MISSING DATA IN TRAFFIC ESTIMATION: A VARIATIONAL AUTOENCODER IMPUTATION METHOD

Guillem Boquet *Jose Lopez Vicario* *Antoni Morell* *Javier Serrano*

Wireless Information Networking (WIN) Group
Universitat Autònoma de Barcelona (UAB)

ABSTRACT

Road traffic forecasting systems are in scenarios where sensor or system failure occur. In those scenarios, it is known that missing values negatively affect estimation accuracy although it is being often underestimate in current deep neural network approaches. Our assumption is that traffic data can be generated from a latent space. Thus, we propose an online unsupervised data imputation method based on learning the data distribution using a variational autoencoder (VAE). This is used as an independent pre-processing step prior to traffic forecasting which is then evaluated against missing data of a real-world dataset. Compared to other methods, we show that VAE improves post-imputation traffic forecasting performance while allowing for data augmentation, data compression and traffic classification at the same time.

Index Terms— traffic forecasting, deep learning, missing data, imputation method, intelligent transportation systems

1. INTRODUCTION

Traffic forecasting is an estimation problem that has been an integral part of most Intelligent Transportation Systems (ITS) and related research [1]. Currently, deep neural networks (DNN) approaches have succeeded in this field mainly because of the ability to efficiently model the nonlinearities of traffic behavior [2] and because of the amount of traffic data available due to ITS growth. Authors of [1] summarized some DNN approaches prior to 2014 but more recent advanced architectures include, for example, [3] where traffic data are treated as images while using CNN to exploit spatiotemporal relationships or [4] and [5] where LSTM networks are used to model the long temporal dependency of traffic. Nevertheless, none of these approaches mentions how missing data are handled or how it affects system performance, although it is known that the performance of DNN models depends directly on the quality of the data [6]. Many real-world traffic datasets used in literature contain missing values (MVs) for many reasons such as system or sensor failure. A basic strategy is to discard entire rows containing MV but this comes at the price of losing data which may be valuable. A better strategy is to pre-process data imputing MVs, i.e., to infer them from the known part of the data. State-of-the-art imputation methods can be categorized [7] as either *discriminative*, such as multiple imputation by chained equations (MICE) [8] and matrix completion [9], or *generative* methods based on DNN. Two well-known imputation methods in traffic forecasting are k-nearest neighbors (KNN) [10] and principal component analysis (PCA) [11]. Related to our work but not to traffic forecasting, [12] and [13] proposed a generative model imputation method using generative adversarial networks (GAN). Also, [14] proposed an overcomplete denoising autoencoder (DAE) to be able to reconstruct data by stochastically corrupting it.

In traffic forecasting, missing data are part of the inherent structure of the problem: all current real-world datasets contain MVs. To circumvent this, in this work we propose to learn the underlying structure that generates traffic data. We assume that traffic is not generated randomly but from a latent subspace. Thus, we formulate it as a generative model which forces us to approximate the joint probability distribution via Bayesian inference. To that aim we propose the use of variational autoencoder (VAE) [15, 16] that allows us to impute missing traffic data in an online unsupervised fashion from the learned data distribution. Also, we constrain the latent space dimensionality resulting in only learning useful properties for traffic forecasting [17, 18] while allowing for data compression. Here, we force the posterior distribution to be continuous which then VAE is able to learn a continuous latent space. This means that traffic of the same class ends up closer together in said space which allows for unsupervised traffic classification while at the same time to detect anomalous traffic. Moreover, we can generate new traffic data as our proposal has learned the traffic data distribution.

Next, in Sec. 2 we introduce the traffic forecasting and missing data problem. In Sec. 3 we formulate it as a latent variable model while providing an implementation that is able to learn the data distribution and impute MVs. Finally, in Sec. 4 we experiment with a real-world traffic dataset. We evaluate our proposal w.r.t the post-imputation performance of a traffic speed forecasting system providing results against different missing rates and compression factors.

This research is supported by the Catalan Government under Project 2017 SGR 1670 and the Spanish Government under Project TEC2017-84321-C4-4-R co-funded with European Union ERDF funds.

