      ▷ Update the parameters of $q(\mathbf{f}_{:d}), \forall d = 1, \ldots D$
        $\Sigma_{f_d}^{(i)} = (\mathbf{C}_{w_d} + \mathbf{K}_d^{-1})^{-1}, \boldsymbol{\mu}_{f_d}^{(i)} = \Sigma_{f_d}^{(i)} \mathbf{c}_{w_d}^T$;
where $\mathbf{C}_{w_d} = \mathbf{I}_T \frac{1}{\sigma^{(i-1)}} \sum_{m=1}^{M_o} [\Sigma_{wd}^{(i-1)}]_{m,m}$, $\mathbf{c}_{w_d} = \frac{1}{\sigma^{(i-1)}} \boldsymbol{\mu}_{w_d}^{(i-1)^T} \bar{\mathbf{R}}_d$, $\bar{\mathbf{R}}_d = [\bar{\mathbf{r}}_{d1}, \ldots, \bar{\mathbf{r}}_{dT}]$, $\bar{\mathbf{r}}_{dt} = \mathbf{y}_{ot} - \boldsymbol{\mu}_b^{(i-1)} - \sum_{d' \neq d} \boldsymbol{\mu}_{w_{d'}}^{(i-1)} [\boldsymbol{\mu}_{f_{d'}}^{(i-1)}]_t$;
    end for
    for $d \leftarrow 1, D$ do                      ▷ Update the parameters of $q(\mathbf{w}_{o,:d}), \forall d = 1, \ldots D$
        $\Sigma_{w_d}^{(i)} = (\mathbf{C}_{f_d} + \mathbf{Q}_{o,d})^{-1}, \boldsymbol{\mu}_{w_d}^{(i)} = \Sigma_{w_d}^{(i)} \mathbf{c}_{f_d}^T$;
where $\mathbf{C}_{f_d} = \mathbf{I}_{M_o} \frac{1}{\sigma^{(i-1)}} \sum_{t=1}^{T} [\Sigma_{f_d}^{(i-1)}]_{t,t}$, $\mathbf{c}_{f_d} = \frac{1}{\sigma^{(i-1)}} \boldsymbol{\mu}_{f_d}^{(i-1)^T} \bar{\mathbf{E}}_d$, $\bar{\mathbf{E}}_d = \left[\bar{\mathbf{e}}_{d1}, \ldots, \bar{\mathbf{e}}_{dM_o}\right]$, $\bar{\mathbf{e}}_{dm} = \bar{\mathbf{y}}_m - [\boldsymbol{\mu}_b^{(i-1)}]_m - \sum_{d' \neq d} [\boldsymbol{\mu}_{w_{d'}}^{(i-1)}]_m \boldsymbol{\mu}_{f_{d'}}^{(i-1)}$;
    end for
    $\Sigma_b^{(i)} = (\mathbf{C}_b + \mathbf{Q}_{o,D+1})^{-1}, \boldsymbol{\mu}_b^{(i)} = \Sigma_b^{(i)} \mathbf{c}_b$;          ▷ Update the parameters of $q(\mathbf{b}_o)$
where $\mathbf{C}_b = \mathbf{I}_{M_o} \frac{1}{\sigma^{(i-1)}}$, $\mathbf{c}_b = \frac{1}{\sigma^{(i-1)}} (\sum_{t=1}^{T} \mathbf{y}_{ot} - \bar{\mathbf{W}}_o \bar{\mathbf{f}}_t) + \mathbf{Q}_{o,D+1} \mathbf{Z}_o \boldsymbol{\mu}_a^{(i-1)}$, $\bar{\mathbf{f}}_t = [[\boldsymbol{\mu}_{f_1}^{(i)}]_t, .., [\boldsymbol{\mu}_{f_D}^{(i)}]_t]^T$, $\bar{\mathbf{W}}_o = [\boldsymbol{\mu}_{w_1}^{(i)}, \ldots, \boldsymbol{\mu}_{w_D}^{(i)}]$;
