A Bayesian Learning Network for Traffic Speed Forecasting with Uncertainty Quantification

Ying Wu and James J.Q. Yu, *Senior Member, IEEE*
Department of Computer Science and Engineering,
Southern University of Science and Technology
Shenzhen 518055, China
12059004@mail.sustech.edu.cn, yujq3@sustech.edu.cn

***Abstract*—Intelligent transportation systems (ITS) depend on accurate and reliable traffic speed prediction to improve the safety, efficiency, and sustainability of transportation activities. Recently, deep learning approaches have significantly contributed to the development of ITS, but are still facing challenges in cyber-physical context due to the aleatoric uncertainty of increasingly uncertain traffic data and epistemic uncertainty of point-to-point estimation training models. In this work, a Bayesian deep learning model reframing with a universal traffic forecasting framework is devised for traffic speed forecasting with uncertainty quantification. The key idea of proposed network is to introduce time-series features in a latent distribution space. Compared to traditional point estimation neural networks, case studies show that the proposed model can predict more reliable results in cross domain learning tests and is capable of discovering good feature representations in missing traffic data or data-deficient scenarios.**

I. Introduction

ITS is one of the essential components in smart cities. Due to emerging technologies including advanced sensing, data transmission, big data computing and smart control technologies, ITS is expected to provide better services for road users, greater reliability for transportation systems, and "smarter" use of transport networks [1]–[3]. Traffic speed prediction is a vital branch of ITS. Many ITS applications (e.g., traffic congestion management [4], bus dispatching [5]) rely heavily on accurate and reliable traffic prediction speed systems.

Owing to the considerable effectiveness of the traffic speed forecast, many researchers have contributed to the development of algorithms to provide more precise traffic forecasts. These algorithms can be roughly divided into two categories, namely, model-driven and the data-driven approaches. They both have their own merits. Model-driven approaches (i.e., queuing network [6]) analyze the physical features and dynamic patterns of traffic systems based on prior knowledge, and are thus best suited to instantaneous traffic prediction tasks (i.e., traffic state estimation [6], [7]). Classic statistical models (i.e., Autoregressive Integrated Moving Average (ARIMA) models [8], K-Nearest Neighbors algorithm (k-NN) [9]) and machine learning models [10]–[12]) are two important representations of the data-driven approaches that are typically more resilient to noise and have high versatility applied in various traffic situations without taking into account complex traffic modeling. They map statistical regularity and relationships from complex high-dimensional data, making them competent for regression and classification tasks.

However, research gaps in the traffic speed prediction tasks still present. Most of the previous traffic prediction methods cannot take into account uncertainty quantification (UQ) of their forecast performance. UQ tries to measure the probability of certain outcomes of a situation in which something is unknown or uncertain. The UQ of output is crucial to decision-making. For instance, in bus dispatching applications such as prediction arriving time tasks, the UQ of the prediction determines how confidence the prediction time is, which can provide better services for passengers. Moreover, the UQ of model parameters is significant in training. The point estimation machine learning methods may be deployed as a rigid decision-making engine during the testing process. Typically, they tend to run models from scratch, assuming them had with minimal prior knowledge. This limits the model's potential performance and application. When the domain distribution changes, the model needs to be reconstructed, such can be very costly computationally and repetitive in real-world applications.

To bridge this research gap, this paper presents a universal traffic forecasting model with Bayesian inference. In comparison to the current point estimation work, the proposed model explicitly takes into account for the uncertainty of the model in both network parameters and the prediction output. Specially, Bayesian deep learning provides us with confidence of network parameters. Since the parameters follow the posterior distribution, accurate predictions can be made by averaging multiple inferences via the Monte-Carlo (MC) sampling. The major efforts of this work are summarized as follows:

- A Bayesian STGCN model is proposed by reframing a universal traffic forecasting framework Spatio-Temporal Graph Convolutional Networks (STGCN) with Bayesian inference.

This work was supported by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032 and by the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation No. 2020B121201001. James J.Q. Yu is the corresponding author.

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

- In contrast to existing point estimation approaches, the Bayesian STGCN model is used not only to forecast but also to measure uncertainty in spatial-temporal data for traffic speed prediction tasks.
- We test the proposed approach in two real world datasets. The properties of our model such as prediction accuracy, model uncertainty, and cross-domain/dataset learning are investigated. The results indicate that the proposed algorithm is capable of transfer knowledge to the issue of traffic speed forecasting for a number of related tasks.