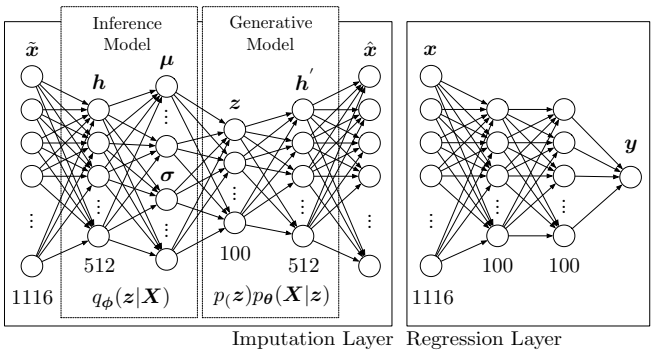

Fig. 1. Implemented road traffic forecasting system.

2. THE TRAFFIC FORECASTING PROBLEM

Let $\boldsymbol{x} \in \mathbb{R}^{n \times d}$ be a data sample from the traffic dataset $\boldsymbol{X}$, where n is the number of past data samples and d the number of detectors. The elements within $\boldsymbol{x}$ are values of traffic variables associated with time and space that may be viewed as an image to account for the spatial and temporal correlation information between traffic network points [3]. Likewise, let $\boldsymbol{y} \in \mathbb{R}^m$ be the future state of $m \leq d$ subset of detectors in the time horizon of h samples.

The goal of a traffic forecasting system modeled given $\boldsymbol{y} = f^*(\boldsymbol{x})$ is to accurate estimate $\boldsymbol{y}$ and thus the challenge is to derive a function f that closely resembles f^*. However, another important challenge is how to handle missing data to not deteriorate the performance of the system. Thus, hereafter we focus on DNN traffic forecasting affected by missing data. Fig. 1 shows the scenario under consideration divided into two parts: an imputation layer (IL) that pre-processes missing traffic data which is then feed to the regression layer (RL) to estimate a future traffic variable. In this context, the IL aims to impute MVs as close to reality as possible. This may be viewed as a reconstruction problem where applying a function to a corrupted input $\tilde{\boldsymbol{x}}$ leads to the actual input. Like in the forecasting problem, this function can be approximated using a deep learning approach to get an estimation of the actual input $\hat{\boldsymbol{x}}$ which we derive in the following section.

3. LEARNING TO GENERATE TRAFFIC DATA

Suppose we have a traffic sample (or image) $\boldsymbol{x}$ that is partially occluded due to sensor or system failure. Missing data could be anything if an underlying structure generating the data does not exist. From traffic theory, we know that spatiotemporal relationships and seasonality exists and therefore one could guess the day type looking only at how morning traffic is developing through time and space. Therefore, if we learn how traffic data are generated, i.e., data distribution, we would be able to reconstruct it when part of the input is missing or even generate new traffic data.

3.1. Generative model

Let $\boldsymbol{z}$ be a continuous random latent variable which represents the structure behind the data, $f^{'}$ denote a function that maps $\boldsymbol{z}$ to data space and (1) a generative model parametrized with $\boldsymbol{\theta}$ where $\hat{\boldsymbol{X}}$ is the estimation of $\boldsymbol{X}$.

$$\boldsymbol{X} \approx \hat{\boldsymbol{X}} = f^{'}(\boldsymbol{z}, \boldsymbol{\theta}) \tag{1}$$

Our motivation is to learn $f^{'}$ that minimizes the error between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ which is equivalent to maximize the probability distribution of the data $p_{\boldsymbol{\theta}}(\boldsymbol{X})$ in terms of $\boldsymbol{\theta}$, a maximum likelihood problem. As we are assuming that $\boldsymbol{X}$ was generated by a random process involving $\boldsymbol{z}$, we could integrate over the joint probability as

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{z}) d\boldsymbol{z} = \int p_{\boldsymbol{\theta}}(\boldsymbol{z})\, p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z}) d\boldsymbol{z}\,. \tag{2}$$

Unfortunately, (2) is intractable under this scenario [15, 16]. To circumvent this, authors of [15] and [16] proposed an efficient algorithm for DNN. Instead of computing the intractable marginal likelihood, they proposed to train as an optimization problem the generative model (1) jointly with a recognition (or inference) model using variational inference to approximate the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z})$. Thus, the data model may be viewed as consisting of two parts. The generative model $p_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z})$ which given $\boldsymbol{z}$ it produces a distribution over the possible corresponding values of $\boldsymbol{x}$. The inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{X})$ parameterized with $\boldsymbol{\phi}$, an approximation to the intractable true posterior, which given a data point $\boldsymbol{x}$ it produces a distribution over the possible values of $\boldsymbol{z}$ from which the data point $\boldsymbol{x}$ could have been generated. From there, a variational lower bound on the marginal likelihood can be derived which can be optimized in terms of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ at the same time [15, 16]. This yielded the known objective function (3) where the first term is the expected reconstruction error and the second term is the Kullback-Leibler (KL) divergence of the approximate posterior from the prior.

