Short-Term Traffic State Prediction from Latent Structures: Accuracy vs. Efficiency

Wan Li[1], Jingxing Wang[2], Rong Fan[2], Yiran Zhang[2], Qiangqiang Guo[2], Choudhury Siddique[2], Xuegang (Jeff) Ban [2*]

[1]Transportation Planning and Decision Science Group, Oak Ridge National Laboratory

[2]The intelligent Urban Transportation System (iUTS) lab, Department of Civil and Environmental Engineering, University of Washington

[*]banx@uw.edu

ABSTRACT

Recently, deep learning models have shown promising performances in many research areas, including traffic states prediction, due to their ability to model complex nonlinear relationships. However, deep learning models also have drawbacks that make them less preferable for certain short-term traffic prediction applications. For example, they require a large amount of data for model training, which is also computationally expensive. Moreover, deep learning models lack interpretability of the results. This paper develops a short-term traffic states forecasting algorithm based on partial least square (PLS) to help enhance real-time decision-making and build better insights into traffic data. The proposed model is capable of predicting short-term traffic states accurately and efficiently by capturing dominant spatiotemporal features and day-to-day variations from collinear and correlated traffic data. Three case studies are developed to demonstrate the proposed model in short-term traffic prediction applications.

Keywords: Short-term traffic state prediction, spatiotemporal dependencies, partial least square regression, latent structures, real-time applications.

1. INTRODUCTION

Short-term traffic states prediction aims at forecasting traffic conditions, e.g., travel times, traffic speeds, and traffic volumes on road segments for a certain future time period, typically less than an hour. Results from many short-term traffic state predictions are critical inputs to real-time traffic operations or decision making, e.g., optimizing traffic signal timing every 15 minutes based on the predicted traffic volumes or queue lengths. This is one of the fundamental objectives of intelligent transportation systems (ITS). Hence accurately and efficiently predicting future traffic is essential for both scientific investigations and real-world traffic applications. With increasingly available traffic data, e.g., loop detector data, camera video data, and GPS data, the underlying complicated spatiotemporal correlations among traffic flow can be better investigated and identified using various data analytics methods.

Extensive efforts have been made in the past to improve the accuracy and efficiency of traffic forecasting algorithms. Since traffic states are characterized by both spatial and temporal dependencies (e.g., nearby locations may have similar or correlated traffic states), those algorithms strived to explore the spatiotemporal characteristics of traffic flow. Initially, the temporal dependencies of traffic states were considered as the core components. Time-series models used to be widely applied for traffic forecasting problems. The most commonly used approach is the Autoregressive Integrated Moving Average (ARIMA) model. With AR (autoregressive), MA (moving average) as well as their combinations (ARMA), ARIMA is more versatile to represent diverse time series (Moreira-Matias et al., 2013) and outperforms other time series model by their high prediction accuracy. However, the weakness of traditional ARIMA became obvious when handling big data with missing values and high variance as well as considering spatial correlations of traffic states data (Lee et al., 2006; Safikhani et al., 2018).

In order to further improve the prediction accuracy, researchers discovered that nonlinear models can better capture the temporal correlations of traffic states than simple linear models. Nonlinear transformation was then conducted to map the prediction problem from the input space into a higher dimensional feature space; then a linear model was used in the feature space for prediction. This is the basic idea of Support Vector Regression (SVR) based prediction models. SVR has been widely used for short-term traffic state prediction, such as traffic speed prediction (Vanajakshi and Rilett, 2004; Wang and Shi, 2013), travel time prediction (Wu et al., 2004), and traffic flow prediction (Ding et al., 2002; Zhang and Xie, 2007; Yang et al., 2014). The limitations of SVR-based prediction models are also obvious. Identification of the input space dimension and kernel function can be time-consuming, which is usually solved by trial-and-error methods using the minimum Final Prediction Error (FPE) rules in most studies. SVM also explicits limited ability to handle big data or spatial correlations of traffic data.

Other prediction algorithms tried to explore the spatial dependencies of traffic flow. The k-nearest neighbors (k-NN) algorithms were often used for short-term traffic state prediction considering spatial features of traffic data. They are non-parametric methods commonly used for classification and regression. The k-NN based traffic forecasting studies include the short-term traffic flow prediction (Zhang et al., 2013), traffic speed prediction (Cai et al., 2016), and travel time prediction (Myung et al., 2011). Although k-NN methods were widely applied due to their simple training process, many researchers also pointed out that the search algorithm in k-NN and method of forecasting result integration should be further investigated to improve model efficiency (Cai et al., 2016).

To consider both the spatial and temporal (i.e., spatiotemporal) features of traffic flow, Bayesian network (BN) models have been applied and become increasingly popular in the transportation field for traffic flow forecasting. As a probabilistic prediction model, BN is able to describe model uncertainty and flexibility (Castillo et al., 2008). For instance, BN was confirmed to be capable of representing the stochastic nature of traffic congestion (Liu et al., 2014). It can also be regarded as extremely natural tools for reproducing the random spatiotemporal correlations of traffic flows in OD (origin-destination) pairs and link traffic flows (Castillo et al., 2008). However, the problem of how many causal nodes to be selected is sometimes difficult as there is always a trade-off between the complexity and accuracy of the model (Li et al., 2019).

Recently, deep learning models have shown promising performance in many research areas due to their ability to model complex nonlinear relationships. Many studies have developed and applied different types of deep learning models in short-term traffic forecasting problems, due to their ability to capture the spatiotemporal correlations of traffic flow. Convolutional Neural Network (CNN) is an image processing algorithm often applied in computer vision. Because CNN can capture the local dependencies of traffic data and is less sensitive to noise, it hs been applied in traffic prediction. Song et al. (2017) compared a CNN based traffic speed prediction model with two Multi-Layer Perceptron (MLP) models for traffic data collected in Seoul. Their CNN model achieved higher prediction accuracy than MLP models. Ma et al. (2017) developed a CNN based model to predict traffic speeds for large scale transportation networks. The proposed model was tested using probe data collected in Beijing. Compared with the benchmark models, their CNN model achieved higher accuracy for long-term traffic speed prediction and was more efficient for large-scale transportation networks. Wang et al. (2016) proposed an error-feedback recurrent CNN (eRCNN) to leverage the correlation among nearby segments for traffic speed prediction. The eRCNN learns from prediction errors, which makes it possible to maintain high prediction accuracy upon abrupt traffic events such as traffic accidents. Ke et al. (2019) developed a two-stream multi-channel CNN (TM-CNN) for traffic speed prediction for multi-lane freeways. The proposed network was able to learn the correlations between individual lanes, in the spatial-temporal dimensions, and between speeds and volumes.

Another commonly used deep learning model is the Long and Short-Term Memory (LSTM) network. LSTM extends the memory of RNN and is well suited to learn from time series data that have relatively long-time lags in between. Cui et al. (2018) proposed a deep stacked bidirectional and unidirectional LSTM (SBU-LSTM) neural network that considered forward and backward dependencies in time series data to predict network-wide traffic speed. Their model captured the spatial features and bidirectional temporal dependencies from history data with a bidirectional LSTM layer. Zhao et al. (2017) applied LSTM to predict short-term traffic volumes on freeways. They first designed an origin-destination correlation (ODC) matrix composed of temporal-spatial correlations of different links in the network. They then integrated the ODC matrix with LSTM via fully connected layers and vector generators. The proposed method was tested on the traffic volume data within the fifth ring of Beijing. Liu et al. (2017) proposed a singular point probability LSTM (SPP-LSTM) model by discarding the LSTM network unit temporarily according to the singular point probability during the training process. Then the SPP-LSTM was integrated with ARIMA to predict the long-term traffic flow. The SPP-LSTM-ARIMA model achieved high accuracy and stability on the traffic flow data collected from the Vancouver-Richmond region in Canada.

Although CNN and LSTM models are powerful tools to capture the spatiotemporal dependencies of traffic flow, the limitations become obvious when the study area is very large. In such cases, deep graph neural networks (GNNs) show the advantages of extracting features from graph-structured data, e.g, social media networks, financial networks, and transportation networks. Cui et al. (2018) proposed a graph convolutional recurrent neural network for network-wide traffic forecasting. They built a traffic graph based on the physical network topology and defined a graph convolution neural network to capture the spatial features. The Traffic Graph Convolution (TGC) was then used to build the LSTM recurrent neural network to forecast the traffic spatial-temporal speed. Apart from the TGC-LSTM, other graph-based methods like Spectral Graph Convolution (Li et al., 2017) and Localized Spectral Graph Convolution (Yu et al., 2017) were also proposed in recent years. The graph-based deep learning methods can capture the spatiotemporal features on networks, which often leads to relatively high prediction accuracy. We note here that traffic state prediction is an extensively studied field with many learning methods developed and applied in the past; an extensive review of this body of literature is out the scope of this paper. The reader can refer to Nguyen (2018) and Shi & Yeung (2018) for comprehensive reviews of this field.

In practical applications, deep learning models can often help achieve accurate predictions and empower more informed decision making. Therefore they are becoming the most advanced and popularly used methods in the field of traffic states prediction. However, deep learning models also have drawbacks that make them less preferable for certain short-term traffic prediction applications. *First*, deep learning models require a large amount of data. If there are only limited observations, they are unlikely to outperform other approaches. For example, previous studies that successfully applied deep learning methods on traffic predictions utilized traffic data either for a long period of time, e.g., more than a year (Li et al., 2019; Cui et al., 2018; Ke et al., 2019; Wu et al., 2018) or cover a large area of road networks, e.g., a city-level network containing hundreds or thousands of links or regions (Yao, et al. 2018a; Yao et al., 2018b; Ma, et al., 2017; Wang et al., 2018; Lv et al, 2015). *Second*, deep learning models are computationally expensive to train. The models may take hours or even days or weeks to train using machines with expensive GPUs. For example. Li et al (2017) reported that their deep learning model for traffic flow prediction takes 5-10 min training time for each epoch. If the training process terminates after 50 epochs, the model will take 4-8 hours to train. Note that if the computations were repeated multiple times for hyperparameter optimization, it would have taken even longer time to train the models. *Third*, deep learning models lack interpretability of the results. They are developed largely based on empirical intuitions and trial-and-error. There is often little understanding of the properties of the models and little interpretation of model parameters with respect to the specific applications (Xu et al., 2018). This also makes it difficult to transfer the model and results from one application to other (even similar) applications.