The rest of this paper is organized as follows. Mathematical representation of investigated traffic speed prediction tasks and Bayesian deep learning are introduced in Section II. Section III presents how to convert the Spatio-Temporal Graph Convolutional Networks to Bayesian traffic forecasting neural networks. Section IV presents real-world benchmark data and experimental settings. Experimental results and analysis are also shown in Section IV. Section V concludes the paper and presents future work.

II. Related work

Owing to its complex interdependence (i.e., temporal relationships in a single traffic series), inconsistent interactivity (i.e., spatial correlations in traffic networks with multiplexed interactions), and extra-disturbing elements (i.e., weather and traffic accidents), traffic forecasting is an incredibly difficult job.

Traditional classic statistical approaches, such as Historical Average (HA) and ARIMA [13]), rely solely on the periodicity of time series to map statistical regularity and relationships from complicated high-dimensional data, which cannot provide an accurate predictor for long-term traffic forecasting on a wide scale traffic data. Recurrent neural network (RNN) and its variants (e.g., Long Short-term Memory (LSTM) [14] and Gated Recurrent Unit (GRU)) are capable of capturing nonlinear traffic dynamics and overcome the issue of backwards propagated error postponement in long temporal dependencies. Despite their superior capability for time series prediction with long temporal dependency, they overlook the significance of spatial dependency for accurate prediction. To better understand the spatial connections in traffic networks, researchers shift to think Graph Neural Networks (GNN) [15], [16] to capture spatial information of road links. Models derived from GNN and RNN (e.g., T-gcn [17], GMAN [18], GCNN [19]) are capable of learning complex topological architectures of traffic networks while understanding dynamic temporal changes.

However, despite a wide range of the existing efficient traffic state prediction methods, there still remain some primary challenges. First and foremost, massive random inaccurate and fuzzy uncertainty exist in the raw traffic data recorded by GPS and sensors. Such uncertainty is the variability in the outcome of an experiment, called aleatoric uncertainty [20]. The typical problem is that the random systematic residual errors exist in the speed data from speedometers on the highway even after correction. Secondly, interpolation uncertainty may arise from a lack of available data for model simulations or experimental measurements. This requires predictors to identify the latent features of the small datasets where the training data is sparse.

III. PRELIMINARIES

A. Problem Formulation

In the traffic network, the traffic state forecast pattern refers to the process of the inference of traffic variables using partially observed traffic big data. Specifically, traffic data is the sampling of dynamics of moving objects using vehicle-mounted GPS, WiFi, Bluetooth and RFID [21], [22] in both temporal and spatial dimensions. The main key of the traffic speed prediction problem is the study and mining of time series data to discover and analyze the underlying knowledge and patterns on transportation activities, vehicular traffic relations and even the city dynamics.

Formally, traffic speed forecasting is a time series prediction task. The input domain of feature space can be defined as $\{x_{t-m+1}, ..., x_t\}$, which is an observation vector at time step t. $\{x_{t+1}, ..., x_{t+n}\}$ is the vector of traffic speed vector in the following *n* time steps given the past *m* steps and $f(\cdot)$ is the mapping function. The process can be defined as:

$$\{x_{t-m+1}, ..., x_t\} \overset{f(\cdot)}{\rightarrow} \{x_{t+1}, ..., x_{t+n}\} \tag{1}$$

B. Bayesian deep learning

To overcome these challenges, the predictors need to catch the latent travel dynamics patterns within an area instead of fixed point-to-point methods. The key problem, however, is that it is difficult for machine learning predictors to model uncertainties. This section introduces a deep learning framework which can produce a state-of-the-art result while recognizing uncertainty. This Bayesian deep learning framework can model complex tasks by leveraging the hierarchical representation of deep learning, while forming uncertainty by placing region over parameters of model, or by learning a mapping to probabilistic outputs.