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; x) = {} & \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] \\ & - D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z})) \end{aligned} \tag{3}$$

3.2. Variational autoencoder

The aforementioned model may be implemented using an autoencoder architecture: the inference model as an encoder and the generative model as a decoder (see Fig. 1). First, we let the prior over the latent variables be an isotropic multivariate Gaussian $p(\boldsymbol{z}) = \mathcal{N}(0, \boldsymbol{I})$. Note that this allows for a continuous latent (or code) space. Then, we let the inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ be a multivariate Gaussian with a diagonal covariance structure. This is modeled as the encoder using a 1-layer multilayer perceptron (MLP) with weights and biases $\boldsymbol{\phi} = \{W_1, W_2, W_3, b_1, b_2, b_3\}$ whose outputs are the mean $\boldsymbol{\mu}$

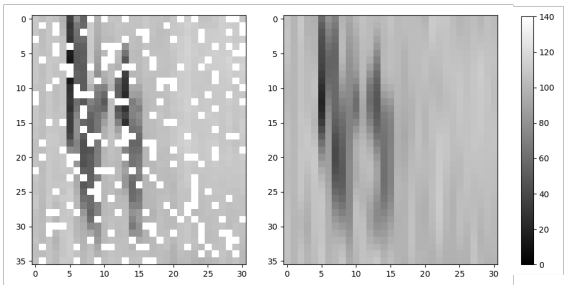

Fig. 2. Reconstruction of a corrupted sample using our VAE implementation. y-axis shows 6 hours of speed data $[km/h]$. x-axis represents each detector.

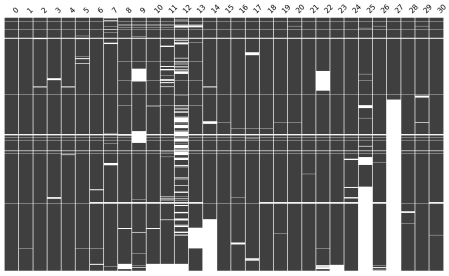

Fig. 3. Distribution over the dataset of the induced missing values (white fields). Each column shows two years of 5-minute samples corresponding to each detector.

and s.d. $\boldsymbol{\sigma}$:

$$\begin{aligned} \boldsymbol{h} &= ReLU(\boldsymbol{W}_1\,\boldsymbol{x} + \boldsymbol{b}_1) \\ \boldsymbol{\mu} &= \boldsymbol{W}_2\boldsymbol{h} + \boldsymbol{b}_2 \\ \boldsymbol{\sigma} &= \boldsymbol{W}_3\boldsymbol{h} + \boldsymbol{b}_3\,. \end{aligned} \tag{4}$$

Likewise, we let $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ be a multivariate Gaussian whose distribution parameters are computed from $\boldsymbol{z}$ with a 1-layer MLP with weights and biases $\boldsymbol{\theta} = \{W_4, W_5, b_4, b_5\}$ (decoder of Fig. 1). The decoder output is defined as

$$\hat{\boldsymbol{x}} = \boldsymbol{W}_5\,ReLU(\boldsymbol{W}_4\,\boldsymbol{z} + \boldsymbol{b}_4) + \boldsymbol{b}_5\,, \tag{5}$$

where its input are codes sampled from the posterior $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. Now, both the prior and the approximated posterior are Gaussian. This allows a reparametrization of z that avoids derivation of the sampling procedure which is needed prior to be able to train the whole network using backpropagation [15, 16]. Moreover, the second term in (3) can be analytically derived resulting in (6), where J is the dimensionality of code space and j indicates each component.

$$D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})) = \frac{1}{2}\sum_{j=1}^{J}(1 + \log {\sigma_j}^2 - {\mu_j}^2 - {\sigma_j}^2) \tag{6}$$

3.3. Missing data imputation

As traffic data are real valued, we use the mean squared error (MSE) between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ as the reconstruction error term in (3) to train the whole network. Once trained, we can reconstruct a corrupted traffic data sample like in Fig. 2. First, MVs need to be random initialized. The resulting image is then encoded sampling from $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are given by the encoder (4), i.e., sampling from the inference model. Next, a reconstructed image $\hat{\boldsymbol{x}}$ can be obtained when the resulting code is mapped back to data space using the decoder (5), i.e., sampling from the generative model. Finally, this imputation procedure can be iterated until convergence simulating a Markov chain that has been shown that converges to the true marginal distribution of missing values given observed values [16].