In order to conduct accurate and efficient short-term traffic predictions, this study presents a local learning framework based on the partial least square (PLS) regression. PLS is a low-rank approximation method aiming at constructing a compact representation of the data with limited loss of information. There are many widely used low-rank approximation methods, see Kumar and Schneider (2016) for an extensive review. In the transportation research area, Principle Component Analysis (PCA) and Singular Value Decomposition (SVD) were commonly applied for traffic prediction and traffic data imputation (Chen et al, 2012; Qu et al, 2009; Verma et al., 2017). The assumption of PCA/SVD is that the traffic data lies in a latent linear subspace of a high dimensional space with "noise" and thus can be represented as a low-rank matrix. Dynamic Mode Decomposition (DMD) is a nonparametric decomposition method which can approximate a nonlinear dynamic system with a linear system effectively and efficiently (Kutz, et al., 2015). It differs from other dimensionality reduction methods such as PCA in the way that it can capture the temporal behaviors associated with a dynamic system. It was applied for exploration of traffic dynamics in the literature (Liu et al., 2016; Dicle et al., 2016). The aforementioned methods may not as effective as PLS for traffic state prediction. PCA/SVD is an unsupervised technique aiming at maximizing the variance of all the predictor variables such that information loss is largely reduced. DMD also cannot guarantee the approximated information is relevant and meaningful for the dependent variables in the prediction. PLS, one the other hand, treats one or more variables as dependent variables and include the relationships between predictor variables and dependent variables in its components calculation (Godoy et al., 2014). Therefore, PLS is selected in this study to compare with deep learning methods. We suspect that when using other low-rank approximation methods, similar conclusions may be drawn when comparing such methods with deep-learning methods.

To the best of the authors' knowledge, Coogan et al. (2017) was the first to apply PLS to traffic prediction, where long-term (4 hours) traffic volume prediction was conducted to help optimize traffic signal timing. In this paper, we apply the PLS method to short-term traffic prediction. The proposed model is capable of extracting dominant information from collinear and correlated traffic data. Specifically, the traffic state data containing spatiotemporal features are decomposed to low-rank representations using PLS to eliminate irrelevant and redundant information. A few underlying latent factors are extracted to account for most of the variations in the original data and predict traffic states in the near future. Three real-world traffic state prediction case studies are developed to demonstrate the effectiveness and efficiency of the proposed PLS model in short-term traffic prediction applications. Comparing with deep learning models, we show that PLS (i) only needs modest training data from local regions; (ii) is faster and more suitable for real-time prediction applications; (iii) can produce accurate results comparable with those from deep learning methods; and (iv) has better interpretability ability. Therefore, PLS can help achieve some balances between prediction accuracy and efficiency. We should note here that this paper is not arguing that the proposed local PLS model is better than deep learning methods. Rather, we hope to point out that since all methods have their strengths and weaknesses, one should carefully evaluate the pros and cons of each method and select the one that fits the specific application scenario the best. The major contributions of this paper are summarized as follows:

(1) We developed a short-term traffic states prediction scheme based on PLS regression. The proposed model is able to produce accurate prediction results with reasonable computational time using modest training data from local regions.
(2) The proposed PLS-based traffic state prediction method is capable of extracting and learning dominant spatiotemporal features and day-to-day variations of the traffic flow by decomposing and denoising the original traffic data to low-rank representations with interpretable latent variables.
(3) We demonstrate the effectiveness and efficiency of the proposed PLS model using three real-world traffic state prediction case studies. The proposed method can produce comparable prediction results with deep learning models but require significantly less computational effort.

2. METHODOLOGY

The learning problem of spatiotemporal traffic forecasting in this paper is formalized using PLS. The problem definition and modeling structure are first described. Then we illustrate how to apply PLS to predict traffic states by accounting for the spatiotemporal dependencies of traffic data.

2.1 Traffic Forecasting Problem

The objective of traffic forecasting is to predict the future traffic states, e.g., traffic speed, travel time, or traffic flows given previously observed data from loop detectors or GPS equipped vehicles on the road network. We denote traffic states at a certain location m in day d as a vector $x_m^d \in \mathbb{R}^T$:

$$x_m^d = [x_m^d(1) \quad x_m^d(2) \quad \dots \quad x_m^d(T)]^T, \quad d = 1, \dots, D, m = 1, \dots, M \tag{1}$$

where m denotes the location that traffic states are measured and T denotes the total time intervals. The $[\cdot]^T$ denotes vector transpose.

For a certain day d, traffic states can be represented as a vector $x^d \in \mathbb{R}^{TM}$:

$$x^d = [x_1^d \ x_2^d \quad \dots \quad x_M^d]^T, \qquad d = 1, \dots, D \tag{2}$$

The aggregate Traffic States Matrix (TSM) is shown in Eq. (3) where each row represents traffic states at time $t = 1, \dots, T$ on all locations $m = 1, \dots, M$ on a certain day d. Each column represents the traffic states at a certain time t on a certain location m on different days $d = 1, \dots, D$. TSM represents the historical traffic data as (one of) the input to PLS for traffic prediction.

$$X_D^{T,M} = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \dots \\ (x^D)^T \end{bmatrix} \in \mathbb{R}^{D \times (TM)} \tag{3}$$

Denote T_p ($T_p < T$) the current time. The traffic states forecasting problem aims to learn a function $\mathrm{F}(\cdot)$ that maps historical traffic states (i.e., TSM) and the states before T_p ($T_p < T$) on day $\bar{d}$ to future states after T_p on day $\bar{d}$ accounting for day-to-day variations and within-day correlations of traffic states learned from previous D days.

$$F\left(X_D^{T,M}, X_{\bar{d}}^{T_p,M}\right) = X_{\bar{d}}^{T-T_p,M} \tag{4}$$

2.2 PLS method

PLS is a generalization of multiple linear regression (MLR). It is robust to datasets with collinear, correlated and noisy issues, and can simultaneously estimate multiple response variables. PLS is a technique that first reduces the number of predictors to a set of uncorrelated (i.e., orthogonal) components and performs least squares regression on these components instead of on the raw data, thus reducing the problem dimension and computational efforts. It is commonly used when the number of predictor variables is more than the number of observations in the datasets and where those variables are highly correlated. In this subsection, we provide a brief description of the standard PLS method. In the next subsection, we present the special treatment and modifications to the standard PLS method for short-term traffic state prediction by exploring the specific structure of the problem.

Suppose we have a dataset with response variables Y, $Y \in \mathbb{R}^{n \times m}$ and a large number of predictor variables X, $X \in \mathbb{R}^{n \times p}$. It is noted that variable X and Y in PLS are column centered. The PLS method finds a few new variables (latent variables) T and U, which are the factor score matrix for X and Y respectively. The matrix P and Q are orthogonal loading matrices. E and F are the error terms. The general underlying model of PLS is:

$$X = TP + E \tag{5}$$

$$Y = UQ + F \tag{6}$$

The PLS algorithm is iteratively computed. It computes the first pair of components $p_1$and q_1 from the original variables X and Y. Then the second pair of components are calculate from the deflated matrix of X and Y, which removes the contributions of the first pair of components. In order to calculate the components, the PLS algorithm finds two set of weight vectors , denoted as w and c, such that the covariance between the score vector t and u is maximized. The score vector contains the projection of each observation (raw data) onto the directions w and c. The first pair of score vector is obtained by:

$$t_1 = Xw_1 \text{ and } u_1 = Yc_1 \tag{7}$$

The vector w_1 and c_1 are the first left and right singular vector from Singular Value Decomposition (SVD) of X^TY. The first pair of components (loading matrix) p_1 and q_1 can be computed as

$$p_1 = X^T t_1 \tag{8}$$

$$q_1 = Y^T t_1 \tag{9}$$

These first pair of components indicates how strong the variable X and Y are related to the the score t_1. Once the first pair of score vector is computed, the PLS algorithms proceeds by deflating the original data X and Y as $X = X - t_1p_1^T$ and $Y = Y - t_1q_1^T$. The vectors p_1 and q_1 are restored in the corresponding matrices. The PLS algorithm continues to calculate the following score vectors using the deflated matrix. For more details on the predictor components are calculated, readers can refer to Abdi (2010) and Rosipal & Kramer (2005) where overviews and tutorials of PLS algorithms are provided.

Fig. 1 illustrates the interpretations of the first pair of the components and the corresponding scores. The predicted and predictor components are the line of the best fit for the data points of X and Y respectively. They represent the directions that capture the maximum variance of the variables. The predictor (or predicted) components establish new coordinate systems with lower dimensions. The score of a data point is the coordinate of the data point in such new coordinate system (also called latent variables). More explanations are provided in Dunn (2019).

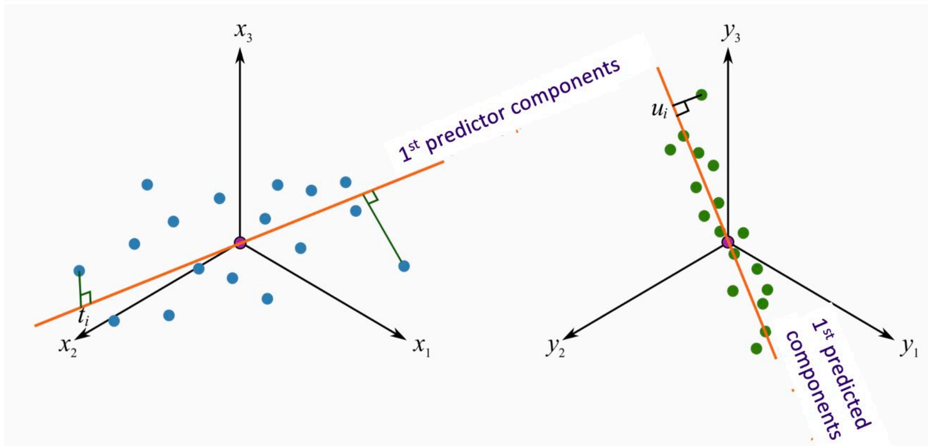


Fig. 1 A Geometric interpretation of PLS (Dunn, 2019)

2.3 Latent structures for spatiotemporal dependencies of traffic flow

The PLS regression model is capable of conducting short-term traffic state prediction based on historical data. It is noted that traffic states before T_p and after T_p are highly correlated due to the temporal dependencies of traffic flow. In practice, it is often more efficient to use the traffic states from the recent past of T_p (e.g., 2 hours) time sequence instead of all the estimates from the starting of the day. Furthermore, it is often more accurate if the prediction is done for the short term say a few minutes. In the following, we describe specific considerations of applying the PLS method to traffic prediction by considering the special features of the prediction problems.