    $\Sigma_a^{(i)} = (\mathbf{Z}_o^T \mathbf{Q}_{o,D+1} \mathbf{Z}_o + \mathbf{I})^{-1}, \boldsymbol{\mu}_a^{(i)} = \Sigma_a^{(i)} (\mathbf{Z}_o^T \mathbf{Q}_{o,D+1} \boldsymbol{\mu}_b^{(i)})$;          ▷ Update the parameters of $q(\mathbf{a})$
    · Optimize the hyperparameters                  ▷ Update $\boldsymbol{\psi}$
    for $d \leftarrow 1, D$ do
        $\boldsymbol{\psi}_d^{GP(i)} = \arg\min_{\boldsymbol{\psi}_d^{GP}} \frac{1}{2} \log |\mathbf{K}_d| - \frac{1}{2} tr(\mathbf{K}_d^{-1} (\boldsymbol{\mu}_{f_d}^{(i)^T} \boldsymbol{\mu}_{f_d}^{(i)} + \Sigma_{f_d}^{(i)}))$;
        $\boldsymbol{\psi}_d^{GRF(i)} = \arg\min_{\boldsymbol{\psi}_d^{GRF}} -\frac{1}{2} \log |\mathbf{Q}_{o,d}| - \frac{1}{2} tr(\mathbf{Q}_{o,d} (\boldsymbol{\mu}_{w_d}^{(i)^T} \boldsymbol{\mu}_{w_d}^{(i)} + \Sigma_{w_d}^{(i)}))$;
    end for
    $\boldsymbol{\psi}_{D+1}^{GRF(i)} = \arg\min_{\boldsymbol{\psi}_{D+1}^{GRF}} -\frac{1}{2} \log |\mathbf{Q}_{o,D+1}| - \frac{1}{2} tr(\mathbf{Q}_{o,D+1} (\boldsymbol{\mu}_b^{(i)^T} \boldsymbol{\mu}_b^{(i)} + \Sigma_b^{(i)}))$;
    $\sigma^{(i)} = \frac{1}{M_o T} (\sum_{t=1}^{T} (\mathbf{r}_t^T \mathbf{r}_t + \boldsymbol{\zeta}_{w_d} \boldsymbol{\zeta}_{f_t}^T) + \zeta_b)$;
where $\mathbf{r}_t = \mathbf{y}_{ot} - \bar{\mathbf{W}}_o \bar{\mathbf{f}}_t - \boldsymbol{\mu}_b^{(i)}$, $\zeta_{wd} = \sum_{m=1}^{M_o} [\Sigma_{wd}^{(i)}]_{m,m}$, $\boldsymbol{\zeta}_{w_d} = [\zeta_{w_1}, .., \zeta_{w_D}]$, $\boldsymbol{\zeta}_{f_t} = [[\Sigma_{f_1}^{(i)}]_{t,t}, .., [\Sigma_{f_D}^{(i)}]_{t,t}]$, $\zeta_b = \sum_{m=1}^{M_o} [\Sigma_b^{(i)}]_{m,m}$;
    Recompute $\mathbf{K}_d, \forall d = 1, ..D$, and $\mathbf{Q}_d, \forall d = 1, ..D + 1$, using the updated hyperparameters $\boldsymbol{\psi}^{(i)}$;
until convergence
```

$$
\begin{aligned}
\mathbf{M}_a &= \mathbf{Z}_{\bar{o}} - \mathbf{M}_b \mathbf{Z}_o \\
\mathbf{M}_b &= \mathbf{Q}_{o\bar{o},D+1}^T \mathbf{Q}_{o,D+1}^{-1} \\
\mathbf{P}_b &= \mathbf{Q}_{o\bar{o},D+1}^T \mathbf{Q}_{o,D+1}^{-1} \mathbf{Q}_{o\bar{o},D+1} \\
\mathbf{Q}_{o\bar{o},D+1} &= -\mathbf{Q}_{11,D+1}^{-1} \mathbf{Q}_{12,D+1} \mathbf{Q}_{o,D+1}^{-1}.