Formally, given training inputs $\{x_{t-m+1}, ..., x_t\}$ and their corresponding outputs $\{x_{t+1}, ..., x_{t+n}\}$, objective predictive function $f(\cdot)$ is likely to produce outputs that can be rewritten as $P(y \mid x)$ with well-learned parameters $\mathbf{w}$ from a probabilistic viewpoint. Each parameter can be trained to minimize the maximum likelihood estimation (MLE):

$$\begin{aligned} \mathbf{w}^{\text{MLE}} &= \arg\min_{\theta} \log P(\mathcal{D} \mid \boldsymbol{w}) \\ &= \arg\min_{\theta} \log P(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{w}) \end{aligned} \tag{2}$$

Instead, the Bayesian neural network assumes that both of these parameters are accompanied by posterior probability distributions over the $P(\boldsymbol{w} \mid \mathcal{D})$ domain of the dataset. With this, we can predict the output of the new input point x^* by ensembling all possible function:

$$\begin{aligned} P(y^* \mid x^*) &= \mathbb{E}_{P(\boldsymbol{w}|\mathcal{D})}[P(y^* \mid x^*, \boldsymbol{W})] \\ &= \int P(y^* \mid x^*, \boldsymbol{W}) P(\boldsymbol{w} \mid \mathcal{D}) d\boldsymbol{w} \end{aligned} \tag{3}$$

Nevertheless, there remains an issue in Eq. 3: how to present a prior distribution of the network parameters, considering the sophistication of the network parameters' posterior distribution. In the Bayesian network, a prior distribution implies a distribution of probabilities that may give rise to a belief before taking any evidence into account. Although the physical definition of the prior belief is ambiguous, a variety of methods can be used to establish prior distribution in previous studies. It can be determined from subjective assessment of empirical analysis, as well as being chosen according to domain principles. In this work, we utilize the standard parametric distribution which assumes the prior probability distributions follow zero-mean Gaussian distributions due to the its merit of regularizations [23].

With this hypothesis, the main challenge is on how to determine the postprior distribution. Previous work in [24]–[26] suggested that it is possible to use a variational approximation to prune the latent parameters θ of a region on the weights $q(\boldsymbol{w} \mid \theta)$. The parameter can be found by minimizing the divergence of Kullback-Leibler (KL) with truth Bayesian posterior on the weights:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\min_{\theta} \mathbf{KL}\left[q(\boldsymbol{w} \mid \boldsymbol{\theta}) \parallel P(\boldsymbol{w} \mid \mathcal{D})\right] \\ &= \arg\min_{\theta} \int q(\boldsymbol{w} \mid \boldsymbol{\theta}) \log \frac{q(\boldsymbol{w} \mid \boldsymbol{\theta})}{P(\boldsymbol{w}) P(\boldsymbol{w} \mid \mathcal{D})} \end{aligned} \tag{4}$$

The resulting cost function is variously known as the negative variational free energy [26]–[28] or expected evidence lower bound (ELBO) [26], [27]. The cost function allowing for MC sampling and prior/posterior combination is formulated as

$$\begin{aligned} \mathcal{F}(\mathcal{D}, \theta) &= \mathbf{KL}\left[q(\boldsymbol{w} \mid \boldsymbol{\theta}) \parallel P(\boldsymbol{w})\right] - \mathbb{E}_{q(\boldsymbol{w}|\boldsymbol{\theta})}\left[\log P(\boldsymbol{w} \mid \mathcal{D})\right] \\ &\approx \sum_{i=1}^{n} q(\boldsymbol{w}^{(i)} \mid \boldsymbol{\theta}) - \log P(\boldsymbol{w}^{(i)}) - \log P(\mathcal{D} \mid \boldsymbol{w}^{(i)}) \end{aligned} \tag{5}$$

where $\boldsymbol{w}^{(i)}$ is the i-th MC sample from $q(\boldsymbol{w}^{(i)} \mid \boldsymbol{\theta})$.

From Eq. 5, it is clear that the cost function is composed of two terms, namely a data-dependent part that refers to the likelihood cost and a prior dependent part that refers to the complexity [26].

IV. Bayesian Spatio Temporal Graph Convolutional Model

In this section, we introduce STGCN, a universal time series system that processes [29]. Afterward, how to ensemble STGCN with the Bayesian deep learning method is discussed.