In TSM (Eq. (3)), each row represents the traffic states for multiple time steps at multiple locations for a certain day. For a training data set containing multiple-day traffic states, there are usually large day-to-day variations. Thus, we need to find a low-rank approximation of the original data that is able to characterize such day-to-day variations and capture the spatiotemporal dependencies of traffic states. Specifically, we aim at finding a collection of predictor components $p^1, p^2, \dots p^N$ with each $p^i \in \mathbb{R}^{T_p M}$, predicted components $q^1, q^2, \dots, q^N$ with each $q^i \in \mathbb{R}^{(T-T_p)M}$, and a vector of score $Ts(d) \in \mathbb{R}^N$ in PLS with

$$Ts(d) = [t_1(d) \quad t_2(d) \quad \dots \quad t_N(d)]^T \tag{10}$$

such that the traffic states before T_p, x_b (predictor variables), and after T_p, x_a (response variables), can be express as:

$$x_b = \overline{x}_b + \textstyle\sum_{i=1}^{N} t_i(d)\, p^i \tag{11}$$

$$x_a = \overline{x}_a + \textstyle\sum_{i=1}^{N} t_i(d)\, q^i \tag{12}$$

where $\overline{x}_b$ and $\overline{x}_a$ are the average traffic states before and after T_p respectively, and N is the rank parameter ($N \geq 1$). The score matrix $Ts \in \mathbb{R}^{D \times N}$ generated from Eq. (10) are the transformed latent variables. They are the extracted *N* dominant traffic features (new variables after projection) for each day. In the score matrix, the dimension of traffic features for each day is reduced from $T_p M$ to *N*. The PLS predictor components $P \in \mathbb{R}^{N \times T_p M}$ are the *N* dominant latent variable directions. They represent all $T_p M$ traffic features for the exacted dominant *N* days. In the predictor component matrix, the dimension of the observations (total number of days) is reduced from *D* to *N*. Using Eq. (11-12), the mean centered measurements of traffic states are approximated as a linear combination of *N* predictor or predicted components where the number of components *N* is usually a lot smaller than the number of variables, i.e., the number of spatial and temporal features (TM). This enables the PLS method and its parameters better interpreted compared with deep learning methods. The components from PLS are ordered based on the descending order of the singular values of mean-centered measurements of traffic states $x_b - \overline{x}_b$ and $x_a - \overline{x}_a$. For example, the first pair of components (p^1, q^1) make the most significant contribution to capture the covariance of original predictor and response variables. More calculation details can be found in Coogan et al. (2017).

After the predictor and predicted components estimated from PLS using historical data, we can predict the future traffic states for a sample day after T_p using p^i and q^i. Here, it is assumed that latent variable directions for the historical data and sampled day data are similar so that they can share the same pairs of predicted and predict components. In order words, the spatiotemporal correlations of traffic data have similar daily periodic patterns. Denote $x_b^{\bar{d}} \in \mathbb{R}^{T_p M}$ the traffic states before T_p on the sample day $\bar{d}$. The scores in PLS are estimated as:

$$\widehat{Ts} = (\left(x_b^{\bar{d}} - \overline{x}_b\right)^T (P^T)^\dagger)^T \tag{13}$$

where P is the vector of predicted components and $(P^T)^\dagger$ is the Moore-Penrose pseudoinverse of P^T. Then the predicted ground truth value of travel time after T_p can be estimated:

$$x_a^{\bar{d}} = \widehat{Ts}^T Q^T + \overline{tt}_a, \;\; x_a^{\bar{d}} \in \mathbb{R}^{(T-T_p)M} \tag{14}$$

where Q^T is the vector of predicted components. P^T and Q^T are estimated from PLS.

The PLS prediction algorithm used in this paper is summarized below. It is noted that at each time step *t*, the optimal rank parameter *ncomp** is updated based on the minimum error rate.

Algorithm to predict traffic states using PLS regression

```
Set current time interval t (e.g., at 10:00 AM)
for time step between t to total time span TT in every Δt min:
    for # of PLS component ncomp from 1 to Ncomp (e.g.,40):
        for k from 1 to K:
            Run PLS regression;
            Estimate the validation error.
        end
        Estimate the average validation error for each ncomp
    end
    Find the optimal ncomp* with the minimum validation error.
    Predict traffic states using PLS regression with optimal ncomp*
end
```

3. NUMERICAL EXPERIMENTS

This section uses three real-world applications to demonstrate the effectiveness and efficiency of the proposed PLS method. We compare the PLS-based prediction method with other state-of-the-art methods from two aspects: accuracy and efficiency. The accuracy is measured by the Rooted Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE). In the literature, the efficiency of a prediction model is usually defined by energy/power consumption (joure) when implemented in software and model execution time (second) (Iqbal et al., 2019; Frincu et al., 2014).

A lower power consumption model usually has a shorter execution time. In this study, we use model training time as a metric to evaluate the efficiency of different models. All experiments were performed on the same commodity computer.

3.1 Short-term traffic speed prediction

Short-term traffic speed prediction plays a vital role in Intelligent Transportation Systems. The objective of this case study is to utilize the historical taxi trajectory data from Oct-Nov 2016 to predict the average speed in every five minutes at the Second Ring Road of Xi'an City, China. This case study was from the Transportation Forecasting Competition (see TRANSFOR 19) organized by the Standing Committee on Artificial Intelligence and Advanced Computing Applications (ABJ70) of Transportation Research Board, supported by IEEE ITSS Technical Activities Sub-Committee "Smart Cities and Smart Mobility" (SC&SM) and sponsored by Didi Chuxing Gaia Open Data Initiative (Gaia Open Dataset). The scope of the competition is to evaluate the prediction accuracy at morning and evening peak hours on a specific day (Dec 1, 2016).

The study area contains a corridor of 1 mile in length with two directions (Southbound and Northbound), as shown in Fig. 2. The first step for speed prediction is to estimate traffic speed from taxi trajectory data. The original datasets contain taxi location and order ID in every 2-4 seconds.

In order to estimate the average speed at time interval k, e.g., every 5 min for the 1-mile road segment, we followed three steps:

(i) Estimate the ground distance between two GPS points i and j, denoted as d_{ij} for GPS points in the 1-mile road segment (Jiménez-Meza, et al, 2013).

(ii) Estimate the instant speed of vehicle n at GPS point i using the time step at location i and j, t_i and t_j:

$$v_{i,n} = \frac{d_{ij}}{t_j - t_i}.\ \forall\, k-1 \leq t_j < t_j \leq k$$

(iii) Estimate the average speed of all vehicles at time interval k, N_k, e.g., every 5 min, on the 1-mile road segment using all GPS points in the segment:

$$\bar{v}_k = \frac{\sum_n^{N_k} \sum_i^I v_{i,n}}{N_k I}$$

It turns out that the average speed estimation itself does not significantly influence the prediction results. We have tested several widely used speed estimation methods given GPS traces (Fred, 1996) while the differences between the estimated speeds are trivial, usually less than 0.1 km/h (or <0.5%). However missing data exists in the trajectory data set and how to interpolate data when no GPS point was recoded at certain time intervals is more important. We tested several strategies to interpolate the average speed at certain intervals without GPS point. The following two rules works the best for this case study (i.e., they produce smaller prediction errors) when implemented:

(i) If there is no GPS point within time interval k: set $\bar{v}_k = \bar{v}_{k-1}$

(ii) If there is no GPS point within the first interval (k = 1) of a day: set $\bar{v}_1 = v_0$, where v_0 is the average speed of the first interval on the predicted day on Dec 1.

Traffic speed matrix $TS \in R^{D\times(T\times L)}$ is constructed as a predictor variable to PLS, where D = 23 weekdays, T = 72 time stamps on off-peak hours (12am-5:55am), and L=2 denoting two directions (Southbound and Northbound). The predicted day (Dec. 1) in the competition is Thursday.

Weekday and weekends have distinct traffic patterns as well as Monday and Fridays. This case study only used historical travel speed on Tuesday, Wednesday, and Thursday as inputs to PLS method in order to avoid redundant and bias information. The predicted variable in PLS, $Y \in R^{1\times(60\times2)}$, is the average traffic speed on morning peak hours from 6 am – 10:55 am on a certain day. The similar traffic speed matrix can be constructed for evening peak hours. Noted that we only use trajectory data of SB and NB directions within the prediction area in PLS although more data are given in the competition because traffic speeds within the study area (bounded area in Fig. 2) have higher spatiotemporal correlations and more relevant for the prediction in the same area.

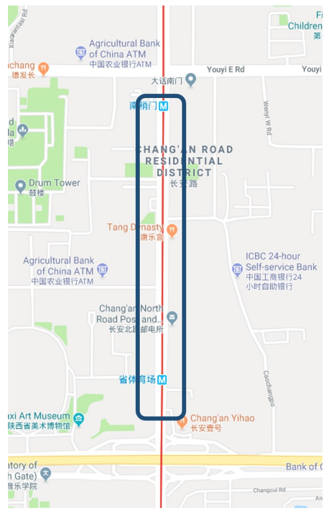


Fig. 2 The target area in Xi'an

Fig. 3 illustrates the accumulated percent variance of variable *X* explained by different number of predictor components for a randomly generated training data set. It indicates that 10 components account for more than 90% of variance. The optimal number of components used in PLS -based traffic speed prediction is determined by the cross-validation procedure.