\end{aligned}
$$

Using (54), (53) and by integrating out $\mathbf{a}$, the integral in (50) can be simplified as:

$$
p(\mathbf{Y}_o^* | \mathcal{Y}) = \int \prod_{t=T+1}^{T+H} \mathcal{N}(\mathbf{y}_{\bar{o}t} | \mathbf{W}_{\bar{o}} \mathbf{f}_t^* + \boldsymbol{\mu}_{b_{\bar{o}}}, \sigma \mathbf{I} + \Sigma_{b_{\bar{o}}}) \\
p(\mathbf{f}_{:d}^*) p(\mathbf{W}_{\bar{o}}) d\mathbf{f}_{:d}^* d\mathbf{W}_{\bar{o}}, \tag{55}
$$

which can be seen that it has a similar form as (46). Finally, the mean and the variance are computed as:

$$
\mathbb{E}(\mathbf{Y}_{\bar{o}}^* | \mathcal{Y}) = \sum_{d=1}^{D} \boldsymbol{\mu}_{w_{\bar{o}d}} \boldsymbol{\mu}_{f_d^*}^T + \boldsymbol{\mu}_{b_{\bar{o}}}, \tag{56}
$$

$$
var(y_{\bar{o},mt} | \mathcal{Y}) = \sigma + [\Sigma_{b_{\bar{o}}}]_{m,m} + \sum_{d=1}^{D} [\boldsymbol{\mu}_{w_{\bar{o}d}}]_m^2 [\Sigma_{f_d^*}]_{t,t} \\
+ [\boldsymbol{\mu}_{f_d^*}]_t^2 [\Sigma_{w_{\bar{o}d}}]_{m,m} + [\Sigma_{w_{\bar{o}d}}]_{m,m} [\Sigma_{f_d^*}]_{t,t}. \tag{57}
$$

Overall, the workflow of posterior and prediction computations can be summarized in Fig. 3. In particular, first, the recorded data traffic at the observed nodes are loaded. Next, in the preprocessing step, we scale down the data. This step is essential because the raw data can have large values and rescaling can help better convergence of the VB algorithm. Subsequently, using Alg. 1, we compute the posterior distribution. Finally, using (43) and (50), the traffic flow at the missing and the observed nodes are predicted. The per-iteration computational complexity of approximating the posterior distribution is $\mathcal{O}(DT^3 + DM^3)$ which is mainly due to matrix inversion operations. The computational complexity of prediction is the same.

VI. SIMULATION RESULTS

In this section, we evaluate the prediction accuracy of the proposed graph-based GP (GGP) model using the following two real-world network traffic datasets.

TABLE 2. The MSE performance for different prediction ahead, *H*.

	Abilen dataset					Geant dataset				
H	GGP	ARIMA	LSTM	GCRN	GMAN	GGP	ARIMA	LSTM	GCRN	GMAN
1	**4.4524**	4.8670	5.5116	6.0652	5.4130	**0.0077**	0.0147	0.0222	0.0228	0.0237
5	51.6860	52.269	**49.514**	53.835	53.314	**0.0301**	0.0326	0.0453	0.0468	0.0575
15	**30.565**	31.965	38.8977	36.4435	39.4435	**0.4800**	0.5172	0.7442	0.6925	0.6963
30	**42.7802**	44.170	44.0926	43.6712	45.5801	1.8624	1.9388	2.9168	1.6046	**1.5810**

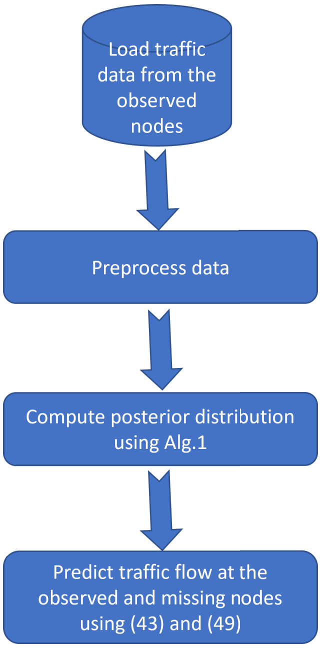
Load traffic data from the observed nodes
Preprocess data
Compute posterior distribution using Alg.1
Predict traffic flow at the observed and missing nodes using (43) and (49)

FIGURE 3. The workflow of posterior and prediction computations.

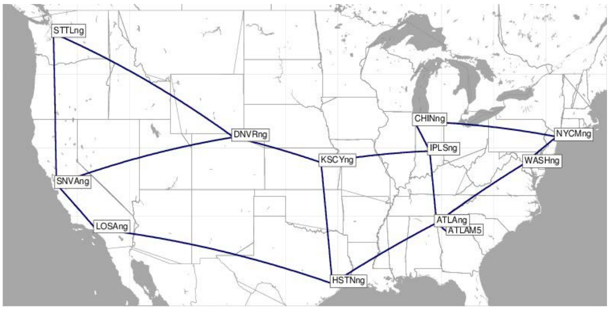

FIGURE 4. The abilene network topology.