A. STGCN

As a graph-based neural network, STGCN sees the traffic network nodes as the sensors mounted on the road, and the edges being determined by the distance between pairs of nodes. Each node considers the average speed of traffic within a window as its input features [16], [29]. STGCN builds two spatial-temporal convolution blocks and a fully-connected output layer, as illustrated in Figure 1. It collects spatial information on the basis of 1-D convolutional layers with ChebNet [30] and captures the temporal features of the 1-D convolution layers [31]. Residual connection, bottleneck strategy, and layer normalization are implemented within each block to prevent overfitting and achieve fast spatial-state propagation [29]. As a consequence, the spatial-temporal convolution block can be described as:

$$v^{l+1} = \Gamma_1^l * \tau \mathrm{ReLU}\left(\Theta^l * \mathcal{G}\left(\Gamma_0^l * \tau v^l\right)\right) \tag{6}$$

where Γ_0^l and Γ_1^l are the upper and lower temporal convolution kernels of layer l, respectively. $\Gamma_1^l * \tau Y = P \odot \sigma(Q)$ is the gated linear units (GLU) convolution shown in Figure 2(b), where P and Q are the splits with the same size of previous output. Θ^l is the spatio convolution kernel of layer l.

The benefits of STGCN can be summarized as follows. First, the architecture incorporates spatial and temporal reliance on traffic prediction tasks and achieves substantial improvement in short- and mid-to-long term forecasting issues compared to current baselines. It expands spatio-temporal sequence learning methods due to its spatio-temporal convolutional "sandwich" structure. More general spatial-temporal sequence learning tasks can be transferred to this architecture. Third, higher training speed and easier convergence is achieved due to convolution rather than RNN-based approaches as in [32]. Finally, it trains with less parameters. Special designs in ST-Conv blocks ensure that STGCN parameters are more time-efficient compared to FC-LSTM [33] and Graph Convolutional GRU (GCGRU) [34].

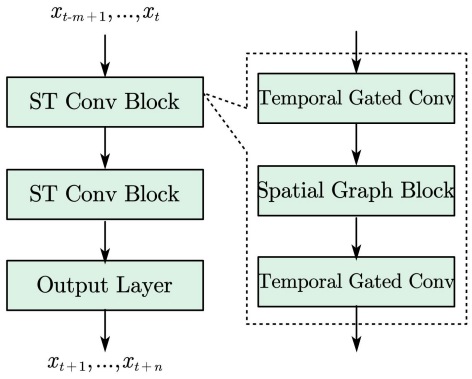
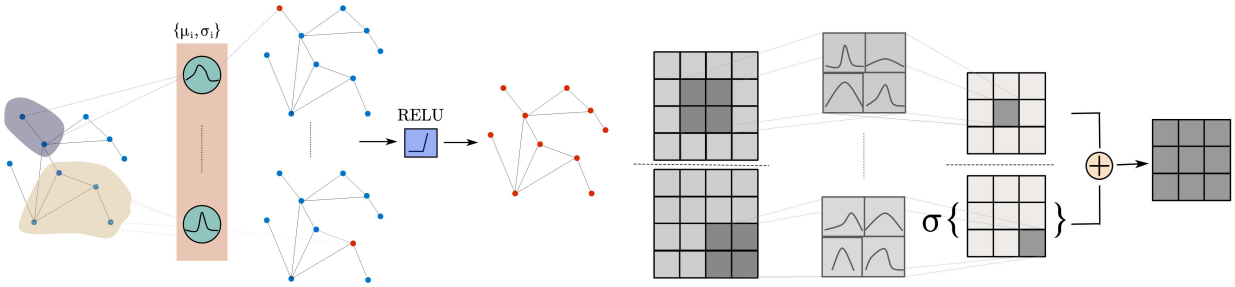


Fig. 1. Hierarchical Architecture of STGCN.

B. Bayesian STGCN with Bayes by Backprop

Although STGCN achieves superior performance in traffic prediction tasks, the method learns directly from data without imposing specific uncertainty on the inference process. STGCN neglects the diversity and ambiguity of network parameters and provides deterministic forecast performance in time series issues. Here, the Bayesian STGCN is proposed in this work to model the uncertainty in the traffic forecasting problem.

Instead of using a single set of fixed parameters, Bayesian STGCN assumes that all parameters follow the posterior probability distribution. However, the expectation of the posterior



(a) Graph CNNs in probability distributions over weights

(b) Gated CNNs in probability distributions over weights

Fig. 2. Bayesian perspective of Graph CNNs and Gated CNNs. Each convolutional weight has a probability distribution

distribution of weights shall be intractable for any practical use as introduced in Section II. Therefore, a variational parameter parameterized by $\theta = \{\mu, \sigma\}$ is adopted to approximate the true probability distribution [26].