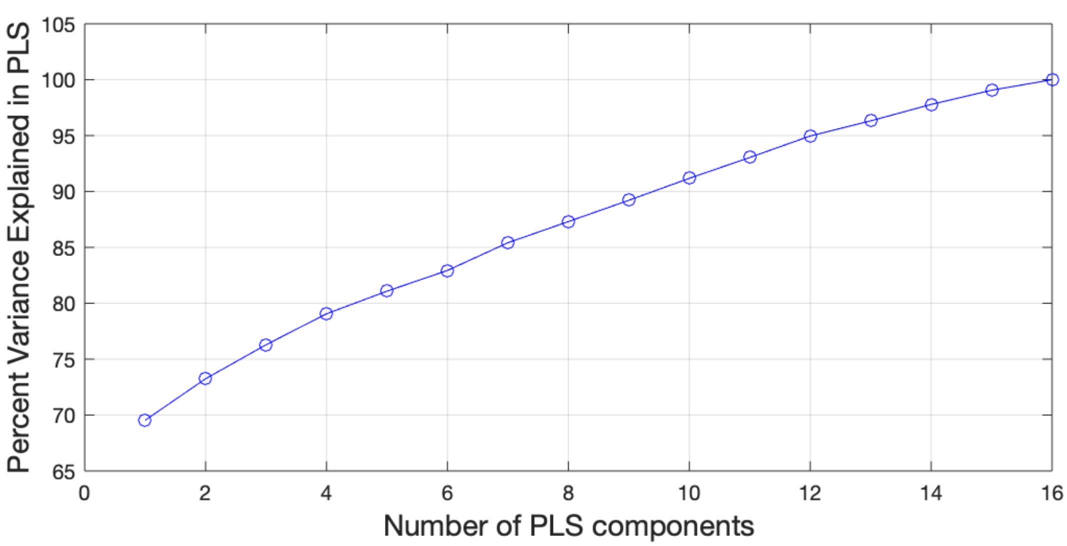
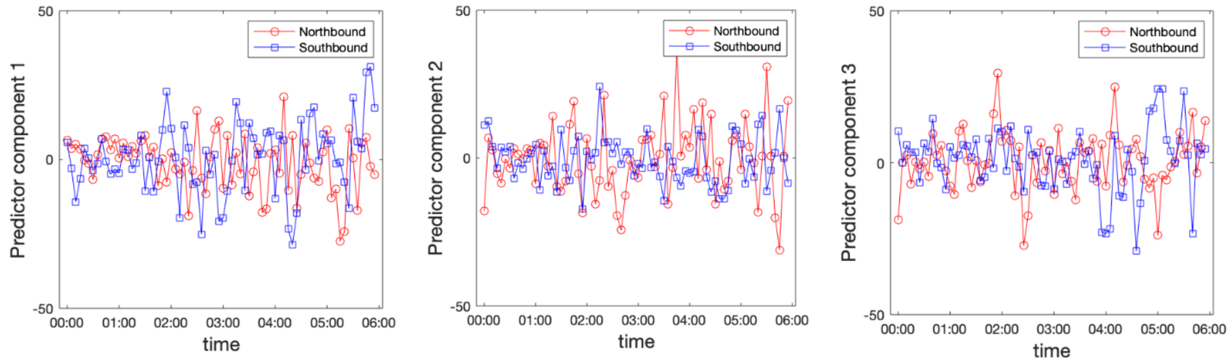


Fig. 3 Percent variance explained in PLS

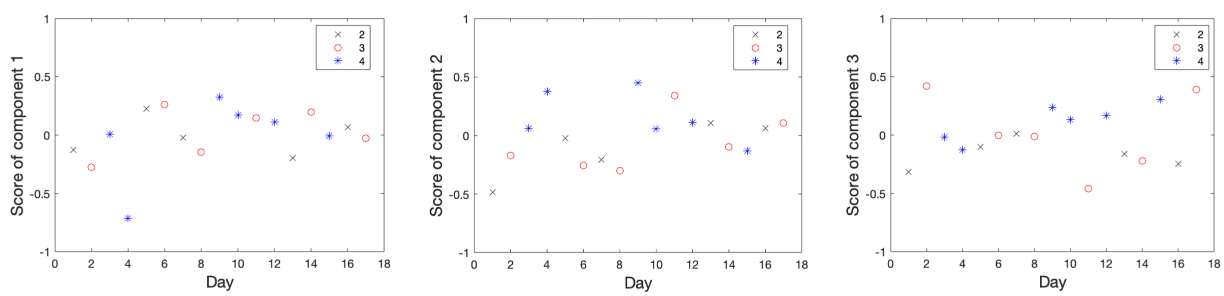
Fig. 4 shows the first three predictor components in PLS from 12am-5:55am (72 intervals) corresponding to the northbound and southbound directions. According to Fig. 3, they account for around 76% variance of predictor variable. Predictor components 1-3 are the first three dominant directions extracted from 17 days in training data. They represent the new observations (days) given all traffic features. Fig. 4 present high frequency of oscillations from 0am to 6am for both northbound and southbound. Because traffic speed data on weekdays were used in this study and they have similar daily patterns, so there is no obvious difference between these three components.



(a) Predictor component 1 (b) Predictor component 2 (c) Predictor component 3

Fig. 4 Predicted Components Matrix *P*

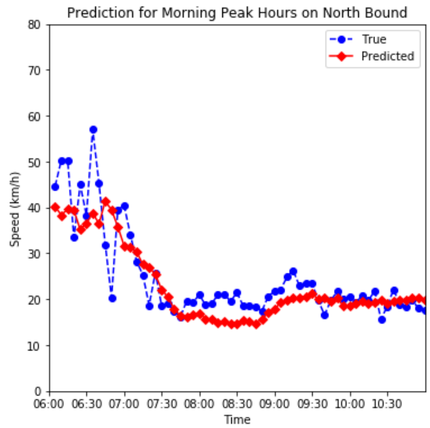
Fig. 5 shows the scores of the prediction components grouped by different days of the week (Tuesday is labelled as "2", Wednesday is labelled as "3", and Thursday is labelled as "4"). The score vectors can be considered as the new latent variables for each day (new traffic state features) projected onto the directions of predictor components. The scores in PLS can be used to identify the data pattern, outliers, and clusters. Because traffic speeds at midnight and early morning are relatively stable, the scores 1-3 do not show large difference.



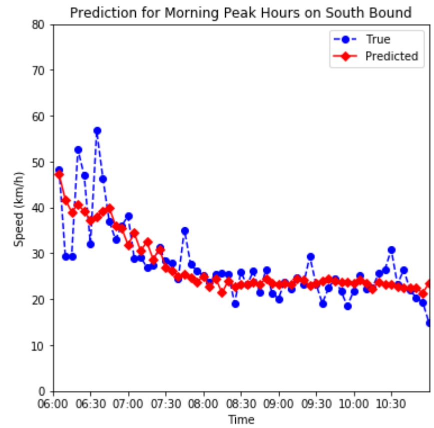
(a) Score 1 (b) Score 2 (c) Score 3

Fig. 5 Score vectors

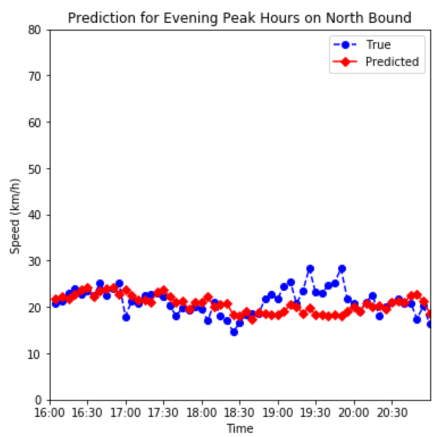
Fig. 6 shows the comparison between predicted traffic speed and the ground-truth traffic speed on Northbound (NB) and Southbound (SB) directions under morning peak and evening peak hours. The predicted speeds from PLS have smaller fluctuations and variations comparing to the real values. It is because we use the decomposed data instead of original raw data for prediction which have helped eliminate redundant and irrelevant information.



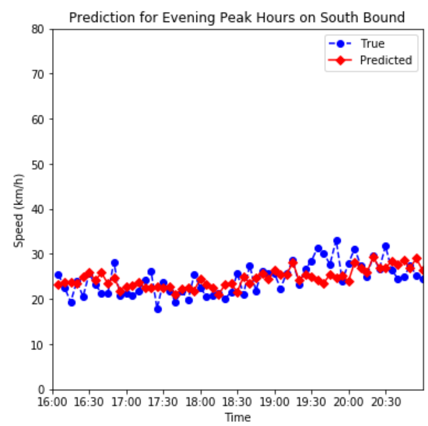
(a) Prediction results for Morning peak hours on NB direction



(b) Prediction results for Morning peak hours on SB direction



(c) Prediction results for Evening peak hours on NB direction



(d) Prediction results for Evening peak hours on SB direction

Fig. 6 PLS Prediction Results for Morning NB, Morning SB, Evening NB, and Evening SB

The proposed model is compared with the state-of-the-art methods. Table 1 shows the comparison results. The proposed PLS model outperforms other methods except for 2-layer CNN and 3-layer CNN model. The model improvements from PLS to ARIMA, k-NN, SVR, and DMD are relatively large while the difference between the deep learning methods (LSTM and CNN) and PLS is small. LSTM used in this case study only considers the temporal correlations of the speed data while CNN uses both spatial and temporal features by using the time-space traffic matrix as inputs to CNN (Ma et al., 2017). Although the CNN performance is slightly better than PLS, it requires much more model training time (> 8 min) compared with PLS (<2s) for this small prediction task. If implemented in real-world applications with a large amount of data, deep learning models will require more training efforts and parameter tuning. PLS is probably more preferable in this case because it is able to generate results with relatively high prediction accuracy (close to CNN) and much less computational effort.

