A variational autoencoder solution for road traffic forecasting systems: missing data imputation, dimension reduction, model selection and anomaly detection

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

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

Abstract

Efforts devoted to mitigate the effects of road traffic congestion have been conducted since 1970s. Nowadays, there is a need for prominent solutions capable of mining information from messy and multidimensional road traffic data sets with few modeling constraints. In that sense, we propose a unique and versatile model to address different major challenges of traffic forecasting in an unsupervised manner. We formulate the road traffic forecasting problem as a latent variable model, assuming that traffic data is not generated randomly but from a latent space with fewer dimensions containing the underlying characteristics of traffic. We solve the problem by proposing a variational autoencoder (VAE) model to learn how traffic data are generated and inferred, while validating it against three different real-world traffic data sets. Under this framework, we propose an online unsupervised imputation method for unobserved traffic data with missing values. Additionally, taking advantage of the low dimension latent space learned, we compress the traffic data before applying a prediction model obtaining improvements in the forecasting accuracy. Finally, given that the model not only learns useful forecasting features but also meaningful characteristics, we explore the latent space as a tool for model and data selection and traffic anomaly detection from the point of view of traffic modelers.

Keywords: Intelligent Transportation Systems, Traffic Forecasting, Missing Data Imputation, Dimension Reduction, Anomaly Detection, Model Selection
2019 MSC: 00-01, 99-00

1. Introduction

Due to recent developments in Intelligent Transportation Systems (ITS), road traffic forecasting conforms a vivid area of research, policy making and technology development. One of its foundations is to predict traffic characteristics, since traffic congestion generates important social, economic and environmental problems [1]. Efforts devoted to mitigate the effects of traffic congestion have been conducted from the 1970s to the present in traffic flow management with the use of Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS). Those systems have been continuously improving and will continue to with the expansion of technology and data provided by sensors in roads and vehicles, the envisaged Vehicle-to-Everything (V2X) paradigm. In that context, data-driven approaches like deep neural networks (DNN) have arisen as a prominent solution as they are capable of mining information from messy and multi-dimensional traffic data sets with few modeling constraints [1, 2]. However, the intrinsic characteristics of road traffic still makes the forecast a challenging problem because of complex spatial dependency on road networks [3], non-linear temporal dynamics with changing road conditions and inherent difficulties of long-term forecasting [4]. In addition to the forecasting problem, more challenges of equal magnitude derive from said

*Corresponding author
Email address: guillem.boquet@uab.cat (Guillem Boquet)

context. To name a few, the quality of data, arterial and network-level predictions, spatiotemporal forecasts and model selection techniques are identified as current major challenges of future road traffic forecasting [1, 2].

Two main categories can be distinguished in data-driven approaches that address the traffic forecasting problem: parametric and non-parametric models. The parametric category is represented by the autoregressive integrated moving average (ARIMA) [5], which remains popular (e.g., seasonal ARIMA coupled with the Kalman filter [6]) despite relying on the assumption of stationarity of traffic data. Non-parametric models are the k-nearest neighbor algorithm (KNN) [7], support vector regression (SVR) [8] and K-means [9], but the current trend is focused on DNN models that show better better results due to advances in deep learning techniques [10, 11]. For instances, convolutional neural networks (CNN) [12] and recurrent neural networks (RNN) [13] are coupled into more complex and deeper networks to characterize the spatiotemporal behavior of traffic and increase the accuracy of traffic forecasting systems [4, 14, 15, 16, 17]. Nevertheless, all these supervised methods are feed with raw data without any or few preprocessing steps. All available data are used for training, which may not be the best option for optimal forecasting performance. Data quality and missing values ??are not considered, which is known to degrade system performance. In addition, an excessive number of features available from data sources are used to forecast, which is computational inefficient and undertakes the risk of over-fitting.

In this manuscript, we propose a unified solution to the aforementioned problems, which have been addressed separately in the literature. Following the DNN trend, we propose to merge the recent advances in variational inference [18] with traffic forecasting systems as we assume and show that the traffic forecasting problem can be formulated as a latent variable model and resolved in an unsupervised manner using the Variational Autoencoder (VAE) framework [19, 20]. According to the best of authors' knowledge, this is the first work that implements a VAE-based model for traffic data. Here, we propose a unique and versatile framework to solve the following major challenges of road traffic forecasting:

1. how to impute missing data to not deteriorate the performance of forecasting systems
2. how to extract useful features while compressing the traffic data
3. how to select the best model and data for a specific traffic estimation problem
4. how to detect anomalous traffic

To answer that, in this work we learn the underlying structures that generate real-world traffic data. We assume that road traffic is not generated randomly, but from a latent subspace, based on the assumption that there are strong spatiotemporal relationships and seasonality between points in the road network. We formulate it as a latent variable model that forces us to approximate the joint probability distribution via variational inference. Thus, we base our model on VAE, which we show is able to learn an approximation of the data distribution of three different real-world traffic data set. Under this framework, the posterior distribution of the latent space is forced to be continuous, which allows the model to decode plausible traffic samples from every point in the subspace, therefore, to online impute unobserved missing traffic data without supervision. In said subspace, traffic of the same class ends up closer together, allowing unsupervised traffic classification and at the same time detecting anomalous traffic. The latent space dimensions are constrained, which results in learning useful properties of traffic to compress the data, that is, feature extraction. Moreover, since VAE is a generative model, the model allows traffic modelers or practitioners to sample from the learned distribution to generate new traffic data and the possibility to explore into the meaningful latent representations. Now, using the contributions of this manuscript, a traffic modeler can implement a model to compress the traffic data and efficiently forecast, impute missing values, select the best data and model for a specific problem and detect anomalous traffic data at the same time with no additional knowledge required. Before, if an ITS that already estimated traffic required addressing other traffic problems, such as missing data imputation or data compression, it was forced to implement other models, resources and practitioners to solve it.

In summary, our main contributions are:

- The applicability of the VAE model for real-world road traffic data as a unified solution to for various traffic forecasting problems.

- A novel online multidimensional imputation method for missing values in road traffic data based on learning the probability distribution of the data given the observed values.

- A novel dimension reduction approach to traffic data to improve efficiency and accuracy of forecasting systems by learning powerful characteristics of traffic in an unsupervised manner.

- A novel tool for traffic modelers using projected data in a unsupervised learned subspace with meaningful dimensions that can be used for model and data selection and anomaly detection.

- Three case studies on real-world data from USA (California) and UK (England) that validates the usefulness of the proposed framework.

2. Related work

To the date of this manuscript, the most up-to-date reviews on road traffic forecasting are Laña et al. in [1] and Vlahogianni et al. in [2]. Our work is related to the current applications of neural networks for road traffic forecasting in different ways. We divided the related work discussion accordingly as we are proposing a transversal solution to different problems that have been addressed separately in literature.

Missing data imputation. Road traffic forecasting systems are deployed in scenarios where sensor and system failure are common. In these scenarios, the missing values ??are known to negatively affect the precision of the forecast [21] although they are often underestimate in current forecast models [1, 2]. The current strategy is to preprocess the data by inferring the missing values from the known part of the data. Three well-known imputation methods in traffic forecasting are ARIMA, KNN and principal component analysis (PCA)-based methods. [22] compared them among others and the results showed that the probabilistic PCA is the most effective in terms of performance and implementation. More recently, [23] proposed a spatial context sensing model based on an automated clustering analysis tool and the information provided by surrounding sensors. [24] proposed a model that combines long-short term memory (LSTM), SVR and collaborative filtering. With a similar approach to ours, [25] proposed a Bayesian imputation model to characterize the data generation process and learn underlying statistical patterns in traffic data.

On the other hand, state-of-the-art imputation methods from other research fields can be classified [26] as either *discriminative*, such as multiple imputation by chained equations (MICE) [27] and matrix completion [28], or *generative* methods based on DNN. For example, [29] proposed an overcomplete denoising autoencoder (DAE) to be able to reconstruct data by stochastically corrupting it. Closer to our work, [30] and [31] proposed a RNN-based VAE which succeed at imputing missing words from sentences. [32] applied a deep sequential VAE with a Gaussian process prior in the latent space to capture temporal dynamics to impute real-world medical data. Similarly, [33] and [34] proposed also a generative model imputation method but using generative adversarial networks (GAN). Contrary to GAN, our VAE-based approach possesses certain desirable properties for traffic forecasting systems, such as stable training [35], interpretable encoder/decoder network [36] and outlier-robustness [37].

Dimension reduction. The number of features available from data sources jointly with the number of available data points in road networks are excessive. Forecasting with all those features can be computational inefficient and undertakes the risk of over-fitting. Therefore, it is essential to reduce the dimension of the feature space before applying a prediction model [38]. Reduction of the data is done by learning the principal components or independent factors of a given data manifold, i.e., feature extraction. Recently, a systematic literature review of feature selection and extraction in spatiotemporal traffic forecasting was reported in [39]. Note that feature extraction does not necessarily mean reducing the dimension of the data space, that is, dimension reduction is a subclass of feature extraction methods. The low-dimensional representation is traditionally obtained by PCA approaches that had been widely used to extract the linear correlations between the variables [40, 38]. In addition, the least absolute shrinkage and selection operator (LASSO) is a well-known technique used [41, 38]. On the other hand, [42] exploited compressed sensing to reduce the complexity of road networks prior to regression. In DNN data-driven approaches, RNN and CNN are used to extract temporal and spatial characteristic within the regression model. [16] used a LSTM and CNN mixed with an attention layer but not as an independent layer to the regression task like our proposal. Similarly to our work, features learned from a stack of autoencoders (SAE) have been previously used in literature to improve traffic forecasting [10, 43]. Contrary to the autoencoder, the VAE encourages the model to generalize features and reconstruct samples as an aggregation of those, forces the latent space to be continuous and is a generative model. Likewise, other VAE approaches have been used successfully for dimensionality reduction within other research fields, such as fault diagnosis [44] and towards sequencing the RNA of individual cells [45].