Each of the weights in graph convolutions layer and gated temporal convolutions layer follows a probability distributions, as illustrated in Figure 2. A sample of weights can be obtained from the Gaussian distribution parameterized by $\{\mu, \sigma\}$. After that, the forward neural network can be estimated as normal.

Then the predictors turns into how to implement backpropagation in Bayes. As defined in Section II-B, local $q(\boldsymbol{w} \mid \boldsymbol{\theta})$ is used by the deterministic function $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$, where $\boldsymbol{w} = t(\boldsymbol{\theta}, \epsilon) = \boldsymbol{\mu} + \log(1 + \exp(\boldsymbol{\rho})) \circ \epsilon$, and $\circ$ is point-wise multiplication in Bayes by the Backprop process [25], [26], [35], [36]. The term of the gradients $\frac{\partial f(\boldsymbol{w}, \boldsymbol{\theta})}{\partial \boldsymbol{w}}$ (refer to Eqs. 7 and 8) for the mean and standard variation are shared which can be done through normal backward propagation. To prune the complexity of the model, the bias of each layer still follows the single-fixed value:

$$\Delta_{\boldsymbol{\mu}} = \frac{\partial f(\boldsymbol{w}, \boldsymbol{\theta})}{\partial \boldsymbol{w}} + \frac{\partial f(\boldsymbol{w}, \boldsymbol{\theta})}{\partial \boldsymbol{\mu}} \tag{7}$$

$$\Delta_{\boldsymbol{\rho}} = \frac{\partial f(\boldsymbol{w}, \boldsymbol{\theta})}{\partial \boldsymbol{w}} \frac{\epsilon}{1 + \exp(-\boldsymbol{\rho})} + \frac{\partial f(\boldsymbol{w}, \boldsymbol{\theta})}{\partial \boldsymbol{\rho}} \tag{8}$$

The cost function is set to minimizing the ELBO cost as described in Eq. 5. The model is trained with the RMSProp optimizer (initial learning rate is 10^{-3} with a decay rate of 0.7 for every 5 epoch). The prior of the network weights [26], [37] is taken as a mixture scale of two Gaussian densities with zero-mean, variance σ_1 and σ_2, respectively:

$$p(\theta) = \prod_j \pi \mathcal{N}(\theta_i \mid (0, \sigma_1^2) + (1 - \pi)\mathcal{N}(\theta_j \mid (0, \sigma_2^2)) \tag{9}$$

where π is the parameter of mixture distribution.

In Eq. 3, the new output x^* is defined as integrating over all possible functions. We can approximate it by MC integration:

$$\begin{aligned} P(y^* \mid x^*, D) &\approx \int P(y^* \mid x^*, \boldsymbol{W}) q(\boldsymbol{w}) d\boldsymbol{w} \\ &\approx \frac{1}{T} \sum_{t=1}^{T} (y^* \mid x^*, \hat{\boldsymbol{W}}_t) \end{aligned} \tag{10}$$

with $\hat{\boldsymbol{W}}_t \sim q(\boldsymbol{w} \mid \boldsymbol{\theta})$, where $q(\boldsymbol{w} \mid \boldsymbol{\theta})$ is the posterior distribution. T denotes the sampling coefficient. The parameter training of Bayesian STGCN can be outlined as Algorithm 1.

Algorithm 1: Parameter Training of Bayesian STGCN

Input: $X = \{x^i\}_{i=1}^N$
Output: $\boldsymbol{\theta}^*$

Randomly initialize $\boldsymbol{\theta}$
while $\boldsymbol{\theta}$ *not not converged* **do**
  Sample $\epsilon_i \sim \mathcal{N}(0, I)$, $\boldsymbol{w} = t(\boldsymbol{\theta}, \epsilon)$.
  Calculate derivative of $\boldsymbol{\theta}$ as loss function L defined in Eq. 5
  Update the posterior variable parameters as $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} - \alpha \Delta_{\boldsymbol{\theta}}$

V. Experiments

In this work, a Bayesian learning network called Bayesian STGCN, is proposed to develop traffic speed forecasts with uncertainty quantification. Several comprehensive case studies with two real-world traffic speed data have been developed to fully evaluate the performance of the proposed network. Subsequently, to evaluate the generation and transfer learning ability of proposed network, we employ Bayesian STCGN and point estimating STGCN with training and testing in different but related domain datasets. Finally, we investigate the sensitivity of MC sampling in the proposed network.