Table 1 Model performance for traffic speed prediction

Method	Prediction Error		Model Training Time (s)
	RMSE	MAPE (%)	
ARIMA	5.72	18.48	< 2s
k-NN	5.98	19.65	< 2s
SVR	5.24	15.71	< 2s
DMD	5.28	15.78	< 2s
2-layer LSTM	5.02	14.58	~10min
3-layer LSTM	4.97	14.23	~15min
2-layer CNN	4.65	13.17	~8min
3-layer CNN	4.55	13.01	~10min
PLS	4.81	13.32	< 2s

3.2 Movement-based traffic volume prediction using sampled taxi data

At signalized intersections, traffic volume is considered as an important input for traffic signal control systems. Traffic volume data are often collected from loop detectors or cameras. However, not all intersections are equipped with such sensors/detectors. Moreover, detector failures are common issues due to pavement cracking or bad weather. In this case study, sampled vehicle trajectory data are applied to prediction turning volumes at intersections using PLS (detector data are used as "ground truth" for model training only). Trajectory data provide many features that can represent the traffic conditions for a certain road segment, e.g., the number of sampled vehicles/trajectories, number of stops, and the average speed on each movement, etc.. These features are considered as predictor variables in PLS. They are used to regress the ground-truth value of traffic volumes by accounting for day-to-day variations and spatiotemporal dependencies of traffic flow.

We test the proposed traffic volume prediction algorithm at an intersection in Jinan, China, as shown in Fig. 7. The loop data and GPS data cover the weekdays and weekends from 8/16/2018 to 9/7/2018 ($D = 23$).

Fig. 7 Intersection of Wenhuaxi Road and Luowen Road

The proposed model is tested on weekdays and weekends separately because of their different traffic patterns. Three features are extracted from trajectory data in every 15 min, including, the number of trajectories, the average number of stops, and the average speed, on each movement. In order to test the PLS-based traffic volume prediction algorithm, all features extracted from trajectory data for all movements from past four hours are used as predictor variables to calculate the ground-truth value of traffic volume at a current time interval. The proposed PLS-based prediction algorithm is robust to missing data problems because it can extract components from the low-rank structure that capture dominant trends of traffic volume, which are very significant for volume prediction from relatively low penetration of trajectory data. The input data for PLS have a large feature space (variables), e.g., 12 movement, 16-time intervals (4 hours), and 3 trajectory features (# of trajectories, # of stops and average speed). This spatiotemporal information contains duplicated information. Fig. 8 shows accumulated percent variance of trajectory data explained by different number of predictor components for a randomly generated training data set. It shows that five components can account for more than 90% of the variance. The optimal number of components used for traffic volume prediction is determined by the cross validation procedure.

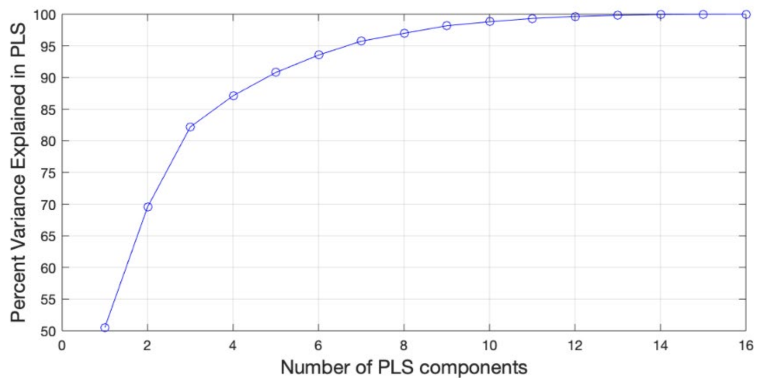
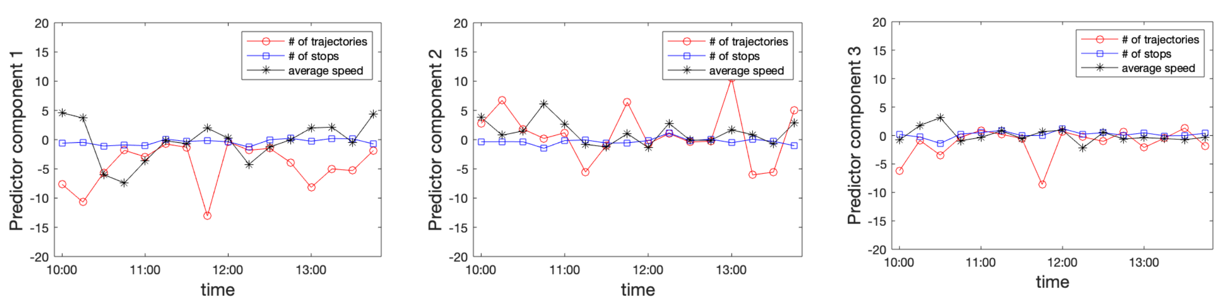
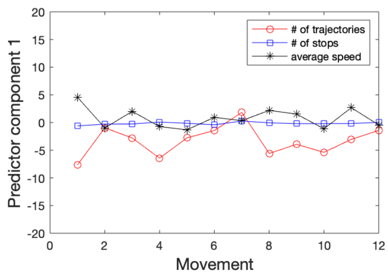


Fig. 8 Percent variance explained in PLS

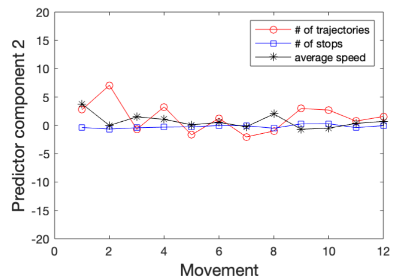
Fig. 9 shows the plot of predictor components 1-3 versus time intervals and movements. The predictor components, $p^i \in \mathbb{R}^{T_p MF}, i = 1,2,3$ with time intevals $T_p = 16$, movements $M = 12$, and traffic features $F = 3$ are generated from trajectory data. At each time *t* and location *m*, there are three traffic features available: # of trajectories, # of stops and average stops. The predictor components for # of stops (blue square line in Fig. 9) are all approximately zero in Fig. 9 (a-c) (and also in Fig. 9 (d-f)), which indicates that comparing to other two features (# of trajectories and average speed), # of stops is not a significant variable to identify the dominant latent variable directions. In other words, # of stops does not show large distinctions on different day of week. Hence, it has little contribution on traffic volume prediction in PLS.



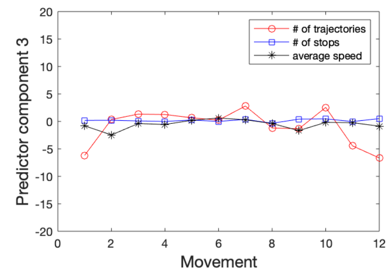
(a) PC 1 vs. time (b) PC 2 vs. time (c) PC 3 vs. time



(d) PC1 vs. movement



(e) PC2 vs. movement



(f) PC3 vs. movement

Fig. 9 Predictor components vs time and movements

Fig. 10 shows the scatter plot of scores for predictor component 1 and 2 grouped by weekdays (labelled as “0”) and weekend (labelled as “1”). The score vectors are the new latent variables (new traffic trajectory features) projected on the directions of predictor components for each day. They can be used to identify the day-to-day difference with respect to the new latent variables. It suggests that trajectory data on weekend generate lower score of predictor component 1 comparing to weekday. Two clusters can be clearly identified from Fig. 10. Thus, the scores of the predictor components captures the day-of-week variations of the raw trajecoty data.

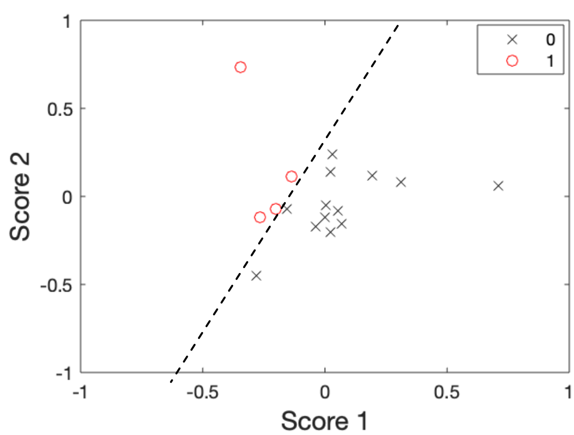
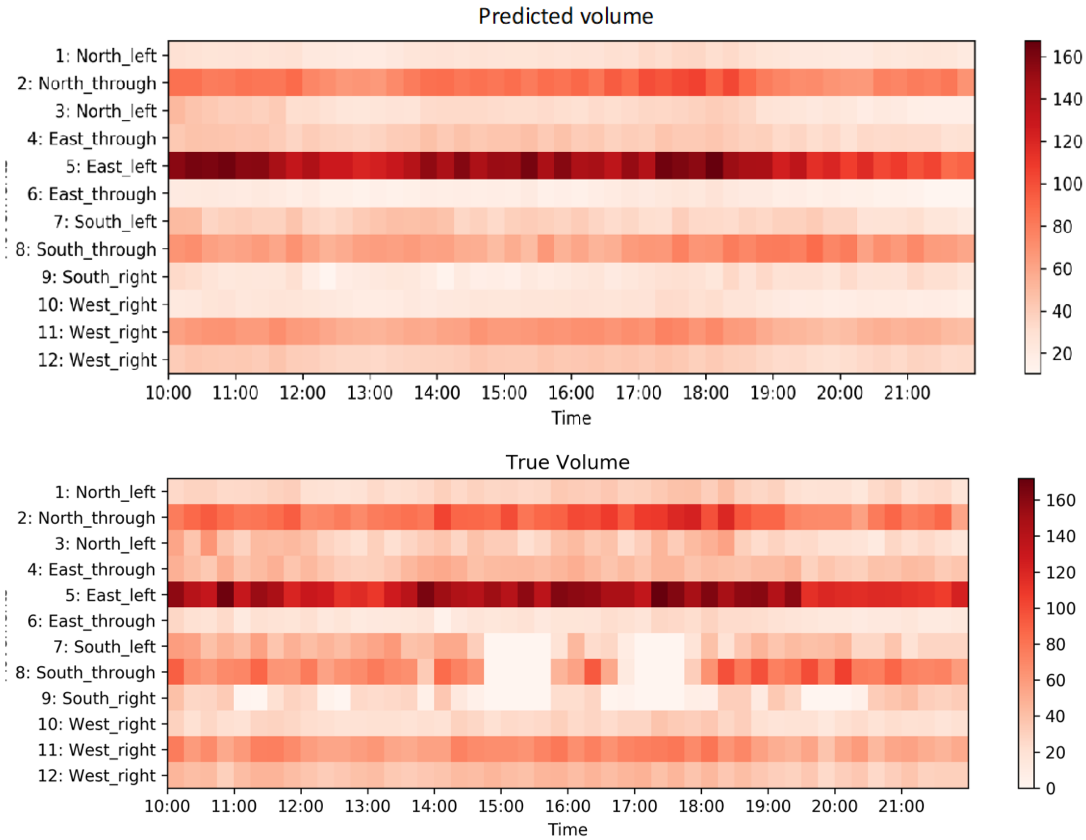


Fig. 10 Scores of Predictor component 1 and 2

Fig. 11 shows the heat map comparing the prediction results for different movements (vertical axis) and time-of-day (horizontal axis). In addition to the missing data issue in movements 7-9 in evening peak hours, the predicted results can capture the general trend of the ground-truth traffic volume by PLS method.