Model explanatory power. The explanatory and representative power of models is valuable for traffic modelers to obtain information on how transportation networks behave and evolve. Some efforts have been devoted to explain the behavior of the models in the literature as a second derivative of traffic forecast. For example, [46] analyzed the spatial features captured by CNN through characterizing the information that retained layer by layer. [41] discussed how the input variables relate to the predicted output using the coefficients of the fitted linear model. In that context, our proposal learns the probability distribution of the given traffic data and a low-dimension continuous latent space. Thus, we exploit these features as a tool to perform model selection and anomaly detection for traffic forecasting systems.

Model selection. There is no best method that suits all situations in traffic forecast [47], which implies an applicability at a higher level of the method to choose the most suitable model given the characteristics of the forecasting problem [1]. Traffic modelers frequently face several optimization challenges related to model selection, while there are no clear baselines to find the best method and its configuration [48]. According to the best of authors' knowledge, few works are related to the traffic forecast context. [49] proposed a metamodeling technique to optimize both algorithm selection and hyperparameter setting and [48] explored the use of Auto-WEKA, an automatic algorithm selection method. On the contrary, we approach the problem from a data perspective. We provide a tool based on the clustering of data in the learned latent space to select the data from which the best forecasting model will be built to solve a specific problem. Similar to our approach, [50] proposed a hybrid method of short-term traffic forecasting using a self-organized Kohonen map as an initial classifier where each class had an individually associated tuned ARIMA model. But, according to the best knowledge of the authors, there is no work that presents a VAE applied to the selection of the model or data.

Anomaly detection. One of the main applications of urban traffic analysis lies in detecting anomalies from traffic data [51]. Recently, [51] and [52] reviewed on existing outlier detection techniques in traffic data in three main categories: statistical, similarity-based, and based on pattern analysis. Among them, some find outliers in subspaces, which is exactly what VAE can provide. In [53], dimensionality reduction is performed by PCA and a kNN-based outlier detection is applied in the derived subspace. On the contrary, the VAE is based on a DNN that has greater modeling capabilities. In fact, a linear autoencoder learns to span the same subspace as PCA. In the learned latent space by our proposed model, traffic samples are clustered and projections close to each are forced to have similar reconstructions that help in the detection of outliers. Moreover, when our proposal is trained with much more normal samples than the anomalous ones, the reconstruction errors of normal data are relatively higher than those of anomalous data. Therefore, the VAE loss function provides an anomaly score function which can be exploited as an anomaly detection technique. In that regard, VAE-based outlier detection methods had been used successfully in other research fields: [54] added a supervised method to the VAE approach to enhance detection of seen anomalies without degrading the performance for unseen anomalies on real industrial data, [55] proposed a a RNN-based VAE to detect anomalies on robot time series data and [56] proposed an anomaly detection method based on a reconstruction probability derived from the VAE loss function.

3. Methodology

In this section, we formulate the traffic forecasting problem as a latent variable model. We propose a VAE implementation to learn how traffic data are generated and to learn an approximation of the probability distribution of the data. Then, we propose a procedure to impute missing values. Finally, we exploit the learned latent subspace proposing a

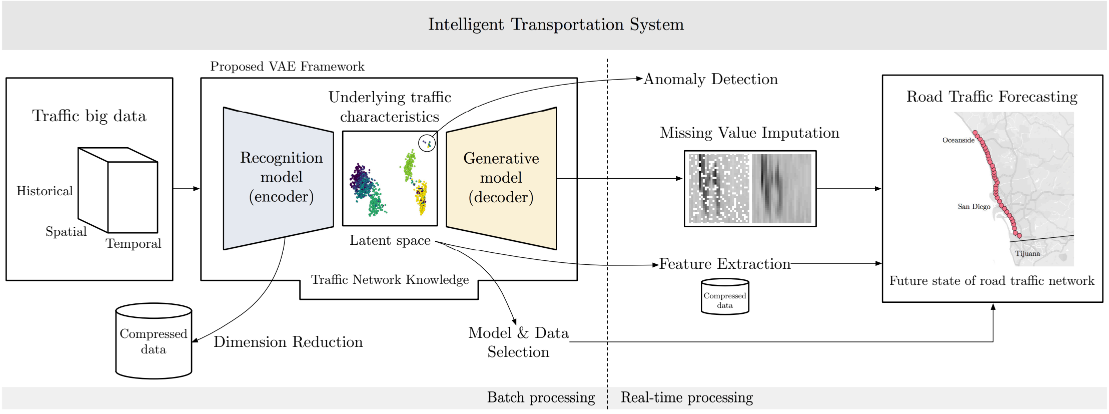


Figure 1: Interrelation of the proposed VAE model with different traffic forecasting tasks. The proposal is a unified framework that aims to solve several important challenges of ITS in road traffic forecasting.

tool for model and data selection, dimension reduction and anomaly detection. A general framework diagram that links the model to the different traffic prediction tasks is depicted in Fig. 1.

3.1. Problem definition

Let $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ be a historic traffic data set composed of N observed traffic variables or *traffic samples* with an unknown ground-truth probability distribution, $\boldsymbol{x}^{(i)} \sim p_{gt}(\boldsymbol{x})$. Let each element within $\boldsymbol{x}^{(i)}$ represent a value of a traffic variable associated with time and space, $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n \times d}$ where n is the number of past traffic variables and d the number of traffic sensors deployed into the road network. The method does not use information about the position of the sensors along the route. Henceforth, the superscript $^{(i)}$ denoting the i-th sample is omitted to avoid clutter, except in cases where some ambiguity may exist. Note that $\boldsymbol{x}$ is real-valued, which is intended to represent traffic variables such as speed, flow, density, etc., hence, the methodology presented during this section will be derived accordingly. Likewise, let $\boldsymbol{y} \in \mathbb{R}^{m}$ denote the future state of $m \leq d$ subset of sensors in the time horizon of h samples. The traffic forecast problem can be modeled as $\boldsymbol{y} = f^{*}(\boldsymbol{x})$, where forecast systems aim to make an accurate estimate of $\boldsymbol{y}$ from $\boldsymbol{x}$, while the challenge remains on deriving a function that closely resembles f^{*}.

Suppose we want to infer the traffic behavior during the next two hours or that we have a partially occluded traffic sample due to a sensor or system failure. Missing data could be anything if there is no underlying structure (or subspace) from which the data are generated. In that sense, we know that strong spatio-temporal relationships exist between road network's points [1]. For instances, due to seasonality, it is possible to discern between a work day or not just by observing how morning traffic develops through time and space.. Therefore, this can be seen as a generative model, where data can be generated from a latent manifold with fewer dimensions, or as a reconstruction problem, where applying a function to a corrupted input $\tilde{\boldsymbol{x}}$ derives the actual input.

As with the forecasting problem, this function can be approximated using DNN to obtain an estimate $\hat{\boldsymbol{x}}$ of the actual input. Next, based on the previous assumption, we formulate the problem as a latent variable model and solve that using a deep learning variational inference approach.

3.2. Latent variable model

Let $\{\boldsymbol{z}^{(i)}\}_{i=1}^{N}$ be the set of vectors composed of continuous random latent variables defining a low-dimensional representation of the significant factors of variation in $\boldsymbol{X}$, $\boldsymbol{z} \in \mathbb{R}^{J}$ with $J \ll dim(\boldsymbol{x})$. Thus, whenever a traffic sample $\boldsymbol{x}$ is feed to the model, $\boldsymbol{z}$ will represent its underlying characteristics. Let f denote a deterministic function derived from a neural network that maps $\boldsymbol{z}$ to the data space and

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

the generative model (Fig. 2) parametrized with $\boldsymbol{\theta}$, where $\hat{\boldsymbol{X}}$ is the estimate of $\boldsymbol{X}$. 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 $\boldsymbol{\theta}$.

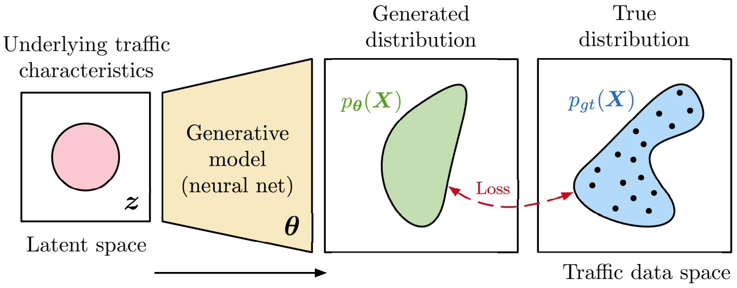


Figure 2: The generative model. The goal is to learn a parametric model capable of producing M new traffic samples $\{x^{(j)}\}_{j=1}^{M}$, $x^{(j)} \sim p_{\theta}(X) \approx p_{gt}(X)$. That is, let the model learn the ground-truth distribution of the data $p_{gt}(X)$ from a random variable z with a simple distribution that captures the underlying characteristics of the traffic in the given road traffic network. Black dots are the observed traffic samples $x^{(i)}$.

Because we assumed that x is generated by a random process involving z, this could be solved by integrating over the joint probability distribution

$$p_{\theta}(x) = \int p_{\theta}(x|z)\, p(z)\, dz \tag{2}$$

while maximizing $\log p_{\theta}(X) = \sum_{i=1}^{N} \log p_{\theta}(x^{(i)})$, a maximum likelihood problem. Note that by maximizing (2) we hope to discover a meaningful representation for the traffic data x in terms of latent features given by $p(z|x)$. Unfortunately, the integral of the marginal likelihood requires computing the intractable true posterior or sampling-based solutions, which are too costly [19, 20]. To circumvent this, (1) is treated as an optimization problem adding a recognition model q to approximate the intractable true posterior $p(z|x)$. Consequently, the whole data model may be viewed as consisting of two parts (Fig. 3):

- the generative model $p_{\theta}(x, z) = p(z)\, p_{\theta}(x|z)$, which produces a distribution over the possible corresponding values of x given z and
- the added recognition model $q_{\phi}(z|x)$ parameterized with ϕ, that given a traffic sample x, produces a distribution over the possible values of z from which x could have been generated.