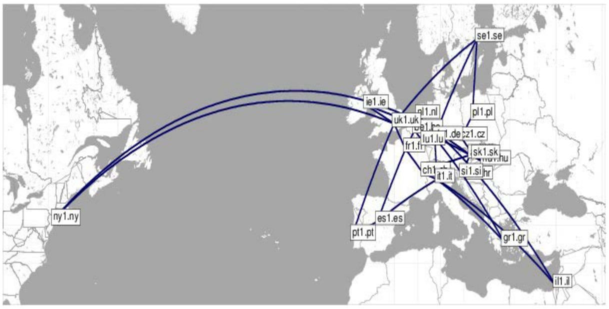

FIGURE 5. The geant network topology.

- Abilene dataset [32]: The Abilene was a high-performance backbone network created by the Internet2 community in the late 1990s. The network consists of 12 routers in United States which are connected by bidirectional links, as shown in Fig. 4. The traffic data on each link were recorded every 5 minutes (in Mb) from 2004/03/01 to 2004/09/10. In the original dataset, the traffic was measured on each link which are in total 30 links. We aggregate the data traffic of the incoming links to each node in order to obtain node-level traffic measurement. We further aggregate the dataset on an hourly basis.
- Geant dataset [32]: Geant is a pan-European data network. The network consists of 22 routers which are connected by bidirectional links, as shown in Fig. 4. The data were taken in 15 minutes steps starting on 04/05/2005 and ending on 31/08/2005. In the original dataset, the traffic was measured on each link which are in total 72 links. We aggregate the data traffic of the incoming links to each node in order to obtain node-level traffic measurement. We also aggregate this dataset on an hourly basis.

To help better convergence of learning procedure, the data have been scaled down by factor of 10000.

In order to compare with the proposed model, we consider the following benchmarks:

- ARIMA model: The model is implemented using the statsmodel python package [33]. We select the orders from the set $\{(1, 0, 1), (3, 0, 3), (5, 0, 5), (7, 0, 7)\}$ which has the best performance.
- LSTM [34]: A fully connected network with two recurrent layers. In each recurrent layer, we select the number of hidden unit form the the set $\{50, 100, 150, 200\}$ which yields the best performance.
- GCRN [16]: It consists of a stack of graph CNN and LSTM. The structure of the LSTM is selected similar the previous model.
- Graph Multi-Attention Network (GMAN) [35]: It is designed using attention mechanisms with gated fusion to model the spatio-temporal correlations. This model has three hyperparameters including the number of spatio-temporal attention blocks, the number of attention heads, and the dimensionality of each attention head. We fix the number of spatio-temporal attention blocks to 3 and number of attention heads to 8 as in [35]. We select the dimensionality of attention heads from the set $\{4, 8, 12, 16\}$ which yields the best performance.

Moreover, we train the neural network models, i.e., LSTM, GCRN and GMAN, using Adam optimizer [36] with an initial learning rate of 0.001.

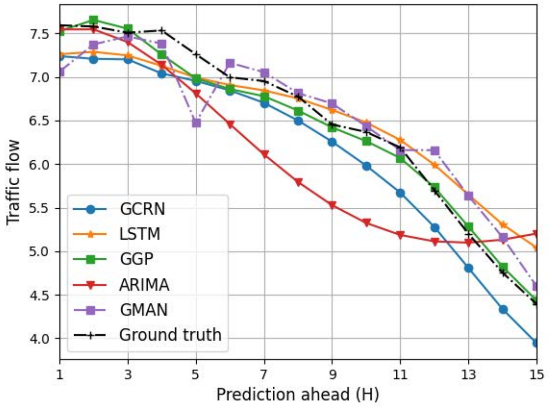

FIGURE 6. The ground truth and predicted traffic flow trajectories of node "ch1.ch" versus prediction ahead, *H*, for Geant dataset.