A. Configurations

PeMSD7 collects traffic data from more than 39,000 senor stations on the highways of Los Angeles County by the Caltrans Output Assessment System (PeMS) [38]. The dataset is aggregated with 5-minute interval. We select PeMSD7(M) with the time covering the weekends of May and June of 2012. It contains 228 stations in 34 days dataset of PeMSD7. The first month of historical speed records is chosen as a training list, and the remainder acts as validation and test sets.

SZ-taxi is deployed and maintained by Shenzhen Transport Committee. The dataset identified the GPS data of the taxicab in Shenzhen. 156 major roads of Luohu District is chosen as the study area.

The data input is standardized by the Z-Score method. The data preprocessing follows [29]. Three metrics are used to assess and evaluate the efficiency of the proposed models, i.e., mean absolute error (MAE), mean absolute percent error (MAPE), and root mean square error (RMSE). All experiments are compiled and tested on a Linux cluster (CPU: Intel(R) Xeon(R) E5-2620 v4, GPU: NVIDIA GeForce RTX 2080 Ti). All the tests adopts approximately 60 minutes historical data (12 data points) and forecast traffic statues in the next 15, 30 and 45 minutes (3, 6, 9 prediction data points).

B. Bayesian STGCN vs Point Estimation STGCN

We first present results on the performance comparison of Bayesian STGCN and point estimation STGCN with three evaluation metrics as well as T=2 and T=50 forward passing through the Bayesian STGCN network as shown in Table I. As expected, point estimation STGCN develops better accuracy in prediction than Bayesian STGCN in Table I (a) with best accuracy among three evaluation metrics. It is construable that the output of Bayesian STGCN forecast is an average of the ensemble of neural networks weighted by the qualified mapping feature distribution set. The point estimation STGCN is not required to consider "averaging" but focuses on "best guessing". Another consistent observation is that, in Bayesian STGCN at T=2, the output uncertainty is quantified by the standard deviation error. For instance, in terms of MAPE, the percentage of uncertainty quantification gradually increases from 0.266% to 0.663% as the window speed forecast passes from 15min to 45min forecast, which follows a common intuition. The third inference that can be drawn from Table I (a) is that the effects of the predictions are more striking at T=50 than at T=2 in the assessment metrics. For instance, in 15 forecasting window, the average value of MAPE substantially declines from 6.203% to 5.696% with a 8.18% relative slides and the quantified uncertainty significantly decreases from 0.266% to 0.019% with 91.6% drops.

Next, we evaluate the cross domain learning performance with both networks. We train the Bayesian STGCN and point estimation STGCN in PeMS dataset but test in SZ dataset respectively. The use of such an estimation technique is often linked to worse efficiency while the algorithm produces over-optimistic outcomes. However, this approach will decide the true capacity of STGCN and Bayesian STGCN to assess the traffic status of the large traffic dataset. The results from Table I (b) indicates that Bayesian STGCN can provide more generality across domains. The outcome of Bayesian STGCN in domain learning efficiency is far higher than at of STGCN with a statistical significance level of 99% (two-sided T-test). Another observation is that although the missing rate in SZ data is 16%, the Bayesian STGCN still still has a striking performance. This shows that the Bayesian STGCN is capable of finding strong representation in traffic data missing issue or data-deficient situations.

Figure 3 shows a histogram with the kernel destiny estimation curve of SZ test daset normalized by the Z-Score method. From this illustration, we can draw the following conclusions.

TABLE I
TESTING GENERALIZATION OF NETWORK, PERFORMANCE COMPARISON OF TWO APPROACHES ON THE DATASET PEMSD7 AND DATASET SZ

(a) Train on PeMS, Test on PeMS (15min/30min/45min)

Model	MAPE (%)	MAE	RMSE
STGCN	**5.242**	**2.228**	**4.053**
	7.493	**3.019**	**5.742**
	9.264	**3.600**	**6.878**
Bayesian STGCN (MC sampling = 2)	6.203±0.266	2.652±0.142	4.605±0.155
	8.687±0.289	3.666±0.209	6.669±0.322
	11.663±0.663	4.571±0.260	8.208±0.315
Bayesian STGCN (MC sampling = 50)	5.696±0.019	2.449±0.018	4.433±0.032
	8.137±0.032	3.448±0.026	6.532±0.048
	10.077±0.042	4.257±0.027	8.118±0.052