From there, a variational lower bound on the marginal likelihood can be derived which does not depend on the intractable $p_{\theta}(z|x)$ or $p_{\theta}(x)$. This yields the known ELBO function [19, 20], which can be optimized in terms of ϕ and θ at the same time minimizing

$$\begin{aligned}\mathcal{L}(\theta, \phi) &= -\mathbb{E}_{z \sim q_{\phi}(z|x)}\left[\log p_{\theta}(x|z)\right] \\ &\quad + D_{KL}\left(q_{\phi}(z|x) \,||\, p(z)\right) \\ &\quad \leq \log p_{\theta}(X)\,,\end{aligned} \tag{3}$$

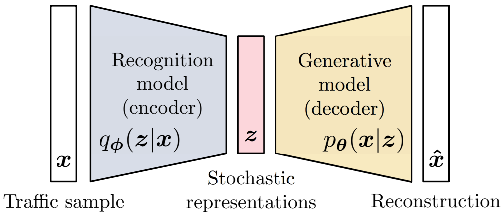

Recognition model (encoder)
$q_{\phi}(z|x)$
x
z
Generative model (decoder)
$p_{\theta}(x|z)$
$\hat{x}$
Traffic sample
Stochastic representations
Reconstruction

Figure 3: The VAE framework adds a recognition model q to approximate the intractable true posterior distribution $p(z|x)$. This can be optimized via stochastic gradient descent (SGD) as the input and output of the model should be the same. The model might not learn the exact data distribution $p_{gt}(X)$ but an approximation, (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.

3.3. Model implementation

The aforementioned model is implemented using VAE [19, 20]: the recognition model using a neural network encoder with weights and biases ϕ and the generative model using a neural network decoder with weights and biases θ, Fig. 3. We model the encoder and decoder using a multilayer perceptron (MLP), considering that during this work we will be experimenting with speed and flow data separately. However, thanks to the versatility and continuous development of neural networks, any architecture could be used as part of the encoder/decoder and even stack several traffic variables as input to the network. For example, [14] treated x as an image to exploit the spatial and temporal correlation information between road network's points using CNN.

Encoder. We let the encoder model a multivariate Gaussian with a diagonal covariance structure, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_z, diag(\boldsymbol{\sigma}_z^2))$. The encoder is implemented using a 1 hidden layer MLP with $\boldsymbol{\phi} = \{W_1, W_2, W_3, b_1, b_2, b_3\}$, whose outputs are the mean $\boldsymbol{\mu}_z$ and s.d. $\boldsymbol{\sigma}_z$ of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ (called the code or latent representation of the data):

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

where REctified Linear activation Unit ($ReLU$) and Leaky ReLU ($LReLU$) are nonlinear activation functions and $\boldsymbol{\sigma}_z$ is filtered through a $ReLU$ while lower bounded ($b_\sigma = 1e^{-5}$) to help for numerical stability during training, since the covariance is positive definite.

Decoder. Likewise, we let the decoder model a multivariate Gaussian, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{I})$. The decoder is implemented using a 1 hidden layer MLP with $\boldsymbol{\theta} = \{W_4, W_5, b_4, b_5\}$, whose output is defined by

$$\begin{aligned} \boldsymbol{h}^{dec} &= LReLU(\boldsymbol{W}_4\,\boldsymbol{z} + \boldsymbol{b}_4) \\ \hat{\boldsymbol{x}} = \boldsymbol{\mu}_x &= \boldsymbol{W}_5\,\boldsymbol{h}^{dec} + \boldsymbol{b}_5\,, \end{aligned} \tag{5}$$

where its input are codes sampled from the posterior $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and $\hat{\boldsymbol{x}}$ is computed using only $\boldsymbol{\mu}_x$. Note that during training via SGD we cannot directly propagate gradients w.r.t. $\boldsymbol{\phi}$ through the sampling operator. Thus, we make use of the *reparametrization trick* in [19] as an equivalent sampling procedure that avoids derivation of $\boldsymbol{z}$, Fig. 4.

Now, we let the prior over the latent variables be the centered isotropic multivariate Gaussian $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$. This jointly with the KL term in (3) allows for a continuous latent space and assumes that latent representations of samples are iid. Notice that any distribution in d dimensions can be generated by taking a set of d variables that are normally distributed and mapping them through non-linear functions. Thus, the model can learn to generate any distribution of traffic data from the Gaussian assumption on the prior. In addition, this may lead into learning more disentangled features because $\boldsymbol{z}$ components are orthogonal, which may help traffic modelers to explore and interpret the latent space.

As both the prior and the approximated posterior are Gaussian, the KL divergence in (3) can be analytically derived [19] resulting in

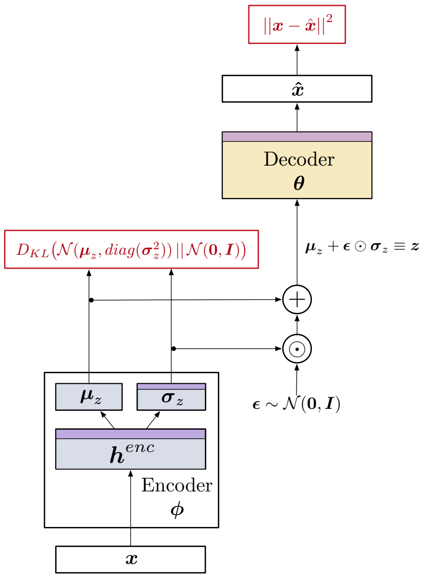

$||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$
$\hat{\boldsymbol{x}}$
Decoder
$\boldsymbol{\theta}$
$D_{KL}(\mathcal{N}(\boldsymbol{\mu}_z, diag(\boldsymbol{\sigma}_z^2))\,||\,\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}))$
$\boldsymbol{\mu}_z + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_z \equiv \boldsymbol{z}$
$+$
$\odot$
$\boldsymbol{\mu}_z$
$\boldsymbol{\sigma}_z$
$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$\boldsymbol{h}^{enc}$
Encoder
$\boldsymbol{\phi}$
$\boldsymbol{x}$

Figure 4: Feed forward pass of the network using the *reparametrization trick* for training via SGD. Red and purple show the loss and non linear activation layers, respectively. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are updated on the backward pass with the backpropagation of the error.

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

where J is the dimension of $\boldsymbol{z}$ and j indicates each component of the encoder moments $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$. In addition, we estimate the expectation of the reconstruction error in (3) using a single sample from $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, as it is enough in practice when using SGD. We minimize the l_2 norm between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ analogously to maximize $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, since the variance for the recognition model is fixed to 1 and speed and flow data are real-valued. Hence, the objective function (3) to minimize becomes

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2 - \frac{1}{2}\sum_{j=1}^{J}(1+\log{\sigma_j}^2-{\mu_j}^2-{\sigma_j}^2)\,, \tag{7}$$

where the second term (6) is a regularization or penalty term during training in an autoencoder sense.

3.4. Model optimization

Regarding the encoder/decoder architecture, [57] stated that simple decoders like a conditional unimodal Gaussian decoder (5) typically results in representations that are good at capturing the global structure but fail at capturing more complex local structure. To address this, autoregressive models (e.g., PixelRNN [58] or PixelCNN [59]) may be integrated with VAE and used as encoder/decoder like in the Variational Lossy Autoencoder [60], or use the Channel-Recurrent Variational Autoencoder [61] that uses recurrent connections across CNN channels to circumvent the simplification of VAE's latent space. These models may be a powerful tool for traffic forecasting [62] because they are good at capturing local statistics [63]. However, even that the architecture presented in Sec. 3.3 is not that complex, we show in the experimentation section that is enough to solve general road traffic forecast problems with the data dealt in this work. This is aligned with recent findings suggesting that Gaussian encoder/decoder assumptions do not reduce the effectiveness of VAEs [64].

Regarding the definition of $p(\boldsymbol{z})$, we let $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})$ for our purpose, which has the following computational and implementation benefits: (i) the samples of z can be drawn from a simple distribution, (ii) it forces a continuous latent space and (iii) the KL divergence is given in closed form. However, by doing so, we assumed that latent representations of samples are iid, which for many datasets, such as time-series of images, can be a strong assumption [65].

Nevertheless, increasing the complexity of the model leads to several issues identified in literature [18]. Therefore, the model should match the complexity of the problem and data because this substantially increases the complexity of model implementation, training and tuning. Ladder Variational Autoencoder (LVAE) [66] were proposed to train deeper architectures for more representational power, but it is known that a VAE with powerful decoding capabilities tend to ignore latent space and use only the decoding distribution to represent the entire data set [67]. We found that in most cases a straightforward implementation ignored the latent space, that is, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ was learned by setting $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \sim p(\boldsymbol{z})$ thus bringing the KL term close to zero (the *posterior collapse* problem [68]). Note that a model that encodes useful information in the latent variable $\boldsymbol{z}$ will have a non-zero KL divergence term. To prevent that, we modified the training objective (7) by weighting the second term with $\beta \in [0, 1]$ and increasing its value on each epoch during training. This *annealing strategy* [30] yielded to better results, despite not optimizing the proper lower bound during the early stages of training.

Finally, since road traffic networks evolve with time, the reconstruction error of new traffic samples can be used as an indicator of when to adjust the model to new data. A high reconstruction error would mean that samples reconstructed conform to a different data distribution than the already learned by the model.

3.5. Missing data imputation

The first implication of Sec. 3.3 model is that new unobserved traffic samples with missing values can be reconstructed from the learned $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. A corrupted data sample $\tilde{\boldsymbol{x}}$ can be reconstructed once the whole network is trained on historical data minimizing (7). The imputation procedure depicted in Fig. 5 consists of:

– random initialize the missing values,

– sample from the recognition model, i.e., encode $\tilde{\boldsymbol{x}}$ sampling from $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z)$ where $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$ are given by the encoder (4) and

– sample from the generative model, i.e., map back the resulting $\boldsymbol{z}$ to the data space using decoder (5) to obtain a reconstructed data sample $\hat{\boldsymbol{x}}$.

This procedure can be iterated until convergence, simulating a Markov chain that has been shown in [20] that converges to the true marginal distribution of missing values given observed values. In practice, a more straightforward method is to sample only using the mean, i.e., $\boldsymbol{z} = \boldsymbol{\mu}_z$, which leads to similar results. Note that (6) forces the model to be able to decode plausible traffic samples from every point in the latent space that has a reasonable probability under the prior. On the contrary, an autoencoder without the latent variable model would have learned a latent space which may not be continuous or allow interpolation.