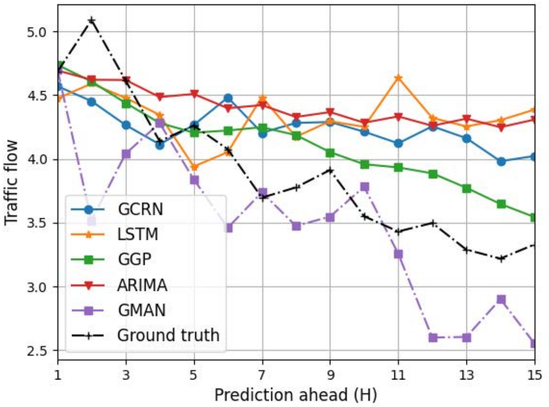

FIGURE 7. The ground truth and predicted traffic flow trajectories of node "ATLAng" versus prediction ahead, *H*, for Abilene dataset.

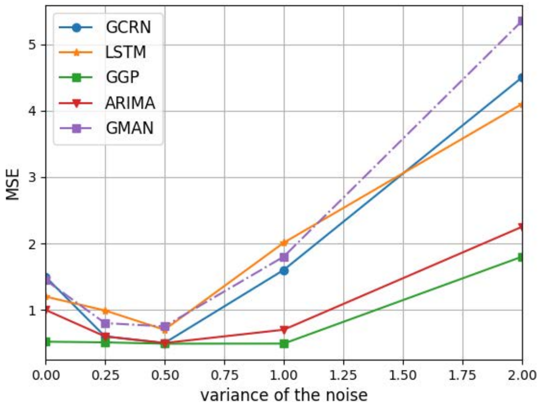

FIGURE 8. The MSE versus the variance of noise, $\bar{\sigma}$, added to the data traffic for Geant dataset.

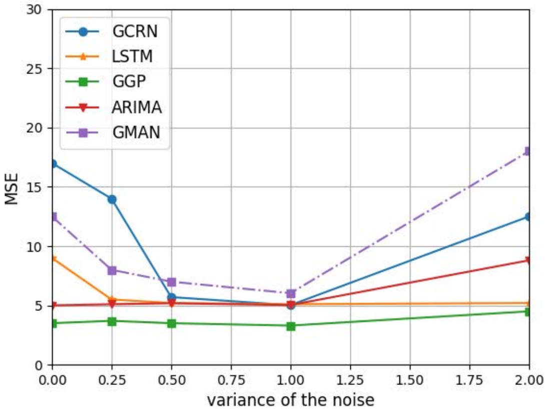

FIGURE 9. The MSE versus the variance of noise, $\bar{\sigma}$, added to the data traffic for Abilene dataset.

Table 2 shows the MSE for different prediction ahead length, H, for different models. Note that the MSE computes the discrepancy between the ground truth and the predicted traffic flow values. Therefore, a prediction model with smaller a MSE has better performance since its prediction is closer to the ground truth. In this experiment, we set $M_{\bar{o}} = 0$ and $D = 5$. We trained the models for $T = \{100, 200, 300, 400, 500\}$ observations and the predictions are computed for the subsequent H time steps for each T. The MSE is computed by taking predictions in all Ts. The table shows that the developed GGP has lower MSE with respect to the other benchmarks. Additionally, it can be observed that, in general, the MSE increases as H increases for all the models. This is expected since predicting a far-distant future is more challenging. Moreover, in Fig. 6 and Fig. 7, we show the ground truth and the predicted traffic flow trajectories versus prediction ahead H, for $T = 500$, for nodes "ATLAng" and "it1.it" in Abilene and Geant datasets respectively. We can see that the predicted traffic flow using GGP model is much closer to the ground truth with respect to the other models.

Further, we investigate the sensitivity of the models against noise. For this, we add artificial noise to the training datasets and train the models. In particular, we add a folded Gaussian noise with zero mean and variance $\bar{\sigma}$. Fig. 8 and Fig. 9 illustrate the MSE versus $\bar{\sigma}$ for Geant and Abilene datasets respectively. The results are obtained by averaging over 25 Monte Carlo simulations. It can be observed that the GGP model is much more robust against the noise with respect to the other models. In addition, we see that, in general, the MSE first decreases and then increases as the noise variance increases. The reason is that injecting small noise to the training dataset can reduce overfitting and improve the prediction accuracy. Note that applying noise to the training dataset is widely used in deep learning literature as a data augmentation technique to reduce overfitting [37]. However, increasing the noise variance eventually increases the MSE because the added noise dominates the underlying patterns in the data that are useful to capture for accurate prediction.