4. EXPERIMENTATION

Experiments were conducted on data collected from 31 loop detectors installed on a south-bound section of Interstate 5 (I-5). Traffic data are available from the freeway Performance Measurement System (PeMS[1]) of the California Department of Transportation (Caltrans) which has been widely used in traffic forecasting literature. Detectors used span spaced equally apart 82.4 km of the highway in San Diego County, concretely from post mile (PM) 1.1 to 52.3. Each detector reports the speed, occupancy and flow. Data is aggregated into 5-minute intervals including a reliable measure of data quality showing the percent of observed samples. Incorrect values are filtered out while missing samples are imputed using linear regression [19]. Data collected covers the entire period from 2015 until 2017.

Although the dataset contained missing values, we could not directly use those for evaluation as their values were imputed. Instead, we considered the PeMS data quality measure and produced artificial missing data (11.28% on test data). We considered all 5-minute samples that do not meet an arbitrary defined 75% quality measure as missing values. Under this assumption, Fig. 3 shows mainly a Not Missing at Random (NMAR) pattern where consecutive missing values are found on not so random time instants and detectors. This is consistent with missingness types analyzed in literature [7, 14].

4.1. Evaluation task

We evaluated various imputation methods prior to the supervised regression task of DNN traffic forecasting (see Fig. 1 and Sec. 2 defined problem). In interest of faster training, we aimed to estimate 1 hour ahead ($h = 12$) traffic speed of sensor number 15 ($m = 1$), the one presenting less corrupted data (0.07%). The last 3 hours of traffic speed samples were used ($n = 36$) as input, $\boldsymbol{x} \in \mathbb{R}^{36\times 31}$. Evaluation was done on all possible 6-hour images containing MVs from 2016 (105360 samples) while the rest was used for training (105072 samples). IL and RL were trained using training data

[1] http://pems.dot.ca.gov

Table 1. $\overline{\text{RMSE}}$ $[km/h]$ — $\overline{\text{MAPE}}$ [%] results on test data. MCAR–(%) indicates the proportion of generated missing data. The data compression factor value is shown between parenthesis near each imputation method.

	Original	NMAR	MCAR–10	MCAR–20	MCAR–40
RL	5.53 — 3.04	19.37 — 13.50	27.24 — 20.05	30.07 — 22.75	33.28 — 26.20
PCA (11.16) + RL	N/A	12.42 — 7.82	10.68 — 6.79	14.35 — 9.40	18.46 — 12.84
AE (11.16) + RL	N/A	9.74 — 5.69	10.69 — 6.91	14.02 — 9.46	18.16 — 12.92
VAE (11.16) + RL	N/A	5.89 — 3.23	8.98 — 5.52	11.79 — 7.46	15.01 — 9.78
VAE (22.32) + RL	N/A	8.70 — 5.27	8.58 — 5.28	10.61 — 6.64	11.98 — 7.70
VAE (111.6) + RL	N/A	7.71 — 4.53	7.86 — 4.58	8.57 — 5.03	9.18 — 5.38

containing the imputed missing values by PeMS. Each experiment was conducted 10 times and we reported the mean of root mean square error (RMSE) and mean absolute percentage error (MAPE) in Table 1. We did not measure the error between original data and reconstruction because imputation requirements may vary depending on the final application.

IL. We compared our proposal of Sec. 3 (VAE) against a non-linear autoencoder (AE) and principal component analysis (PCA). All missing values were treated as zero prior to each imputation method for fair comparison. Details of VAE can be found in Sec. 3.2. On the AE, $ReLU$ was used for each hidden layer of 512 neurons except for the output. We trained both with a batch size of 128 using a random validation split of 10% for early stopping. We used Adam optimizer with a learning rate of $5e^{-4}$ [20]. Code dimension was first arbitrary set to 100 resulting in a data compression factor of 11.16. Input was normalized to zero mean and unit variance.