3.6. Dimension reduction

The second implication is that the learned latent space can be exploited in several different ways that are of interest to traffic forecasting systems. The latent space defined by $\boldsymbol{z}$ is forced to capture useful information about the data because $\boldsymbol{z}$ is limited to having a dimension smaller than $\boldsymbol{x}$. Therefore,

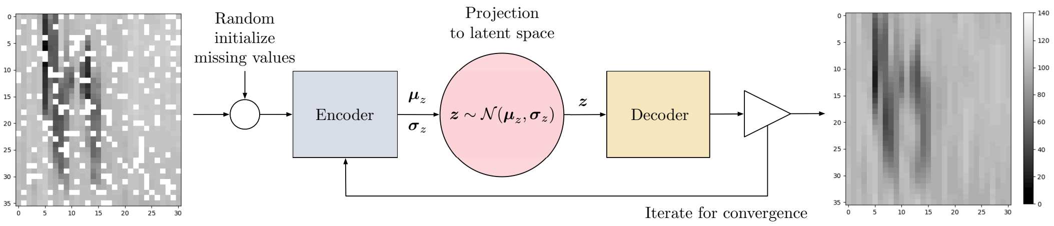


Figure 5: Reconstruction of a corrupted sample using the imputation procedure of Sec. 3.5 on *PeMS* data. y-axis shows 3 hours of 5-minute samples. x-axis represents 31 sensors. The colored variable is the traffic speed $[km/h]$.

VAE is learning the principal components or independent factors of the highly non-linear latent manifold of the given traffic data set. Recall that a linear autoencoder minimizing the mean squared error (MSE) learns to span the same subspace as PCA. This can be exploited as an unsupervised dimension reduction or feature extraction independent layer for traffic forecasting systems. On the one hand, the data can be compressed using the encoder to store and reconstruct them when necessary using the decoder. To aim for the lowest possible dimension of $\boldsymbol{z}$ (i.e., maximum compression) one may find it empirically or rely on the recently published article [69], which inspects the mutual information evolution between layers. On the other hand, features learned may be used by a regression layer to improve traffic estimation as the compressed information filters out useless information and allows data-driven models to easily learn. In this case, the performance is less conditioned to the dimensions of $\boldsymbol{z}$ since in practice we have obtained similar results for different latent space dimensions, except with very small or very large dimensions (the reader is referred to [70] for a theoretical explanation of that). Finally, the whole procedure consists of:

- pre-train the model to reconstruct its input in an unsupervised manner
- use the pre-trained encoder as input to a regression model for supervised traffic estimation
- fine-tune the entire network if the regression model is a DNN (we found that yields better results than fixing the weights and biases of the encoder), if not, supervise train with the latent representations.

It is worth mentioning that, unlike SAE [10, 43], stacking several VAE in our framework would not lead to an enhanced representation power [71]. Also, currently, researchers have put an effort to learn meaningful (humanly interpretable) and disentangled latent dimensions with VAE [72, 73].

3.7. Model selection

The latent space can be exploited as a tool for the selection of models and data, since similar data is encoded closer in the latent space. Traffic samples are clustered in an unsupervised manner in the latent space learned by VAE. This can be used to distinguish between work days, weekends, holidays, anomalous days, etc. or to compare the traffic from different road traffic networks and time periods. This explanatory power makes the model adaptable and responsive to dynamic traffic and road environment changes over time. Traffic modelers may use the tool as an indicator of model performance against new data, thus, the need to train a new model, or to gain deeper knowledge of the traffic behavior by exploring the latent space. In that way, accuracy of traffic forecasting systems can be enhanced by splitting the data into the classes learned by the model and fitting a separate model to each class [50]. This can be done by projecting the new data into the learned subspace and comparing it with new data using clustering algorithms [53]. Further, modelers can visually search for correlations and seasonality by using visualization techniques of high-dimension data sets such as PCA or t-SNE [74], as we show in the experimentation section.

3.8. Anomaly detection

The anomaly detection with VAE can be done online and offline. A simple but powerful approach is to visually compare projected samples in latent space, which may be useful for traffic modelers. For example, by projecting the samples in the latent space using PCA and displaying them colored by type of day, the modeler can see if a Tuesday sample deviates significantly from his cluster, which may mean that an anomaly is occurring or that it is a holiday if it's closer to Sunday's cluster.

On the other hand, a more interesting scenario for ITS is to detect anomalies automatically. For statistical methods, key statistics are used when anomalies are detected if the statistic exceeds a certain threshold value. If anomalies are labeled, one may project a sample to the latent space, compute the Euclidean distance of the sample to its class centroid and then establish the threshold, for example by means of the AUROC. If anomalies are not labeled, one may assume that clusters are Gaussian distributed and set the threshold proportional to the s.d. or use kernel density estimation setting a minimum probability threshold. Nevertheless, VAE inherently provides the two typically steps of statistical anomaly detection techniques: dimension reduction and an statistical anomaly criterion. VAE provides a probability measure with the KL divergence term in (7) rather than a reconstruction error as an anomaly score function. Probabilities are more objective than reconstruction errors and do not require model specific thresholds for judging anomalies [56]. When the VAE is trained with far more normal samples than anomalous ones, the VAE learns to model the distribution of normal traffic data, thus a traffic sample can be detected as anomalous if it statistically deviates from what the model has learned [54]. This particularly suits the traffic domain because traffic data sets are usually imbalanced, samples are only labeled by days and most of the anomalies are still unseen. We have explored some of this suggestions under Sec. 7 and left the others as future work.

4. Real-world traffic data sets

We gathered and cleaned three different kind of real-world data that are described in this section to later evaluate the proposed methodology, Fig. 6. It should be noted that there is a lack of benchmark data sets in traffic forecasting literature that has been identified as a problem to compare different proposals [1]. The three datasets come from highways, however, please keep in mind that the method can also be applied to any road traffic network and urban areas. Its success depends on the encoder/decoder architecture chosen to mine the existing relations between sensors. In that sense, literature has shown that neural networks are capable of mining spatio-temporal characteristics to perform forecast on complex networks [14]. During experimentation all code was written in Python with the help of Tensorflow library for DNN coding. All testing was performed with an Intel Xeon W-2123 + 4 × NVIDIA GeForce RTX 2080 Ti on an Ubuntu server.

4.1. Data description

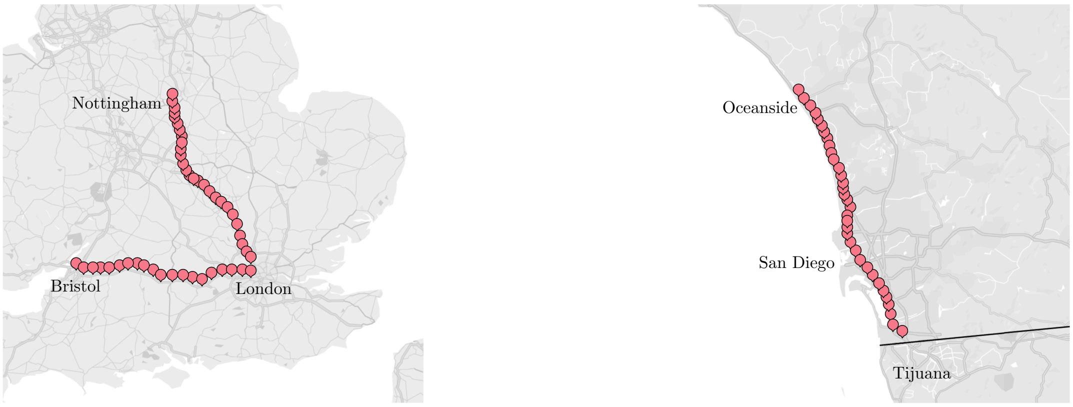
PeMS. We collected data from 31 loop detectors installed on a south-bound section of Interstate 5 (I-5), Fig. 6b. Traffic data are available from the freeway Performance Measurement System (PeMS) of the California Department of Transportation (Caltrans)[1] that has been widely used in traffic forecasting literature. Detectors used span spaced equally apart 82 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, which are 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 [75]. Data collected covers the two-year period from 2015 until 2017.

UKM1. We gathered traffic speed and flow data[2] from 19 junctions (J27 to J1) of the English M1 section from Nottingham to London, Fig. 6a. The M1 is a major motorway of the Strategic Road Network (SRN) which runs between London to Leeds in the United Kingdom. The data are averaged between junctions and aggregated into 15-minute intervals. Junctions span 210 km and consist of different road lengths. Speeds are estimated using a combination of sources, including automatic number plate recognition (ANPR) cameras, in-vehicle global positioning systems (GPS) and inductive loops built into the road surface. Data collected covers the four-year period from 2011 until 2015.

UKM4. We gathered from the same source the traffic speed and flow data from 19 junctions (J22 to J2) of the English M4 section from Bristol to London, Fig. 6a. The SRN's M4 motorway connects London to South Wales. Similarly to *UKM1*, junctions span 180 km and consist of different road lengths. Data collected goes from 2011 until 2015.

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

[2] https://data.gov.uk



(a) M1 and M4 highways.

(b) Interstate 5 highway.

Figure 6: Approximate location of the traffic sensors in (a) England and (b) California.

5. Experimentation: Imputation method

In this section, we evaluated our proposal as an imputation method by using a defined set of synthetically generated missing data while determining to what extent an improvement on the imputed values yields an enhanced accuracy of the subsequent traffic forecast model. Because imputation requirements may vary depending on the final application [23], we evaluated the final performance of the whole traffic forecasting system instead of measuring the distance between the real data and its reconstruction. There are cases in which improving the data imputation does not necessarily mean that prediction will improve, e.g., when there is sufficient information in the observed data for the traffic forecasting system to estimate. The reader may think on the increase in root mean squared error (RMSE) when the reconstruction is the same as the original but shifted by one value.