Next, we show the prediction error in the presence of missing nodes. We set $T = 300$, $D = 5$ and $H = 15$. For this experiment, it is not straightforward to implement the GCRN because it requires the data traffic of all the nodes. Therefore, we only use LSTM and ARIMA as benchmarks. For these models, we use an iterative mean imputation method to predict the data traffic at the missing nodes. In particular,

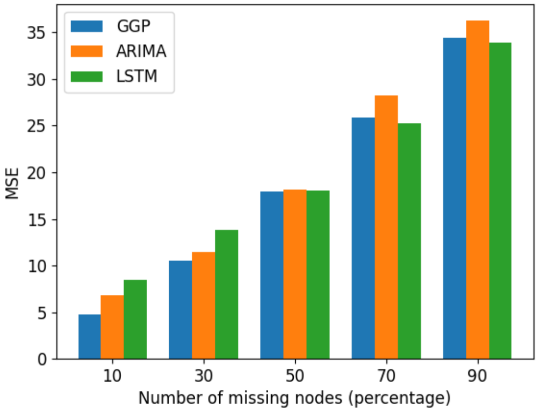

FIGURE 10. The MSE for different number of missing nodes for Abilene dataset.

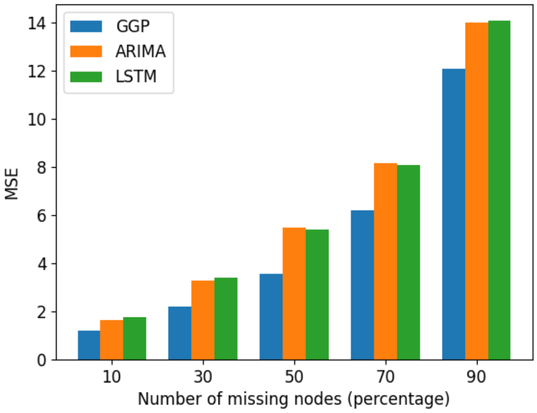

FIGURE 11. The MSE for different number of missing nodes for Geant dataset.

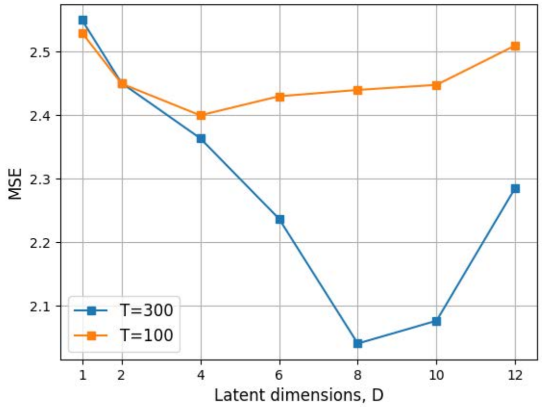

FIGURE 12. The MSE for different values latent dimensions *D* for Geant dataset.

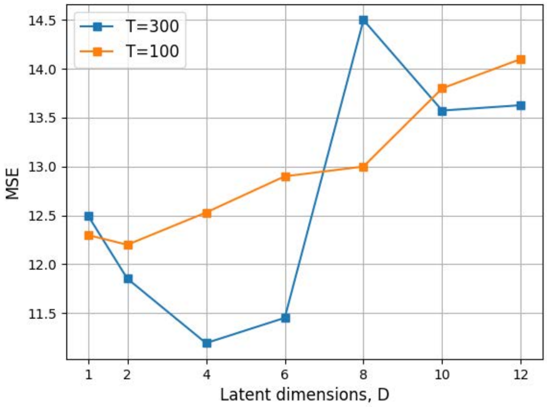

FIGURE 13. The MSE for different values latent dimensions *D* for Abilene dataset.