(b) Train on PeMS, Test on SZ (15min/30min/45min)

Model	MAPE(%)	MAE	RMSE
STGCN	19.132	10.774	15.316
	28.68	13.499	18.657
	28.641	17.568	23.529
Bayesian STGCN (MC sampling = 2)	20.649 ±0.818	10.010±1.018	13.517±1.722
	21.049±0.759	12.210±1.954	16.110±2.821
	21.388±0.804	13.114±2.317	18.085±3.185
Bayesian STGCN (MC sampling = 50)	**18.9464 ±0.118**	**9.607±0.123**	**12.896±0.228**
	19.261±0.165	**11.572±0.153**	**15.020±0.239**
	20.507±0.136	**12.550±0.152**	**16.146±0.230**

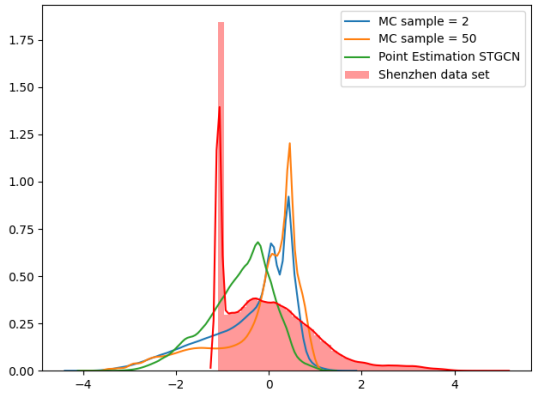
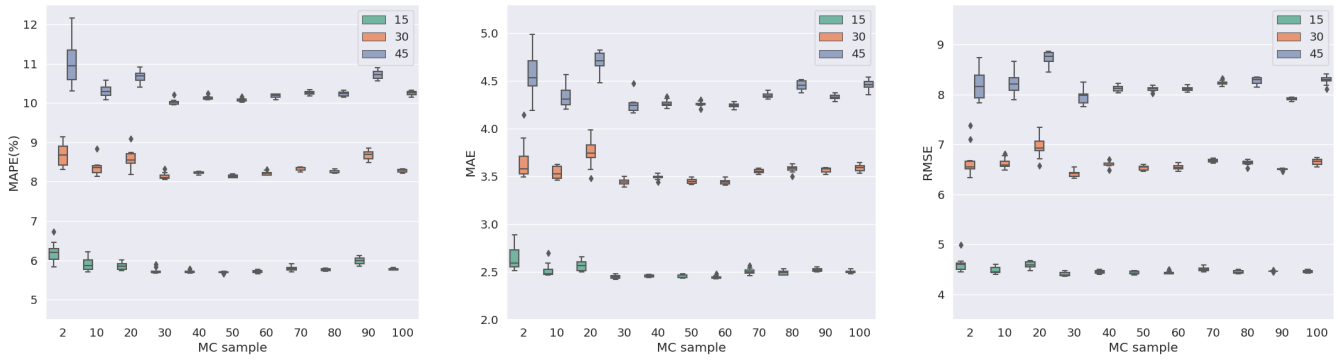


Fig. 3. The histogram of prediction output with kernel destiny curve regarding to the increasing of MC sampling.

In SZ data histogram (blue), the SZ data protrudes significantly in the -1 bin due to the fact that the missing rate of the raw real-world dataset. Subsequently, in point estimation STGCN histogram (green), the prediction output $P(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$ follows the Gaussian distribution relating to the square error residual sum. Root stands for square error (RMSE) corresponding to



(a) MAPE (b) MAE (c) RMSE

Fig. 4. The histogram of prediction output evaluation (MAPE, MAE, RMSE)with kernel destiny curve regarding to the increasing of MC sampling.

the standard deviation σ in the output distribution.

C. Sensitivity of Hyperparameters MC sampling

The sensitivity of the MC sampling number in Bayesian STGCN is evaluated as shown in Figure 4. We note that the MAPE, MAE and RMSE of Bayesian STGCN improve smoothly with the increase in MC sampling and variance convergence with more than 30 samples. This means that an appropriate increase in MC sampling can improve the performance of Bayesian STGCN. However, the improvement in accuracy is at the cost of complexity, with training in computational time of 9.381s at T=2 significantly increasing to 250.76s at T=100. Therefore, a trade-off between performance and computational cost needs to be considered for large datasets.