5.1. Inducing missingness

Although the original *PeMS* data contained missing values, we could not directly use those for evaluation as their values were previously imputed by PeMS [75]. Instead, we considered the PeMS data quality measure and produced artificial missing data on the test data, that we will refer to as *PeMS-NMAR*. All 5-min data samples available on PeMS are attached to a data quality measure. Those samples are the average of 30-sec samples in 5 minutes and the quality measure shows the percentage of valid samples during that time. In this section, we considered all 5-minute samples that did not meet a 75% quality measure as missing values. In other words, averaged values with more than 25% of invalid samples were treated as missing values. The data distribution of Fig. 7 resulted from said assumption in which 11.28% of the speed data is missing. Fig.7 shows a Not Missing at Random (NMAR) pattern where consecutive missing values are found in not so random time instants and sensors. This is consistent with real-world missingness types analyzed in literature [26, 29, 76].

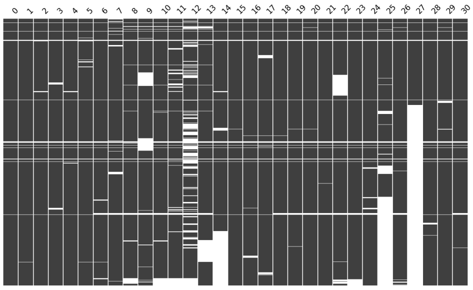


Figure 7: Distribution over the *PeMS* data set of the induced missing values (white fields). Each column shows two years of 5-minute samples of mean speed data corresponding to each detector.

Additionally, we investigated the robustness of the system against higher shares of missing values by removing additional observations from the data following a Missing Completely at Random (MCAR) pattern. There might be cases where the improvement in imputation accuracy is large but the improvements in estimation accuracy may not be significant. This might be because there is sufficient information in the observed data for the traffic forecasting system to estimate. Thus, the prediction accuracy was evaluated for 10%, 20% and 40%

missing data proportion on the test data which we will refer as *PeMS-MCAR-(%)*.

5.2. Evaluation task

Fig. 8 shows the scenario considered divided into two parts: an imputation layer that preprocesses corrupt speed traffic samples that are then fed separately to a regression layer to estimate future traffic speed. In interest of faster training, we set the RL 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 speed samples were used ($n = 36$) as input ($\boldsymbol{x} \in \mathbb{R}^{36\times31}$). Evaluation was done on all possible 3-hour speed traffic samples of *PeMS-NMAR* and *PeMS-MCAR* from 2016 (105360 samples) while the rest was used for training (105072 samples). Each experiment was conducted 10 times and we reported the mean of RMSE and mean absolute percentage error (MAPE) of the prediction task as the performance metrics in Table 1.

Imputation layer (IL). We compared our proposal (VAE) against a 1-hidden layer non-linear autoencoder (AE) and PCA. Details of VAE implementation are found in Sec. 3.3 and Fig. 8. We set $\boldsymbol{z} = \boldsymbol{\mu}_z$ for simplicity. The VAE model had 1,298,724 trainable parameters and the average time to convergence was 22 minutes using one GPU. Regarding the AE, *ReLU* was used for each layer except for the output. We trained both autoencoders with a batch size of 128 using a random validation split of 10% for earlystopping and 512 neurons per layer. The latent space dimension was first arbitrary set to 100. We used Adam [77] optimizer with a learning rate of $5e^{-5}$. Input was normalized to zero mean and unit variance and all missing values were treated as zero prior to each imputation method for fair comparison.

Regression layer (RL). A 2-hidden layer MLP was trained where each layer was composed of 100 neurons with sigmoids activations. However, note that it could be another type of model, not necessarily a DNN. $l2$ regularization was used on the weights to prevent overfitting. Input was normalized to zero mean and unit variance. The MSE was minimized using SGD with default Adam. We use a 10% random split of the training set as the validation set for early stopping of the training procedure. The RL showed better performance compared to a naive approach, where the last input sample is used as the estimation. On the original test data without missing values, RL showed a 34.8% and 25.3% improvement on RMSE and MAPE, respectively, which was considered as a benchmark and enough for the evaluation purpose.

5.3. Performance and discussion

The proposed VAE implementation showed an RMSE improvement of 69.6%, 52.6% and 39.5% over RL, PCA and AE on *NMAR* test speed data, 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 latent space. Results indicate that learning the $p_{\boldsymbol{\theta}}(\boldsymbol{X})$ 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 missing values 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 and the linear model can 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, which makes the proposed method more suitable for real-world data set where mostly NMAR patterns are found. We also varied the latent space dimension and provided some results on Table 1, where the compression factors applied on the data are shown between parenthesis near each IL method. Results showed that accuracy increased jointly with the compression factor but to a certain extent. Constraining the latent space dimensions forces the network to learn better features until the space becomes small enough. Same thing happened while increasing the dimensions. 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 optimal latent space dimension should be empirically defined as a

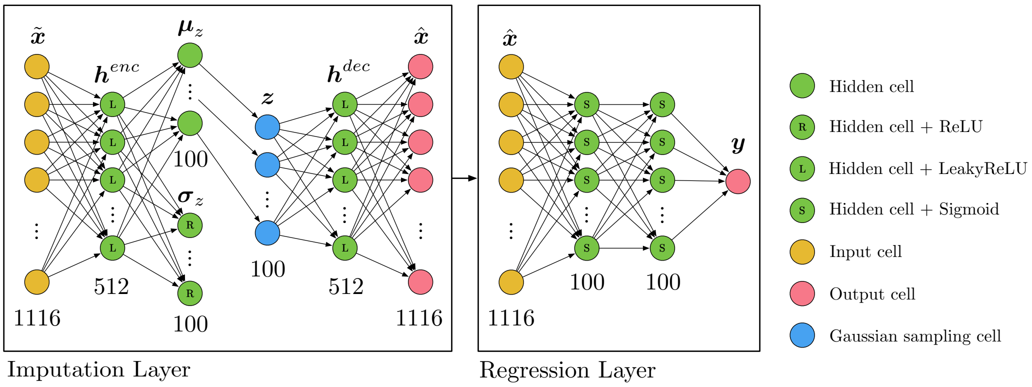


Figure 8: The production traffic forecasting system evaluated in Sec. 5. First, the imputation layer imputes the missing values of the 1116 dimension input. The VAE approach is showed following the imputation procedure of Sec. 3.5 using $\boldsymbol{z} = \boldsymbol{\mu}_z$. Then, an independent regression layer estimates the future traffic speed of one sensor using the reconstructed sample. Three different imputation layers based on VAE, AE and PCA were evaluated computing the RMSE and MAPE of the prediction task.

Table 1: $\overline{\text{RMSE}}$ $[km/h]$ — $\overline{\text{MAPE}}$ [%] results on the estimation of 1 hour ahead traffic speed of sensor number 15 over the *PeMS* test speed data. The first row shows the performance of the RL alone and should be compared to the results when an IL is added, i.e., IL + RL. The compression factor value is shown between parenthesis near each imputation method which was computed as the ratio between the dimensions of the data space and the latent space. MCAR–(%) indicates the proportion of generated missing data. In bold are the results closer to the performance of the RL on the *Original* data containing no missing values (the closer the better).

	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**

hyperparameter or by means of mutual information between layers like recently suggested in [69].

6. Experimentation: Dimension reduction

Under this section, we experimented with our proposal as a data compression tool. Our goal was to explore if the subspace learned by VAE results in representative and powerful features of the traffic data that can be used to perform traffic forecast. To that aim, we set a more complex problem and we aimed to estimate 1 hour ahead speed of all the network sensors using the last 12 hours of data of *PeMS* and the last 18 hours of *UKM1* and *UKM4*. Similarly to Sec. 5, we considered a feature extraction layer and a regression layer but, in this case, models were evaluated on *PeMS*, *UKM1* and *UKM4* test data. The latent space dimension was set to 100, thus models were forced to extract 100 features from a 4464 and 1368 input data space depending on the data set.

Feature extraction layer (FL). We compared our proposal (VAE) against an autoencoder (AE) and PCA similarly to Sec. 5. The implemented VAE is shown in Fig. 9 and described in Sec. 3.6. Training was done using KL cost annealing to avoid the *posterior collapse* problem. We set the initial weight of KL cost term to be zero and increased it at each training step k as $\beta_k = 0.0001 \times 1.05^{k-1}$, $\beta \in [0, 1]$. Gradients were clipped by their l_2 norm to 0.5 for training stability. The VAE models had 37,810,744/12,445,216 trainable parameters for the 4464/1368 input and the average time to convergence was 37/25 minutes using one GPU. Regarding the AE, the hidden layers were composed of 4096 neurons followed by *ReLU* activations. We trained both using Adam($5e^{-5}$) with a batch size of 128 and earlystopping. Input was normalized to

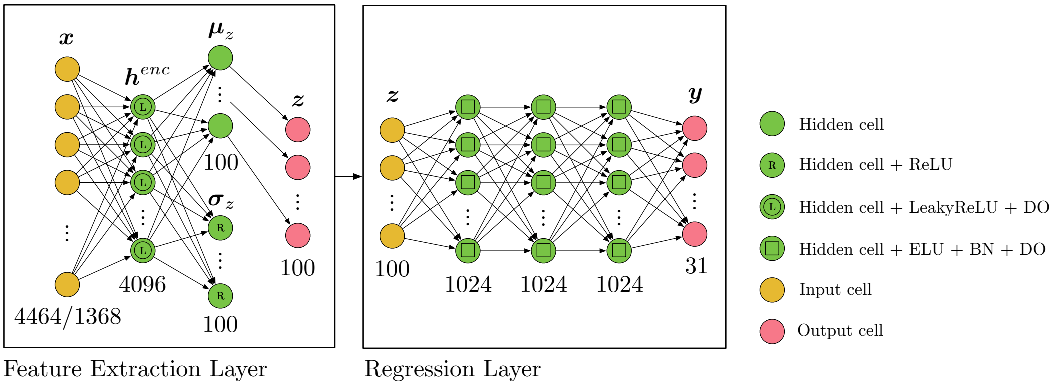


Figure 9: The production forecasting system evaluated in Sec. 6. Here, we show the encoder of an independently pre-trained VAE used as a feature extraction layer prior to long-term wide-network traffic forecast. Like in Sec. 5, we used directly $\boldsymbol{\mu}_z$ as the feature vector. The AE and PCA approaches were also evaluated as feature extraction layers. Note that the 4464 or 1368 input dimension is reduced down to 100 features to perform the forecast. DO stands for a Dropout layer with a drop probability of 0.5 and BN for a Batch Normalization layer.

zero mean and unit variance.