(a) Ground truth volume

Fig. 11 Volume prediction results for 12 movements of an intersection

Table 2 shows prediction results for an intersection and all movements on the weekend. Movement 7-9 has a low prediction accuracy due to missing data existing in loop data. The prediction results for through movement are better than turning movement for each approach.

Table 2 Volume prediction results for each movement on weekdays

Movement	**MAE** (vehicle/15min)	**RMSE** (vehicle/15min)	**MAPE** (%)
1: North left	3.08	3.98	11.57
2: North through	7.40	9.43	8.99
3: North right	6.42	8.20	19.13
4: East left	2.33	3.03	5.97
5: East through	7.16	9.12	5.21
6: East right	2.24	2.88	11.82
7: South left	8.04	11.32	24.16
8: South through	15.54	21.96	25.36
9: South right	8.13	10.92	36.96
10: West left	2.93	3.72	12.66
11: West through	5.30	6.54	8.52
12: West right	3.87	4.61	9.95

Table 3 compares the traffic volume prediction results by different methods. The proposed PLS method outperforms ARIMA, k-NN and SVR, and slightly worse than deep learning models, including LSTM and CNN. However, the model training efforts of PLS is significantly less than the deep learning methods.

Table 3 Model performance for traffic volume prediction

Method	**Prediction Error**		**Model Training Time (s)**
	RMSE	**MAPE (%)**	
ARIMA	10.87	21.69	< 2s
k-NN	9.91	19.75	< 2s
SVR	9.16	18.33	< 2s
DMD	10.32	20.54	< 2s
2-layer LSTM	8.31	16.61	~15min
3-layer LSTM	8.16	16.32	~20min
2-layer CNN	7.71	15.43	~5min
3-layer CNN	7.65	15.31	~10min
PLS	8.34	16.68	< 2s

3.3 Data fusion travel time prediction using GPS data and loop detector data

Reliable travel time information is valuable in urban transportation planning, operation, and management. This case study aims to combine multiple estimates of travel times (i.e., travel time estimation results) in a partial least square (PLS) based prediction algorithm to predict travel times. We test the proposed travel time prediction methods on a southbound ten-mile segment of Interstate 5 in Seattle, WA, as shown in Fig. 12. This segment passes downtown Seattle and experiences heavy congestion during peak hours. We have access to a 3-month historical data set, for both loop detector data and GPS data. The loop data for those days were collected from the DRIVE Net website (DRIVE NET). GPS data were provided by an industry partner. The loop data and GPS data cover the weekdays and weekends from 1/1/2012 to 3/31/2012 (D = 91) during morning peak hours from 7:00 AM to 10:30 AM.

Fig. 12 Test site in Seattle, WA

The objective is to predict travel time at *t+1* given the *H* intervals of historical travel time (up to time *t*) estimated from multiple measurements. We first compute travel times in every 5 min for the corridor with different methods using the two data sources, GPS data and loop detector data; see Fig. 13. The data contain information on segment-wise average speeds at each time interval. By using the data, we estimate both instantaneous and experienced travel times (Li et al., 2018). Fig. 13 present the historical travel time used in PLS for prediction at time *t+1*. There are four measurements of travel time at each time instant. In particular, we use the experienced travel time estimates calculated from GPS data as the "ground truth" travel times (predicted variable in PLS). All other estimates are considered as predictor variables. The PLS method is applied to predict ground-truth travel time given correlated travel time measurements in the past 2 hours ($H = 24$).

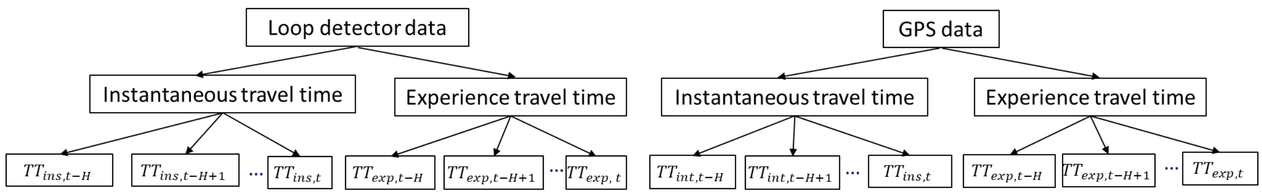


Fig. 13 Historical travel time used for prediction at *t+1*

Fig. 14 shows the average travel time distributions on weekdays and weekends for the study site (a 10-mile segment on I5 in the Seattle downtown area). On weekdays, the morning and evening peak hours can be identified in the blue dot line with longer travel times. On weekends, travel time around noon are usually higher. The day-of-week variations are expected to be captured by the PLS method.

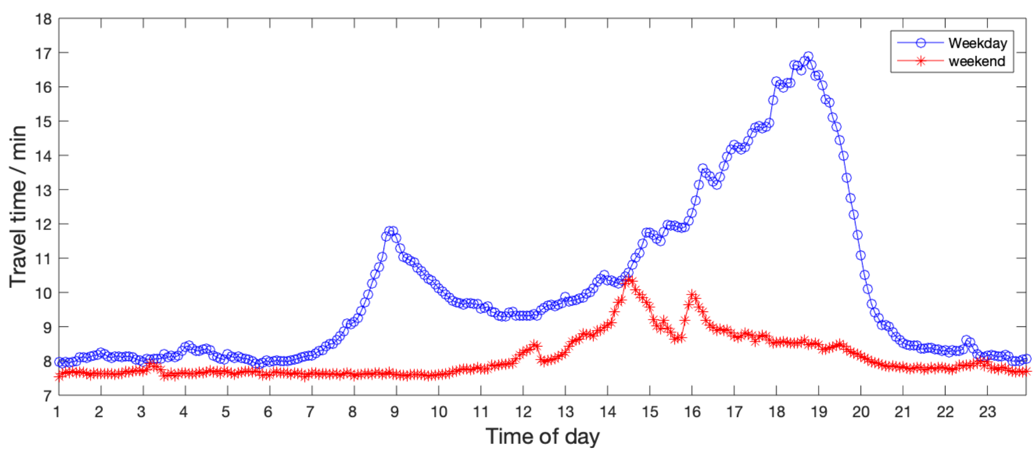


Fig. 14 Average travel time on weekdays and weekends

As aforementioned, the predictor components and predicted components in PLS are generated to explain the variance of the predictor and response variables. The input data to PLS for this travel time prediction task has the dimension of $\# \text{ of days} \times \text{features } (24 \text{ time intervals} * 4 \text{ measurements})$. Fig. 15 shows that 7 pairs of predictor components can explain more than 90% of the variance of historical travel time. The optimal number of components used for travel time prediction at each interval was determined by cross-validation.

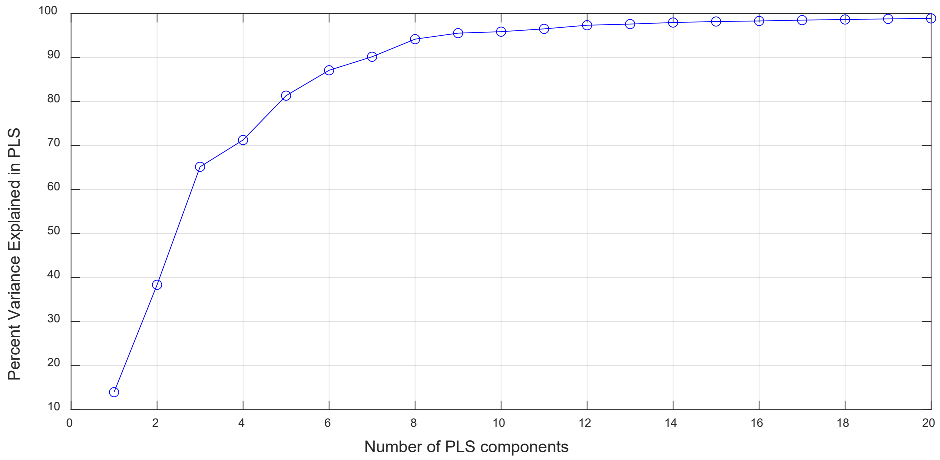
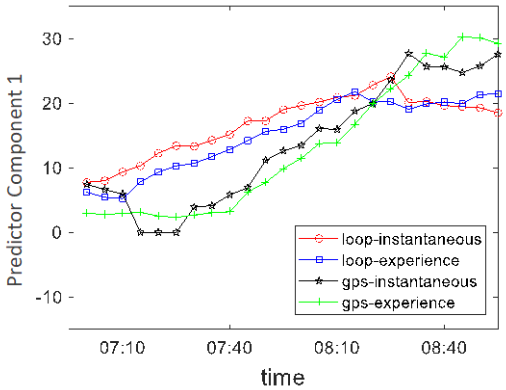
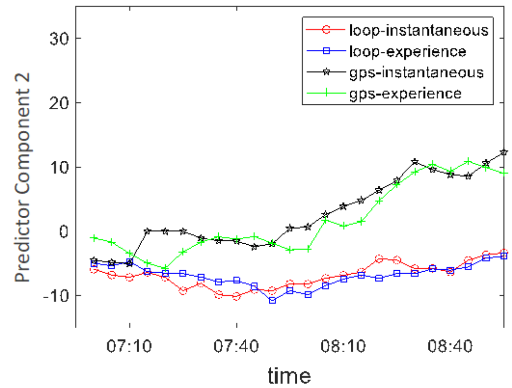