RL. We trained a 2-layer MLP where each hidden layer was composed of 100 neurons with sigmoid activations. $l2$ regularization was used to prevent overfitting. Input was normalized to zero mean and unit variance. The MSE was minimized using stochastic gradient descent with default Adam. Training was stopped using early stopping to ensure for the best generalization. RL architecture showed better performance compared to a naive approach (where the last input sample is used as estimation). On the original test data, RL showed a 34.8% and 25.3% improvement on RMSE and MAPE, respectively, which was considered as a benchmark for the evaluation purpose.

4.2. Results

The proposed VAE implementation showed an RMSE improvement of 69.6%, 52.6% and 39.5% over RL, PCA and AE on *NMAR*, respectively. Likewise, VAE showed superior performance for each different missing value proportion on *MCAR*. For example, on *MCAR–40*, VAE showed an RMSE improvement of 54.9%, 18.7% and 17.3% over RL, PCA and AE, respectively. The main difference between VAE and AE is that a regularizing term on the objective function is imposed on the former to force the model to learn a continuous code space. This indicates that learning the $p(\boldsymbol{X})$ helps to infer missing data as the model is able to decode plausible unseen data samples from every point in the latent space that has a reasonable probability under the prior, which validates our initial assumption. We also found that non-linearity helps to impute MVs when larger gaps of missing data are found (NMAR pattern). Looking at the VAE and AE performance against PCA in Table 1 on NMAR data, the linear model performs poorly. However, no relevant differences were found between PCA and AE on *MCAR*. In this case, the PCA performs similarly to AE because of the MCAR pattern which implies less consecutive missing values, thus the linear model is able to perform better. Another interesting finding is that VAE performed better in NMAR than MCAR–10 even when the missing data proportion of the former is greater. We also varied VAE's code dimension and provided some results on Table 1. Results showed that accuracy increased jointly with the compression factor but to a certain extent. Constraining code space forces the network to learn better features until the space becomes small enough. Same thing happened while increasing the code dimension. This suggested the existence of a lower and higher bound where only an insignificant improvement can be observed, which led us to conclude that the dimension of the code must be empirically defined. All this makes the proposed method suitable for real-world dataset where mostly NMAR patterns are found (e.g., Fig. 3).

5. CONCLUSION & FUTURE WORK

In this work, we proposed an implementation of VAE that was able to capture the traffic data distribution and therefore its underlying characteristics of traffic. We showed that DNN traffic forecasting accuracy deteriorates with increasing missing data proportion and thus we evaluated different imputation methods prior to traffic speed forecasting. Our proposal showed superior performance against other approaches for each evaluation task (e.g., 52.6% RMSE improvement). Besides, the learned latent space can be exploited for data compression, data augmentation, traffic classification and anomaly detection which are left as future work.

6. REFERENCES

[1] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias, "Short-term traffic forecasting: Where we are and where were going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[2] Nicholas G Polson and Vadim O Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[3] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818, 2017.

[4] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng 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.

[5] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[6] Samuel Dodge and Lina Karam, "Understanding how image quality affects deep neural networks," in *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*. IEEE, 2016, pp. 1–6.

[7] Roderick JA Little and Donald B Rubin, *Statistical analysis with missing data*, vol. 333, John Wiley & Sons, 2014.

[8] S van Buuren and Karin Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of statistical software*, pp. 1–68, 2010.

[9] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Advances in neural information processing systems*, 2016, pp. 847–855.

[10] Pinlong Cai, Yunpeng Wang, Guangquan Lu, Peng Chen, Chuan Ding, and Jianping Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[11] Yuebiao Li, Zhiheng Li, and Li Li, "Missing traffic data: comparison of imputation methods," *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 51–57, 2014.

[12] Jinsung Yoon, James Jordon, and Mihaela van der Schaar, "Gain: Missing data imputation using generative adversarial nets," *arXiv preprint arXiv:1806.02920*, 2018.

[13] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi, "Vigan: Missing view imputation with generative adversarial networks," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 766–775.

[14] Lovedeep Gondara and Ke Wang, "Multiple imputation using deep denoising autoencoders," *arXiv preprint arXiv:1705.02737*, 2017.

[15] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[17] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, Fei-Yue Wang, et al., "Traffic flow prediction with big data: A deep learning approach.," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[18] Hao-Fan Yang, Tharam S Dillon, and Yi-Ping Phoebe Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2371–2381, 2017.

[19] Chao Chen, Jaimyoung Kwon, and Pravin Varaiya, "The quality of loop data and the health of californias freeway loop detectors," *PeMS Development Group*, 2002.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.