Regression layer (RL). We trained a non linear 3-hidden layer MLP with 1024 neurons per layer plus an Exponential Linear Units (*ELU*) activation function, a batch normalization (BN) layer and a dropout layer to avoid overfitting, Fig. 9. That is, 2,234,399 trainable parameters not counting BN layers. Labels were normalized to zero mean and unit variance. The MSE was minimized using SGD and Adam($1e^{-5}$) with a batch size of 128. The best generalization was selected as the final model using a validation split of 10%.

6.1. Performance and discussion

Main results are reported in Table 2. First two rows show that the tuned RL improved accuracy for all data sets. The RL performed the forecast from 4464 samples input for *PeMS* and 1368 for the other data sets, which equals to 12 and 18 hours of data respectively. The rest of the rows show the models evaluated that first projected the data to a 100 dimension subspace which was then used as input to train another RL with the same architecture. The data compression factor was 44,64 on *PeMS* and 13,68 on *UKM1* and *UKM4*. In Table 2, VAE outperforms all the compared models. Although the improvement is slight, below 5% on the RMSE, it even exceeds the performance of the original RL for all the data sets despite having significantly reduced the space dimension of the input. Therefore, the introduction of non-linearities and the latent variable model of VAE is well suited to extract useful features to perform traffic forecasting while at the same time for cloud computing and storage as significant compression factors are achieved.

Our proposal is intended to be a tool, independent from the model used in the prediction part. However, care must be taken in choosing the forecasting model because compressed data can degrade the performance of models that exploit the spatial or temporal structure of the data. Recall that we let $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$, i.e., latent components are orthogonal. Therefore, it is safe to conclude that using a convolutional or recurrent approach to forecast using the compressed data will not lead to improvement over MLP since the latent representations of the data do not keep the temporal or spatial structure. In that sense, we made some testing with LSTM and CNN prediction models that confirmed a degradation on the accuracy performance.

During training, we found that increasing the number of hidden layers derived in VAE ignoring most of the latent space. Instead, using one layer with a higher number of neurons led to learning better features for the traffic forecast task. In that sense, the KL cost annealing and the dropout layer also proved to be useful. The former helped to avoid the posterior collapse problem and the later to prevent overfitting of the model. Here, we spent the same time tuning each FL independently of the RL and results were considered enough to validate the VAE as a prominent solution for dimension reduction of traffic data. However, we believe that there exists room for improvement on optimizing the VAE model for this specific data set which is beyond the scope of this manuscript, e.g., using hyperparameter optimization techniques.

Table 2: $\overline{\text{RMSE}}$ $[km/h]$ values of 10 experiment runs of the RL forecast are reported showing the dimension reduction results on the test data of *PeMS* (4464 space dimension), *UKM1* and *UKM4* (1368 space dimension) data sets.

Model	RL input	*PeMS*	*UKM1*	*UKM4*
Naive	$\mathbb{R}^{4464;\,1368}$	10.83	13.06	15.24
RL	$\mathbb{R}^{4464;\,1368}$	7.51	11.11	10.73
PCA + RL	$\mathbb{R}^{100}$	7.61	11.38	10.56
AE + RL	$\mathbb{R}^{100}$	7.63	11.28	10.37
VAE + RL	$\mathbb{R}^{100}$	**7.49**	**10.89**	**10.23**

7. Experimentation: Model selection and anomaly detection

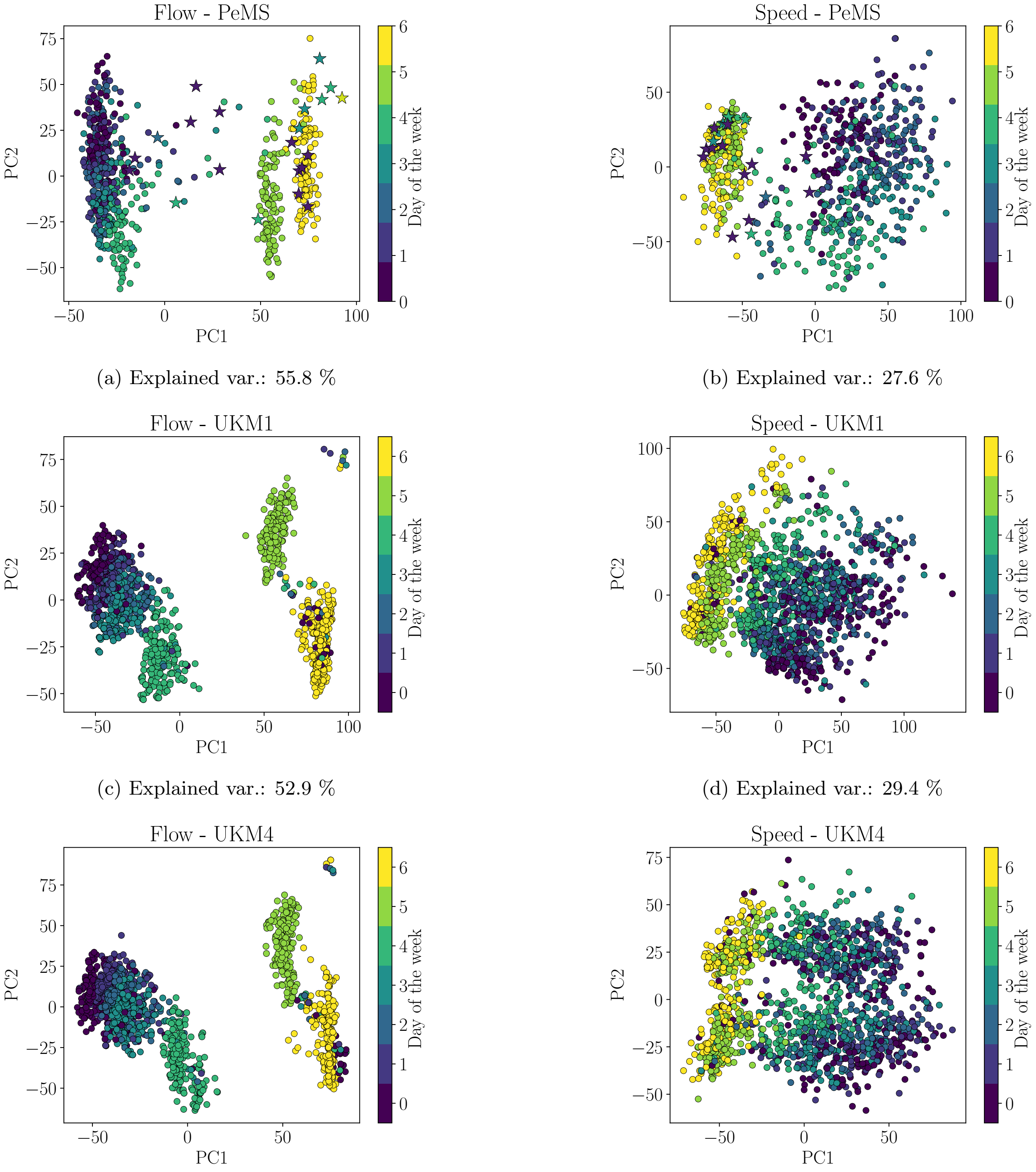
In this section, we experimented with the representational power of the VAE model and its learned latent space. To that aim, we trained the same model of Sec. 6, but using unique day samples of traffic flow and speed for all of the three data sets. Then, we projected the data to the learned subspace and analyzed it from the point of view of traffic modelers with the goal to improve the prediction accuracy of a traffic forecasting system. Note that the model can be used as an unsupervised tool to learn insights about traffic data without previous knowledge of the road traffic network.

7.1. Model selection

Fig. 10 shows the two principal components (PC) of the latent space. The pattern of flow and speed differs between weekends and weekdays, even a separate cluster for Fridays can be clearly distinguished from the flow. The flow is classified similarly for the three data sets, instead, the model classifies speed differently for *PeMS* rather than for the rest of data sets. Only in *UKM1*, the model can cluster between speed samples from Saturday and Sunday as speed has more complex behavior than flow. In *PeMS*, the weekend cluster is more separated from the weekdays cluster suggesting a greater difference between both and the possibility that two specific models for each cluster perform better than a global one. In *UKM4*, the model also clearly identified two clusters which are distinguished by different instants of time, Fig. 11. A similar trend can be slightly appreciated for *UKM1*. The data from 2012 and 2013 are classified in the upper cluster, while the data for the years 2011 and 2014 are classified in the lower one. Those differences at the time of fitting the forecast model can influence its performance since the 2012 data may not be beneficial for predicting 2014 traffic, as pointed out by the model.

Evaluation details. We divided the data sets into weekends and weekdays. For each data set a speed forecasting model was fitted to see if the overall performance of the two separate models was an improvement over a single model for the whole week. To that aim, we set the following problem: predict 1 hour ahead of all network sensors using the last 3 hours of data. We implemented the following DNN model: Dropout(0.5)–MLP(512)–LeakyReLU–MLP(512)–LeakyReLU, trained to minimize the MSE using l_2 regularization on the weights, Adam and normalized inputs. Three new data sets were made from whole week (*WW*), just weekdays (*WD*) and just weekends (*WE*) for *PeMS*, *UKM1* and *UKM4*. The whole last year of each data set was used for evaluation and the rest for training. We fitted the DNN model to each of the new data sets, resulting in three different models: *MLP–WW*, *MLP–WD* and *MLP–WE*, respectively. Finally, the RMSE performance of the *MLP–WW* model was used as a benchmark and we reported the RMSE improvement in % of the rest of the models with respect to it.