Fig. 15 Percent of variance explained by the PLS components

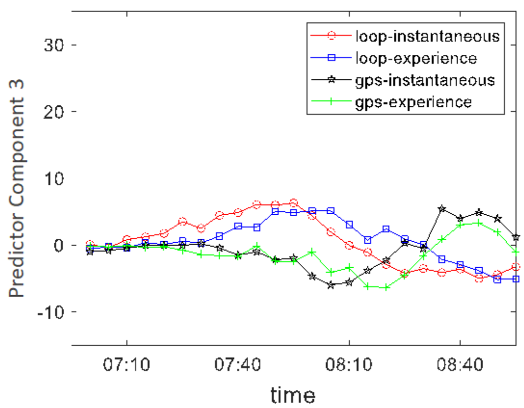
Fig. 16 shows the first four predictor components, $p^i \in \mathbb{R}^{T_p M}, i = 1,2,3,4$, for travel time estimated from 7:00 am to 8:55 am (24 time intervals), where the historical time intervals $T_p = 24$ and travel time measurements $M = 4$. It suggests that the first predictor component represent overall higher travel time values while its trend is consistent with the real travel time (increasing over time). Fig. 16 (b) shows for the predictor component 2. It suggests that the components estimated using loop detector data are smaller than those from GPS data, which is also consistent with the raw travel time data that travel time estimated from loop detector are overall underestimated comparing to GPS data. The predictor components 3 and 4 are overall smaller comparing to components 1 and 2 because they capture less variance of the predictor variable in PLS.



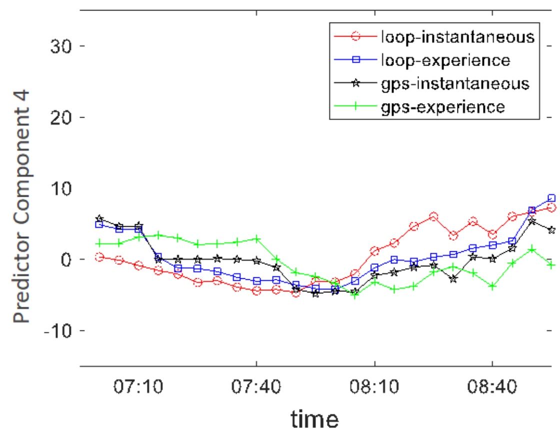
(a) Predictor component 1



(b) Predictor component 2



(c) Predictor component 3



(d) Predictor component 4

Fig. 16 Travel Timing Distribution on Weekdays and Weekends

Fig. 17 shows the scores of components 1 and 2 grouped by weekdays and weekends/holidays. It shows that the score of component 1 for weekend/holidays (red circles labelled as “1” in the figure) are smaller (and negative) comparing to weekdays (labelled as “0”) while scores of the component 2 for weekends/holidays are positive. This indicates that predictor component 1 mainly captures the time-of-day variations of travel time and the corresponding scores differentiate the travel time patterns between weekdays and weekend/holidays.

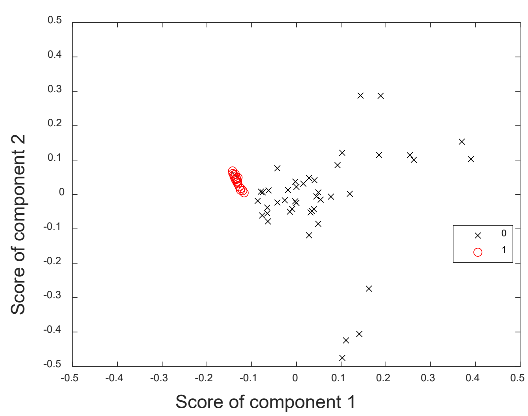
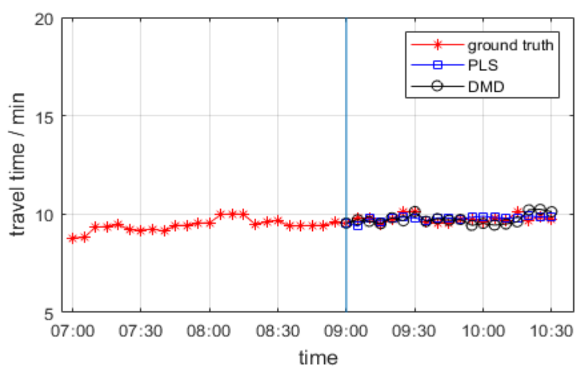
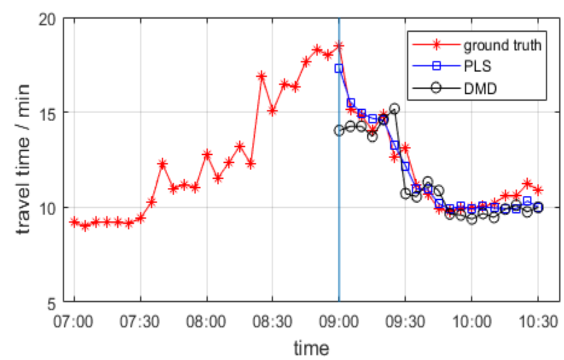


Fig. 17 Score of predictor components grouped by weekdays (label = 0) and weekends/holidays (label = 1)

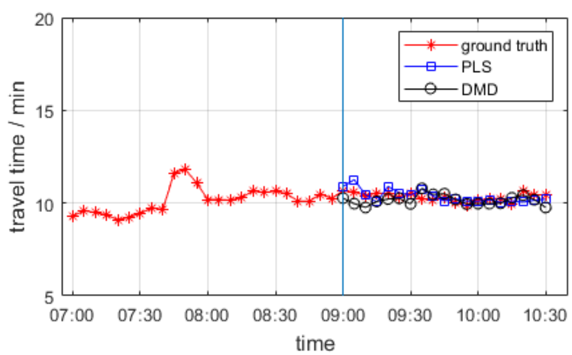
Fig. 18 shows the predicted travel time using PLS, another decomposition method DMD, and the ground truth value of travel. Fig. 18 (a)(c) present the travel time on weekends. They are stable comparing to weekdays in Fig. 18 (b)(d). In Fig. 18 (b), the travel times predicted from PLS and DMD are all close to ground truth values except that DMD method underestimates the travel time at the beginning of the prediction from 9:00 am to 9:20 am. As recorded by Washington Incident Tracking System (WITS), an incident occurred on 8:55am on 01/12/2019 at mile point 162.8 of southbound of I5 (WSDOT). It took 11 min to clear the incident. Fig. 18 (d) shows that PLS method can produce accurate predictions under such abnormal traffic conditions while DMD does not capture disturbance due to the accident at the beginning period of the prediction. The DMD is expert in extracting temporal dependencies of the data. It requires the predictor and predicted variables maintain the same dimension at each time interval in order to be applied for prediction. Hence, there is only one travel time measurement (experience travel time using GPS data) used as the predictor variables in DMD and some useful information from other measurements (as indicated in Fig. 13) are lost. More modeling details of the DMD refer to Li et al. (2018).



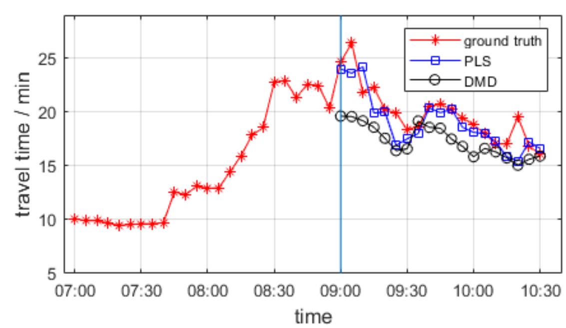
(a) 02/05/2012, Sun



(b) 01/18/2012, Wed



(c) 01/15/2012, Sun



(d) 01/12/2012, Thu

Fig. 18 Prediction Results

Table 4 shows that the proposed model generated comparable travel time prediction results to LSTM with significantly less model training time.

Table 4 Model performance for travel time prediction

Method	Prediction Error MAPE (%)	Model Training Time (s)
k-NN	6.3	< 1s
SVR	5.6	< 1s
DMD	6.7	< 1s
LSTM	4.3	22 min
PLS	4.8	< 1s

4. CONCLUSION

This study developed a PLS-based method in the applications of short-term spatiotemporal traffic state prediction. The PLS algorithm decomposes the original traffic state data and constructs components as a linear combination of the original predictors while considering the spatiotemporal correlations between the predictor and response variables. We conducted three real-world case studies to demonstrate the effectiveness and efficiency of the PLS model. The results showed that the PLS-based method has advantages over other regression methods, including the ability to handle the case of more predictor variables than the number of observations, robust to data sets with noisy and redundant information, and more stable and consistent when the sets of predictor variables are correlated rather than orthogonal. PLS also outperformed other state-of-the-art methods, including ARIMA, SVR, k-NN, and DMD in different traffic state prediction tasks. Compared with deep learning models, the proposed PLS method was able to produce comparably accurate predictions with much less computational efforts, and has better interpretability ability.

This study provides some insights on the accuracy and efficiency of different prediction models and concludes that the proposed PLS model can generate comparable prediction results as deep learning methods while largely reducing the training time and computational efforts, thus better balancing prediction accuracy and efficiency. This is practically important for real-world applications. First, traffic prediction models are different from image classification models or object detection models with fixed application scenarios (where deep learning model are widely applied). Traffic is a dynamic system and traffic demand profiles change over time. Traffic prediction models, such as those studied in this paper, should be trained periodically and kept up to date. Hence, although the time saving of PLS for one training process may not be very significant (e.g., for the case studies in the paper), considering the amount of training work repeatedly done in real-world applications, the model could be a competitive substitute of deep learning methods in terms of balancing prediction accuracy and reduced training time. Second, notice that the model training time of deep learning methods are 10-20 min but PLS is in seconds, representing about three order of magnitude improvement. The three case studies conducted in this paper focus on relatively small scale: such as a road segment or a few intersections. As the scale gets larger, e.g., a city-wide network, the training time can be much longer, and thus the reduction of training time will become more critical. For those large-scale applications, the proposed PLS model can also help balance the prediction accuracy and model training time.