VI. Conclusion

In this work, we propose a Bayesian STGCN to address traffic speed forecasting problem. In comparison to current point estimation methods, the proposed method integrates uncertainty quantification in decision-making. The proposed method assumes that all parameters follow the posterior distribution which is approximated by Bayes by Backprop variational inference. In order to build strategies for computing credible predictions, MC sampling is designed to average over these probabilistic predictions. Case studies on two real datasets demonstrate that the proposed model can develop more reliable results in cross-domain learning tests.

UQ in deep forecasting models is a critical issue for optimum decision-making, but spatio-temporal traffic data makes the problem harder due to its unclear uncertainty estimation, efficient optimization in the regression latent space and heterogeneous traffic data sources. In the future work, we expect to investigate extensions and application of Bayesian deep learning neural networks in ITS and explore alternative approaches for uncertainty in deep learning (e.g., dropout as a Bayesian approximation [39], [40], embedding network [41]).

References

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010.

[2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[3] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[4] A. de Palma and R. Lindsey, "Traffic congestion pricing methodologies and technologies," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1377–1399, 2011.

[5] K. Gkiotsalitis and E. Van Berkum, "An exact method for the bus dispatching problem in rolling horizons," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 143–165, 2020.

[6] X.-y. Xu, J. Liu, H.-y. Li, and J.-Q. Hu, "Analysis of subway station capacity with the use of queueing theory," *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 28–43, 2014.

[7] C. P. Van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. Van Lint, and H. J. Van Zuylen, "Localized extended kalman filter for scalable real-time traffic state estimation," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 1, pp. 385–394, 2011.

[8] N. L. Nihan, "Aid to determining freeway metering rates and detecting loop errors," *Journal of Transportation Engineering*, vol. 123, no. 6, pp. 454–458, 1997.

[9] Z. Liu, S. Sharma, and S. Datla, "Imputation of missing traffic data during holiday periods," *Transportation Planning and Technology*, vol. 31, no. 5, pp. 525–544, 2008.

[10] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, vol. 50, pp. 148–163, 2018.

[11] D. Zang, J. Ling, Z. Wei, K. Tang, and J. Cheng, "Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3700–3709, 2018.

[12] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, 2019.

[13] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[14] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[15] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[16] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, p. 4–24, Jan 2021. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2020.2978386

[17] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[18] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[19] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 890–897.

[20] M. H. Shaker and E. Hüllermeier, "Aleatoric and epistemic uncertainty with random forests," in *International Symposium on Intelligent Data Analysis*. Springer, 2020, pp. 444–456.

[21] Y. Chen, M. Guizani, Y. Zhang, L. Wang, N. Crespi, and G. M. Lee, "When traffic flow prediction meets wireless big data analytics," *arXiv preprint arXiv:1709.08024*, 2017.

[22] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual reviews in control*, vol. 43, pp. 128–151, 2017.

[23] J. J. Q. Yu, "Sybil attack identification for crowdsourced navigation: A self-supervised deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[24] G. E. Hinton and D. Van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the sixth annual conference on Computational learning theory*, 1993, pp. 5–13.

[25] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011, pp. 2348–2356.

[26] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[28] K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny, "Variational free energy and the laplace approximation," *Neuroimage*, vol. 34, no. 1, pp. 220–234, 2007.

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

[30] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[31] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 1243–1252.

[32] 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 *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 802–810.

[33] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014.

[34] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[35] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to bayesian convolutional neural network with variational inference," *arXiv preprint arXiv:1901.02731*, 2019.

[36] T. Liu, Y. Liu, J. Liu, L. Wang, L. Xu, G. Qiu, and H. Gao, "A bayesian learning based scheme for online dynamic security assessment and preventive control," *IEEE Transactions on Power Systems*, 2020, in press.

[37] M. Fortunato, C. Blundell, and O. Vinyals, "Bayesian recurrent neural networks," *arXiv preprint arXiv:1704.02798*, 2017.

[38] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.

[39] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," 2016.

[40] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, vol. 1, p. 3, 2016.

[41] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2017.