Results. The results of *MLP–WD* and *MLP–WE* models are shown in Table 3. From these results, it can be concluded that predicting the speed by using two separate models for weekdays and weekends in *UKM1* and *UKM4* shows little improvement over the results of the models for the whole week. On the other hand, in the case of *PeMS*, training a separated model only on weekend data improves the RMSE by 17.7% on weekend test data. However, no improvement on the *WD* data was found by the *MLP–WD* meaning that the performance resembles to the *MLP–WW* model. The latter model, which was trained on WW data, mainly learns how traffic behaves on weekdays because weekend samples are imbalanced w.r.t weekday samples. Those results are related to the cluster separation that exhibit the two classes in the latent space, which can be seen in the two-dimensional visualization of Fig. 10. More precisely, the euclidean distances in the latent space ($\mathbb{R}^{100}$) between weekday and weekend cluster centroids of *PeMS*, *UKM1* and *UKM4* are 87.3, 65.6 and 62.7, respectively. *PeMS*' clusters are the ones that the VAE model projected more separated



(a) Explained var.: 55.8 %

(b) Explained var.: 27.6 %

(c) Explained var.: 52.9 %

(d) Explained var.: 29.4 %

(e) Explained var.: 49.8 %

(f) Explained var.: 22.6 %

Figure 10: Traffic flow and speed samples projected to the latent space and colored by day of the week (0-6: Monday to Sunday). Day samples were projected to a 100-dimension latent space learned by the VAE model in an unsupervised manner. Then, the two principal components (PC) of the projected data were plotted with the help of PCA. The cumulative explained variance of PC1 and PC2 is shown below each figure, which means that the other dimensions that are not seen still capture more traffic characteristics. Only in *PeMS*, holidays samples are plotted with a star marker.

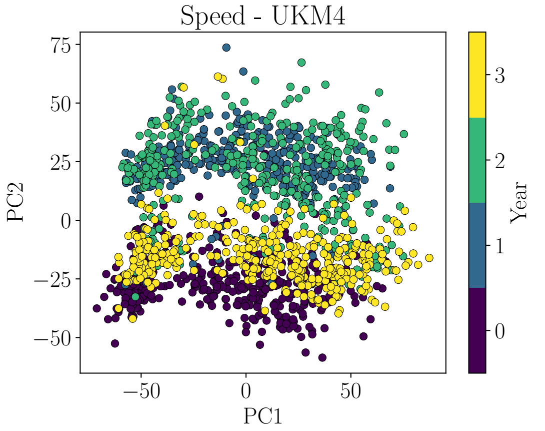


Figure 11: 2D PCA visualization of one-day UKM4 speed samples projected to the latent space ($\mathbb{R}^{100}$) and colored by year (0-3: 2011 to 2014). The clustering shows that traffic behavior changed between years.

Table 3: RMSE improvement [%] results of *MLP-WD* and *MLP-WE* models w.r.t. *MLP-WW* on the test data from all three data sets. The highlighted cells indicate the type of test data on which the model was intended to improve the performance.

Data set	Split type	*MLP–WD*	*MLP–WE*
PeMS	WW	-7.0	-45.0
	WD	0.1	-52.0
	WE	-26.0	17.7
UKM1	WW	-1.6	-17.6
	WD	1.0	-21.3
	WE	-7.1	2.2
UKM4	WW	0.5	-20.9
	WD	3.8	-26.0
	WE	-6.9	3.0

apart, that is, that were considered more dissimilar. This validates the latent space as an indicator of the performance of separated models for different classes of data. Therefore, the VAE model can be used by traffic modelers as a tool to decide when it is best to make use of different models instead of one unique model to predict traffic.

7.2. Anomaly detection

There are two interesting scenarios for anomaly detection in traffic forecasting: offline and online. Offline anomaly detection is helpful for traffic modelers to analyze historic data. Training a VAE model with unique day samples leads to Fig. 10-like images which can be used to detect anomalies. We

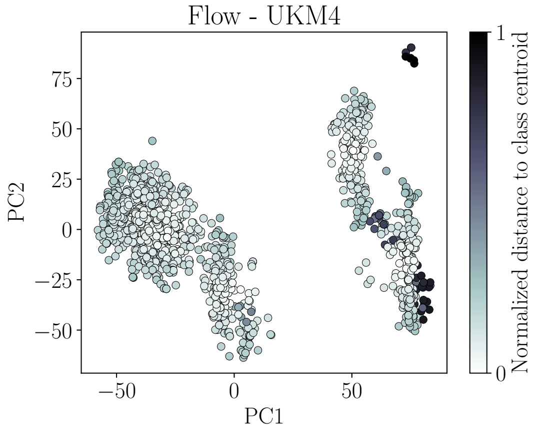


Figure 12: 2D PCA visualization of one-day UKM4 flow samples projected to the latent space ($\mathbb{R}^{100}$). Samples are colored proportionally to the distance (normalized between 0 and 1) to their corresponding cluster centroid, which highlights the anomalous traffic samples (darker). See Fig. 10e for comparison.

plotted the samples corresponding to holidays with a star marker on *PeMS* in Fig. 10a and 10b. Although holidays can be considered non-anomalies, it is more likely that during these the behavior of the traffic will deviate from the usual. First thing that Fig. 10a shows is that the majority of the holiday days behave like Sundays, which confirms a common and known fact of most road networks.

In Fig. 10a, a few of the samples are projected in the middle between the workday and weekend clusters, thus we inspected more closely those samples because we did not have anomalous traffic labeled. We visually compared the Monday and Sunday samples closer to their centroid against the holiday sample (Monday) placed between both clusters in Fig. 10a. This simplifies the analysis because the three data points compared vary greatly along the x-axis (PC1), while the variability in the y-axis (PC2) is much smaller. In this case, PC1 and PC2 represented the 55.8% of the variance of the 100 features learned by VAE. The anomaly is that the targeted sample does not behave like a Monday or Sunday which should be expected because the sample is labeled as a holiday Monday. To understand what caused the anomaly, we investigated the latent space by not varying PC2 and comparing the three mentioned traffic samples which produced a variation only on PC1. Upon investigation, we found an increase of traffic flow around sensor 9 for all three samples, which means that

PC2 is modeling where the traffic intensity is located in the road traffic network. Contrary, the main difference was the intensity of traffic flow and the peak hours. The intensity decreased proportionally from the Monday sample to Sunday while a light peak hour moved from morning to the afternoon, meaning that PC1 component is modeling those traffic features. Therefore, we can conclude that the anomaly was the intensity of the flow and when happened, not where it was located. We are not able to justify this behavior as we do not have the data labeled. However, this anomaly may be explained by the effect of non-traffic features (e.g., weather conditions, unusual events, etc.). That said, traffic modelers may consider the holiday sample as an anomaly and plan accordingly to absorb the specific traffic intensity at noon on that holiday. Furthermore, a traffic modeler could answer questions such as *what the holiday would have been like on a Wednesday?* by exploring the latent space, because the model learned meaningful dimensions plus it is a generative model with a continuous latent space.

On the other hand, one could elaborate more complex analytical methods for automatic anomaly detection, instead of inspecting it visually. For example, Fig. 12 shows the same samples of Fig. 10e, but colored proportionally to the Euclidean distance to their respective class centroids. A quick visual comparison shows that all holidays and anomalies are distinguished in darker color without previous knowledge or labeled data, validating the viability of the approach. In cases where the model can clearly cluster data (e.g., the flow from *UKM1;4*), a threshold should be defined by means of the AUROC for labeled anomaly data or by assuming that clusters are Gaussian distributed and setting the threshold proportional to the s.d., for unlabeled data. Similarly, in [53] dimensionality reduction is performed by PCA and then kNN outlier detection is applied. Nevertheless, note that same method applied to *UKM4*'s speed, Fig. 10f, would not lead to such clear results (at least in $2D$) due to the topology of the clusters and the higher complexity of speed data. Instead of using the Euclidean distance, a metric can be derived from (7) that provides an anomaly score function in terms of probabilities [56]. The intuition behind it is that anomalous data will have higher reconstruction errors because VAE is trained with far more normal samples than anomalous ones, that is, VAE learns to model normal traffic data. Finally, the best methodology to automatically detect anomalies in terms of implementation cost and accuracy will likely depend on the data under consideration and needs further exploration, hence, will be left as future work.

8. Conclusion

In this paper, we proposed a transversal solution for road traffic forecasting systems based on the assumption that traffic data can be generated from a manifold with reduced dimensions. We formulated the forecasting problem as a latent variable model and proposed the variational autoencoder (VAE) as a method to unsupervisely learn an approximation of the probability distribution of the traffic data. This was evaluated on three different real-world traffic data sets addressing some current major challenges of traffic forecasting and obtaining relevant results. First, the proposed model was used as an imputation method, showing significant improvements on unseen traffic samples with missing values. Secondly, we showed that the model can learn useful features for traffic forecasting systems, allowing for dimension reduction of traffic data without loss in accuracy on the forecast. In fact, a regression model trained with data compressed by a factor of 44,6 using the VAE exceeded the performance of the same model trained with the raw data. Third, we exploited the learned latent space from the point of view of traffic modelers in order to improve traffic forecasting systems. Without previous knowledge of the road traffic networks, projecting the data in said space allowed us to hypothesize and gain insights about the traffic data that were later validated with experimentation, concluding that our proposal can be used as a tool for model and data selection and anomaly detection.

Acknowledgment

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.

References

[1] I. Lana, J. Del Ser, M. Velez, E. I. Vlahogianni, Road traffic forecasting: Recent advances and new challenges, IEEE Intelligent Transportation Systems Magazine 10 (2) (2018) 93–109.

[2] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias, Short-term traffic forecasting: Where we are and where were going, Transportation Research Part C: Emerging Technologies 43 (2014) 3–19.

[3] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias, Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume, Transportation Research Part C: Emerging Technologies 14 (5) (2006) 351–367.

[4] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting.

[5] M. S. Ahmed, A. R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, no. 722, 1979.

[6] M. Lippi, M. Bertini, P. Frasconi, Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning, IEEE Transactions on Intelligent Transportation Systems 14 (2) (2013) 871–882.

[7] F. G. Habtemichael, M. Cetin, Short-term traffic flow rate forecasting based on identifying similar traffic patterns, Transportation Research Part C: Emerging Technologies 66 (2016) 61–78.

[8] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, S. M. Easa, Supervised weighting-online learning algorithm for short-term traffic flow prediction, IEEE Transactions on Intelligent Transportation Systems 14 (4) (2013) 1700–1707.

[9] R. García-Ródenas, M. L. López-García, M. T. Sánchez-Rico, An approach to dynamical classification of daily traffic patterns, Computer-Aided Civil and Infrastructure Engineering 32 (3) (2017) 191–212.