This study showed the first step to conduct short-term traffic states prediction using PLS with some promising prediction results. For future work, we will continue to refine the model by collecting and testing on more data and larger networks. In addition, the robustness of the proposed PLS model will be further tested, e.g., under the conditions of missing data and abnormal traffic conditions occurred due to incidents or special events.

We also hope that this paper may point to an important issue for short-term traffic prediction applications (and probably other transportation applications as well), i.e., how to balance data needs, computational efforts, prediction accuracy, and interpretability of the results for the applications. As deep learning and other learning methods are currently all under rapid developments and improvements, while at the same time real-time decisions in transportation are becoming increasingly critical, we expect that the methods and results in this paper may motivate more discussions/investigations in this area, which will ultimately provide useful insights to important transportation applications and decisions.

5. REFERENCES

Abdi, H. (2010). Partial least squares regression and projection on latent structure regression (PLS Regression). *Wiley interdisciplinary reviews: computational statistics*, 2(1), 97-106.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, *62*, 21-34.

Castillo, E., Menéndez, J.M., Sánchez-Cambronero, S., 2008. Predicting traffic flow using Bayesian networks. *Transportation Research Part B: Methodological* 42, 482–509. https://doi.org/10.1016/j.trb.2007.10.003

Chen,C., Wang, Y., Li, L, Hu, J., and Zhang, Z. (2012). The retrieval of intra-day trend and its influence on traffic prediction. *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 103–118.

Coogan, S., Flores, C., & Varaiya, P. (2017). Traffic predictive control from low-rank structure. *Transportation Research Part B: Methodological*, *97*, 1-22.

Cui, Z., Ke, R., & Wang, Y. (2018). Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.

Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2018). High-order graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. arXiv preprint arXiv:1802.07007.

de Jong, S. (1993). SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, *18*(3), 251-263.

Dicle, C., Mansour, H., Tian, D., Benosman, M., & Vetro, A. (2016). Robust low rank dynamic mode decomposition for compressed domain crowd and traffic flow analysis. In *2016 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.

Ding, A., Zhao, X., & Jiao, L. (2002). Traffic flow time series prediction based on statistics learning theory. In *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems* (pp. 727-730). IEEE.

DRIVE Net. Digital Roadway Interactive Visualization and Evaluation Network. The University of Washington. http://uwdrive.net/STARLab.

Dunn, Kevin. (2019). Process Improvement Using Data. https://learnche.org/pid/latent-variable-modelling/principal-component-analysis/interpreting-score-plots-and-loading-plots#lvm-interpreting-scores

Frincu, M., Chelmis, C., Noor, M. U., & Prasanna, V. (2014, October). Accurate and efficient selection of the best consumption prediction method in smart grids. In *2014 IEEE International Conference on Big Data (Big Data)* (pp. 721-729). IEEE.

Gaia Open Dataset. https://outreach.didichuxing.com/research/opendata/en/

Godoy, J. L., Vega, J. R., & Marchetti, J. L. (2014). Relationships between PCA and PLS-regression. *Chemometrics and Intelligent Laboratory Systems*, *130*, 182-191.

Hall, F. L. (1996). Traffic stream characteristics. *Traffic Flow Theory. US Federal Highway Administration*, *36*.

Iqbal, M. F., Zahid, M., Habib, D., & John, L. K. (2019). Efficient Prediction of Network Traffic for Real-Time Applications. *Journal of Computer Networks and Communications*, *2019*.

Jiménez-Meza, A., Arámburo-Lizárraga, J., & de la Fuente, E. (2013). Framework for estimating travel time, distance, speed, and street segment level of service (los), based on GPS data. *Procedia Technology*, *7*, 61-70.

Ke, R., Li, W., Cui, Z., & Wang, Y. (2019). Two-Stream Multi-Channel Convolutional Neural Network (TM-CNN) for Multi-Lane Traffic Speed Prediction Considering Traffic Volume Impact. arXiv preprint arXiv:1903.01678.

Kishore Kumar, N., & Schneider, J. (2017). Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*, *65*(11), 2212-2244.

Kutz, J. N., Fu, X., Brunton, S. L., & Erichson, N. B. (2015). Multi-resolution dynamic mode decomposition for foreground/background separation and object tracking. In *Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on* (pp. 921-929). IEEE.

Lee, S., Lee, Y.-I., Cho, B., 2006. Short-term travel speed prediction models in car navigation systems. *Journal of Advanced Transportation 40*, 122–139. https://doi.org/10.1002/atr.5670400203

Li, W. & Ban, X. (2019). A Deep Learning Approach for Lane-Based Short-Term Traffic Volume Prediction at Signalized Intersections. In the 98th Annual Meeting of Transportation Research Board. No. 19-05692

Li, W., Siddique, C., Ban, X. 2018. Data-Driven Travel Time Prediction from Latent Structures using Multiple Data Sources. In the *7th International Workshop on Urban Computing, ACM KDD.*

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Li, Z., Jiang, S., Li, L., Li, Y., 2019. Building sparse models for traffic flow prediction: an empirical comparison between statistical heuristics and geometric heuristics for Bayesian network approaches. *Transportmetrica B: Transport Dynamics 7*, 107–

123. https://doi.org/10.1080/21680566.2017.1354737

Liu, B., Cheng, J., Cai, K., Shi, P., & Tang, X. (2017, October). Singular Point Probability Improve LSTM Network Performance for Long-term Traffic Flow Prediction. *In National Conference of Theoretical Computer Science* (pp. 328-340). Springer, Singapore.

Liu, C., Huang, B., Zhao, M., Sarkar, S., Vaidya, U., & Sharma, A. (2016). Data driven exploration of traffic network system dynamics using high resolution probe data. In *2016 IEEE 55th Conference on Decision and Control (CDC)* (pp. 7629-7634). IEEE.

Liu, Y., Feng, X., Wang, Q., Zhang, H., Wang, X., 2014. Prediction of Urban Road Congestion Using a Bayesian Network Approach. Procedia - Social and Behavioral Sciences, *The 9th International Conference on Traffic and Transportation Studies* (ICTTS 2014) 138, 671–678. https://doi.org/10.1016/j.sbspro.2014.07.259

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 865-873.

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

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L., 2013. Predicting Taxi–Passenger Demand Using Streaming Data. *IEEE Transactions on Intelligent Transportation Systems 14*, 1393–1402. https://doi.org/10.1109/TITS.2013.2262376

Myung, J., Kim, D. K., Kho, S. Y., & Park, C. H. (2011). Travel time prediction using k nearest neighbor method with combined data from vehicle detector system and automatic toll collection system. *Transportation Research Record*, *2256*(1), 51-59.

Nguyen, H., Kieu, L. M., Wen, T., & Cai, C. (2018). Deep learning methods in transportation domain: a review. *IET Intelligent Transport Systems*, *12*(9), 998-1004.

Qu, L., Li, L., Zhang, Y., & Hu, J. (2009). PPCA-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Transactions on intelligent transportation systems*, *10*(3), 512-522.

Rosipal, R., & Krämer, N. (2005, February). Overview and recent advances in partial least squares. In *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"* (pp. 34-51). Springer, Berlin, Heidelberg.

Safikhani, A., Kamga, C., Mudigonda, S., Faghih, S.S., Moghimi, B., 2018. Spatio-temporal modeling of yellow taxi demands in New York City using generalized STAR models. International Journal of Forecasting S0169207018301468. https://doi.org/10.1016/j.ijforecast.2018.10.001.

SC&SM. Smart Mobility and Transportation 5.0. IEEE Intelligent Transportation Systems Society. https://ta.itss-ieee.org/smart-mobility-and-transportation-5-0/

Shi, X., & Yeung, D. Y. (2018). Machine Learning for Spatiotemporal Sequence Forecasting: A Survey. *arXiv preprint arXiv:1808.06865*.

Song, C., Lee, H., Kang, C., Lee, W., Kim, Y. B., & Cha, S. W. (2017, June). Traffic speed prediction under weekday using convolutional neural networks concepts. *In 2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1293-1298). IEEE.

TRANSFOR 19. 2019. Transportation Forecasting Competition. https://github.com/TRANSFORABJ70/TRANSFOR19

Vanajakshi, L., & Rilett, L. R. (2004, June). A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed. In *IEEE Intelligent Vehicles Symposium, 2004* (pp. 194-199).

Verma, S., Narayanan, A., & Zhang, Z. L. (2017, September). Multi-low-rank approximation for traffic matrices. In *2017 29th International Teletraffic Congress (ITC 29)* (Vol. 1, pp. 72-80). IEEE.

Wang, D., Zhang, J., Cao, W., Li, J., & Zheng, Y. (2018). When will you arrive? estimating travel time based on deep neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, J., Gu, Q., Wu, J., Liu, G., & Xiong, Z. (2016). Traffic speed prediction and congestion source exploration: A deep learning method. *In 2016 IEEE 16th International Conference on Data Mining (ICDM)* (pp. 499-508). IEEE.

Wold, H. (1973). Nonlinear iterative partial least squares (NIPALS) modelling: some current developments. In Multivariate Analysis–III (pp. 383-407). Academic Press.

WSDOT. Overview of TIM Program. http://nchrptimpm.timnetwork.org/?page_id=1394. Accessed on Nov 10, 2019.

Wu, C. H., Ho, J. M., & Lee, D. T. (2004). Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, *5*(4), 276-281.

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

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How Powerful are Graph Neural Networks? arXiv preprint arXiv:1810.00826.

Yang, Z., Mei, D., Yang, Q., Zhou, H., & Li, X. (2014). Traffic flow prediction model for large-scale road network based on cloud computing. *Mathematical Problems in Engineering*, *2014*.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., ... & Li, Z. (2018a). Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., & Li, Z. (2018b). Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254*.

Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875.

Zhang, L., Liu, Q., Yang, W., Wei, N., & Dong, D. (2013). An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioral Sciences*, *96*, 653-662.

Zhang, Y., & Xie, Y. (2007). Forecasting of short-term freeway volume with v-support vector machines. *Transportation Research Record*, *2024*(1), 92-99.

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