we fix the predictions at the observed nodes and initialize the missing nodes with zero values. Next, each missing node computes the mean of data traffic from its one-hop neighbors and sends the computed mean to its neighbors. This procedure is repeated until convergence. Fig. 10 and Fig. 11 depict the MSE versus the number of missing nodes, $M_{\bar{o}}$ for the Abilene and Geant networks respectively. The missing nodes are selected randomly, similar to the previous scenario, the results are obtained by averaging over 25 Monte Carlo simulations. We can observe that the developed GGP model outperforms the other benchmarks on both datasets. Moreover, it can be seen that as the number of missing nodes increases the MSE increases for all the models. This is expected because as $M_{\bar{o}}$ increases, the number of observations decreases and it becomes much more difficult to extract useful patterns among the nodes in the network.

Finally, we show the MSE versus the latent dimensions D for $T = 100$ and $T = 300$ in Fig. 12 and Fig. 13. In this experiment, the number of missing nodes is 30% of the total number of nodes. From the figures, we can see that, in general, the MSE first decreases and then slightly increases as D increases. The decreasing trend is because the model is not yet expressive enough to explain the data traffic well which results in to underfit. The increasing trend is because the model is too much flexible and fits to the noise rather than useful patterns which results in to overfit. Moreover, it can be seen that as the number of observations T increases, a larger D is required to achieve accurate prediction. For example, in Fig. 12, we can observe that 8 dimensions are enough for $T = 300$ while 4 dimensions are needed for $T = 100$.

VII. CONCLUSION

In this paper, we introduced a graph-based Gaussian process model for network traffic analysis and prediction when the network graph is not fully observable. The model is Bayesian non-parametric and highly flexible in capturing complex nonlinear patterns in the data. We defined a structured kernel function which can model long-term and short-term temporal trends for accurate prediction. Additionally, the model can learn spatial interactions among the node in the network which can be used to predict the missing values. Next, we developed a variational inference algorithm to approximate the intractable posterior distribution which can be efficiently implemented. Finally, using two real-world datasets, we showed simulation results to demonstrate that the proposed model can achieve improved prediction accuracy with respect to the state-of-the-art approaches.

REFERENCES

[1] A. Baiocchi, *Network Traffic Engineering Stochastic Models and Applications*. Hoboken, NJ, USA: Wiley, 2020.

[2] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, ''Network function virtualization: State-of-the-art and research challenges,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2015.

[3] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, ''Artificial neural networks-based machine learning for wireless networks: A tutorial,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[4] H. Feng and Y. Shu, ''Study on network traffic prediction techniques,'' in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, vol. 2, Sep. 2005, pp. 1041–1044.

[5] A. Azari, P. Papapetrou, S. Denic, and G. Peters, ''Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA,'' in *Proc. Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2019, pp. 129–144.

[6] T. Karagiannis, M. Molle, and M. Faloutsos, ''Long-range dependence ten years of internet traffic modeling,'' *IEEE Internet Comput.*, vol. 8, no. 5, pp. 57–64, Oct. 2004.

[7] C. Chen, J. Hu, Q. Meng, and Y. Zhang, ''Short-time traffic flow prediction with ARIMA-GARCH model,'' in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 607–612.

[8] S. Periyanayagi and V. Sumathy, ''S-ARMA model for network traffic prediction in wireless sensor networks,'' *J. Theor. Appl. Inf. Technol.*, vol. 60, no. 3, pp. 1–7, 2014.

[9] I. Loumiotis, E. Adamopoulou, K. Demestichas, P. Kosmides, and M. Theologou, ''Artificial neural networks for traffic prediction in 4G networks,'' in *Proc. Int. Wireless Internet Conf.* Cham, Switzerland: Springer, 2014, pp. 141–146.

[10] A. Azzouni and G. Pujolle, ''NeuTM: A neural network-based framework for traffic matrix prediction in SDN,'' in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2018, pp. 1–5.

[11] C.-W. Huang, C.-T. Chiang, and Q. Li, ''A study of deep learning networks on mobile traffic forecasting,'' in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[12] A. Azari, P. Papapetrou, S. Denic, and G. Peters, ''User traffic prediction for proactive resource management: Learning-powered approaches,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[13] A. Rago, G. Piro, G. Boggia, and P. Dini, ''Multi-task learning at the mobile edge: An effective way to combine traffic classification and prediction,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10362–10374, Sep. 2020.

[14] 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.