[10] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, et al., Traffic flow prediction with big data: A deep learning approach., IEEE Trans. Intelligent Transportation Systems 16 (2) (2015) 865–873.

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

[12] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction., in: AAAI, 2017, pp. 1655–1661.

[13] N. Laptev, J. Yosinski, L. E. Li, S. Smyl, Time-series extreme event forecasting with neural networks at uber, in: International Conference on Machine Learning, no. 34, 2017, pp. 1–5.

[14] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.

[15] Z. Zhao, W. Chen, X. Wu, P. C. Chen, J. Liu, Lstm network: a deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems 11 (2) (2017) 68–75.

[16] Q. Liu, B. Wang, Y. Zhu, Short-term traffic speed forecasting based on attention convolutional neural network for arterials, Computer-Aided Civil and Infrastructure Engineering 33 (11) (2018) 999–1016.

[17] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, Y. Liu, Deep learning: A generic approach for extreme condition traffic forecasting, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 777–785.

[18] C. Zhang, J. Butepage, H. Kjellstrom, S. Mandt, Advances in variational inference, IEEE transactions on pattern analysis and machine intelligence.

[19] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.

[20] D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, arXiv preprint arXiv:1401.4082.

[21] T. Pamuła, Impact of data loss for prediction of traffic flow on an urban road using neural networks, IEEE Transactions on Intelligent Transportation Systems.

[22] Y. Li, Z. Li, L. Li, Missing traffic data: comparison of imputation methods, IET Intelligent Transport Systems 8 (1) (2014) 51–57.

[23] I. Laña, I. I. Olabarrieta, M. Vélez, J. Del Ser, On the imputation of missing data for road traffic forecasting: New insights and novel techniques, Transportation research part C: emerging technologies 90 (2018) 18–33.

[24] L. Li, J. Zhang, Y. Wang, B. Ran, Missing value imputation for traffic-related time series data based on a multi-view learning method, IEEE Transactions on Intelligent Transportation Systems.

[25] X. Chen, Z. He, L. Sun, A bayesian tensor decomposition approach for spatiotemporal traffic data imputation, Transportation Research Part C: Emerging Technologies 98 (2019) 73–84.

[26] R. J. Little, D. B. Rubin, Statistical analysis with missing data, Vol. 333, John Wiley & Sons, 2014.

[27] S. v. Buuren, K. Groothuis-Oudshoorn, mice: Multivariate imputation by chained equations in r, Journal of statistical software (2010) 1–68.

[28] H.-F. Yu, N. Rao, I. S. Dhillon, Temporal regularized matrix factorization for high-dimensional time series prediction, in: Advances in neural information processing systems, 2016, pp. 847–855.

[29] L. Gondara, K. Wang, Multiple imputation using deep denoising autoencoders, arXiv preprint arXiv:1705.02737.

[30] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, arXiv preprint arXiv:1511.06349.

[31] M. Jang, S. Seo, P. Kang, Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning, Information Sciences 490 (2019) 59–73.

[32] V. Fortuin, G. Rätsch, S. Mandt, Multivariate time series imputation with variational autoencoders, arXiv preprint arXiv:1907.04155.

[33] J. Yoon, J. Jordon, M. van der Schaar, Gain: Missing data imputation using generative adversarial nets, arXiv preprint arXiv:1806.02920.

[34] C. Shang, A. Palmer, J. Sun, K.-S. Chen, J. Lu, J. Bi, Vigan: Missing view imputation with generative adversarial networks, in: Big Data (Big Data), 2017 IEEE International Conference on, IEEE, 2017, pp. 766–775.

[35] I. Tolstikhin, O. Bousquet, S. Gelly, B. Schoelkopf, Wasserstein auto-encoders, arXiv preprint arXiv:1711.01558.

[36] A. Brock, T. Lim, J. M. Ritchie, N. Weston, Neural photo editing with introspective adversarial networks, arXiv preprint arXiv:1609.07093.

[37] B. Dai, Y. Wang, J. Aston, G. Hua, D. Wipf, Connections with robust pca and the role of emergent sparsity in variational autoencoder models, The Journal of Ma-

chine Learning Research 19 (1) (2018) 1573–1614.

[38] S. Yang, S. Qian, Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents, arXiv preprint arXiv:1901.06766.

[39] D. Pavlyuk, Feature selection and extraction in spatiotemporal traffic forecasting: a systematic literature review, European Transport Research Review 11 (1) (2019) 6.

[40] Q. Li, H. Jianming, Z. Yi, A flow volumes data compression approach for traffic network based on principal component analysis, in: 2007 IEEE Intelligent Transportation Systems Conference, IEEE, 2007, pp. 125–130.

[41] N. G. Polson, V. O. Sokolov, Deep learning for short-term traffic flow prediction, Transportation Research Part C: Emerging Technologies 79 (2017) 1–17.

[42] N. Mitrovic, M. T. Asif, J. Dauwels, P. Jaillet, Low-dimensional models for compressed sensing and prediction of large-scale traffic data, IEEE Transactions on Intelligent Transportation Systems 16 (5) (2015) 2949–2954.

[43] H.-F. Yang, T. S. Dillon, Y.-P. P. Chen, Optimized structure of the traffic flow forecasting model with a deep learning approach, IEEE transactions on neural networks and learning systems 28 (10) (2017) 2371–2381.

[44] G. San Martin, E. López Droguett, V. Meruane, M. das Chagas Moura, Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis, Structural Health Monitoring (2018) 1475921718788299.

[45] D. Wang, J. Gu, Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder, Genomics, proteomics & bioinformatics 16 (5) (2018) 320–331.

[46] Y. Wu, H. Tan, L. Qin, B. Ran, Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding, Transportation Research Part C: Emerging Technologies 90 (2018) 166–180.

[47] J. Van Lint, C. Van Hinsbergen, Short-term traffic and travel time prediction models, Artificial Intelligence Applications to Critical Transportation Issues 22 (1) (2012) 22–41.

[48] J. S. Angarita-Zapata, I. Triguero, A. D. Masegosa, A preliminary study on automatic algorithm selection for short-term traffic forecasting, in: International Symposium on Intelligent and Distributed Computing, Springer, 2018, pp. 204–214.

[49] E. I. Vlahogianni, Optimization of traffic forecasting: Intelligent surrogate modeling, Transportation Research Part C: Emerging Technologies 55 (2015) 14–23.

[50] M. Van Der Voort, M. Dougherty, S. Watson, Combining kohonen maps with arima time series models to forecast traffic flow, Transportation Research Part C: Emerging Technologies 4 (5) (1996) 307–318.

[51] Y. Djenouri, A. Belhadi, J. C.-W. Lin, D. Djenouri, A. Cano, A survey on urban traffic anomalies detection algorithms, IEEE Access.

[52] Y. Djenouri, A. Zimek, Outlier detection in urban traffic data, in: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, ACM, 2018, p. 3.

[53] T. T. Dang, H. Y. Ngan, W. Liu, Distance-based k-nearest neighbors outlier detection method in large-scale traffic data, in: 2015 IEEE International Conference on Digital Signal Processing (DSP), IEEE, 2015, pp. 507–510.

[54] Y. Kawachi, Y. Koizumi, N. Harada, Complementary set variational autoencoder for supervised anomaly detection, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 2366–2370.

[55] M. Sölch, J. Bayer, M. Ludersdorfer, P. van der Smagt, Variational inference for on-line anomaly detection in high-dimensional time series, arXiv preprint arXiv:1602.07109.

[56] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lecture on IE 2 (2015) 1–18.

[57] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, arXiv preprint arXiv:1512.09300.

[58] A. v. d. Oord, N. Kalchbrenner, K. Kavukcuoglu, Pixel recurrent neural networks, arXiv preprint arXiv:1601.06759.

[59] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al., Conditional image generation with pixelcnn decoders, in: Advances in Neural Information Processing Systems, 2016, pp. 4790–4798.

[60] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, P. Abbeel, Variational lossy autoencoder, arXiv preprint arXiv:1611.02731.

[61] W. Shang, K. Sohn, Y. Tian, Channel-recurrent autoencoding for image modeling, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1195–1204.

[62] D. Pavlyuk, Short-term traffic forecasting using multivariate autoregressive models, Procedia Engineering 178 (2017) 57–66.

[63] G. Ostrovski, W. Dabney, R. Munos, Autoregressive quantile networks for generative modeling, arXiv preprint arXiv:1806.05575.

[64] B. Dai, D. Wipf, Diagnosing and enhancing VAE models, in: International Conference on Learning Representations, 2019.
URL https://openreview.net/forum?id=B1e0X3C9tQ

[65] F. P. Casale, A. Dalca, L. Saglietti, J. Listgarten, N. Fusi, Gaussian process prior variational autoencoders, in: Advances in Neural Information Processing Systems, 2018, pp. 10369–10380.

[66] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, O. Winther, Ladder variational autoencoders, in: Advances in neural information processing systems, 2016, pp. 3738–3746.

[67] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, K. Murphy, Fixing a broken elbo, in: International Conference on Machine Learning, 2018, pp. 159–168.

[68] A. Razavi, A. v. d. Oord, B. Poole, O. Vinyals, Preventing posterior collapse with delta-vaes, arXiv preprint arXiv:1901.03416.

[69] S. Yu, J. C. Principe, Understanding autoencoders with information theoretic concepts, Neural Networks.

[70] B. Dai, D. Wipf, Diagnosing and enhancing vae models, arXiv preprint arXiv:1903.05789.

[71] S. Zhao, J. Song, S. Ermon, Infovae: Information maximizing variational autoencoders, arXiv preprint arXiv:1706.02262.

[72] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-vae: Learning basic visual concepts with a constrained variational framework, in: International Conference on Learning Representations, 2017.

[73] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-vae, arXiv preprint arXiv:1804.03599.

[74] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.

[75] C. Chen, J. Kwon, P. Varaiya, The quality of loop data and the health of californias freeway loop detectors, PeMS Development Group.

[76] J. Van Lint, S. Hoogendoorn, H. J. van Zuylen, Accurate freeway travel time prediction with state-space neural networks under missing data, Transportation Research Part C: Emerging Technologies 13 (5-6) (2005) 347–369.

[77] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.