[15] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, ''Interactive temporal recurrent convolution network for traffic prediction in data centers,'' *IEEE Access*, vol. 6, pp. 5276–5289, 2018.

[16] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, ''Structured sequence modeling with graph convolutional recurrent networks,'' in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 362–373.

[17] B. Yu, H. Yin, and Z. Zhu, ''Spatio–temporal graph convolutional networks: A deep learning framework for traffic forecasting,'' 2017, *arXiv:1709.04875*.

[18] D. Andreoletti, S. Troia, F. Musumeci, S. Giordano, G. Maier, and M. Tornatore, ''Network traffic prediction based on diffusion convolutional recurrent neural networks,'' in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2019, pp. 246–251.

[19] M. Kalander, M. Zhou, C. Zhang, H. Yi, and L. Pan, ''Spatio–temporal hybrid graph convolutional network for traffic forecasting in telecommunication networks,'' 2020, *arXiv:2009.09849*.

[20] S. Sun and X. Xu, ''Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 466–475, Jun. 2011.

[21] A. Bayati, V. Asghari, K. Nguyen, and M. Cheriet, ''Gaussian process regression based traffic modeling and prediction in high-speed networks,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.

[22] A. Bayati, K.-K. Nguyen, and M. Cheriet, ''Gaussian process regression ensemble model for network traffic prediction,'' *IEEE Access*, vol. 8, pp. 176540–176554, 2020.

[23] 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.

[24] C. E. Rasmussen, ''Gaussian processes in machine learning,'' in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.

[25] A. G. Wilson, ''Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes,'' Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 2014.

[26] D. J. C. MacKay, ''Introduction to Gaussian processes,'' *NATO ASI F, Comput. Sys. Sci.*, vol. 168, pp. 133–166, Jan. 1998.

[27] F. Lindgren, H. Rue, and J. Lindstrom, ''An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach,'' *J. Roy. Stat. Soc.*, vol. 73, no. 4, pp. 423–498, 2011.

[28] A. Venkitaraman, S. Chatterjee, and P. Handel, ''Gaussian processes over graphs,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5640–5644.

[29] K. B. Petersen and M. S. Pedersen, ''The matrix cookbook,'' *Tech. Univ. Denmark*, vol. 7, p. 510, Nov. 2008.

[30] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*. London, U.K.: University of London, University College London, 2003.

[31] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[32] *Networks With Multiple Demand Matrices*. Accessed: Dec. 30, 2021. [Online]. Available: http://sndlib.zib.de/home.action

[33] S. Seabold and J. Perktold, ''Statsmodels: Econometric and statistical modeling with Python,'' in *Proc. 9th Python Sci. Conf.*, 2010, pp. 1–5.

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

[35] C. Zheng, X. Fan, C. Wang, and J. Qi, ''GMAN: A graph multi-attention network for traffic prediction,'' in *Proc. AAAI*, 2020, vol. 34, no. 1, pp. 1234–1241.

[36] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

SAJAD MEHRIZI received the M.Sc. degree in electrical and computer engineering from the University of Khaje Nasir Toosi, Tehran, Iran, in 2015, and the Ph.D. degree in electrical engineering from the Interdisciplinary Center for Security and Trust, SnT, University of Luxembourg, Luxembourg City, Luxembourg, in 2021. His research interests include machine learning and Bayesian statistics for wireless communications, with focus on content caching.

SYMEON CHATZINOTAS (Senior Member, IEEE) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Surrey, U.K., in 2006 and 2009, respectively. He is currently a Full Professor/the Chief Scientist I and the Co-Head of the SIGCOM Research Group, SnT, University of Luxembourg. In the past, he was a Visiting Professor at the University of Parma, Italy. He was involved in numerous research and development projects at the National Center for Scientific Research Demokritos, the Center of Research and Technology Hellas, and the Center of Communication Systems Research, University of Surrey. He has (co)authored more than 400 technical papers in refereed international journals, conferences, and scientific books. He was a co-recipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award, the CROWNCOM 2015 Best Paper Award, and the 2018 EURASIC JWCN Best Paper Award. He is currently in the Editorial Board of the IEEE Open Journal of Vehicular Technology and the *International Journal of Satellite Communications and Networking*.