Spatial and Temporal Patterns in Large-Scale Traffic Speed Prediction

Muhammad Tayyab Asif, *Member*, *IEEE*, Justin Dauwels, *Member*, *IEEE*, Chong Yang Goh, Ali Oran
Esmail Fathi, Muye Xu, Menoth Mohan Dhanya, Nikola Mitrovic and Patrick Jaillet

***Abstract*—The ability to accurately predict traffic speed in a large and heterogeneous road network has many useful applications, such as route guidance and congestion avoidance. In principle, data driven methods such as Support Vector Regression (SVR) can predict traffic with high accuracy, because traffic tends to exhibit regular patterns over time. However, in practice, the prediction performance can vary significantly across the network and during different time periods. Insight into those spatial and temporal trends can improve the performance of Intelligent Transportation Systems (ITS). Traditional prediction error measures such as Mean Absolute Percentage Error (MAPE) provide information about individual links in the network, but do not capture global trends. We propose unsupervised learning methods, such as k-means clustering, Principal Component Analysis (PCA), and Self Organizing Maps (SOM) to mine spatial and temporal performance trends at both network level and for individual links. We perform prediction for a large, interconnected road network, for multiple prediction horizons, with SVR based algorithm. We show the effectiveness of the proposed performance analysis methods by applying them to the prediction data of SVR.**

***Index Terms*—Large-scale network prediction, spatial and temporal error trends.**

I. Introduction

Intelligent Transport Systems (ITS) can provide enhanced performance by incorporating data related to future state of the road networks [1]. Traffic prediction is useful for many applications such as route guidance and congestion avoidance [1]–[3]. Traffic prediction requires learning non-linear relationships between past and future traffic states [2], [3]. Data driven methods such as Support Vector Regression (SVR) tend to provide better prediction results than competing methods [2]–[9]. However, these studies usually consider scenarios such as expressways, or a few intersections. Studies such as done in [2], [3], [10]–[12] consider highways or motorways only. Some other studies such as [6], [8] do consider more general scenarios, albeit for some small regions. Consequently, the performance patterns in large heterogeneous networks have not been investigated. Min et al. considered a moderate sized road network consisting of about 500 road segments [13]. However, they developed a custom model for the test area, which is not available. This limits any meaningful comparison with their proposed method. They analyzed the performance by taking into account different road categories. We will also consider road category wise comparison as one of the indices for performance evaluation. Traffic prediction for large networks requires modular and easily scalable algorithms. The methods should also provide accurate prediction for multiple prediction horizons. In this study, we analyze the performance of data driven methods such as SVR for large-scale prediction. The network comprises of roads with different speed limits, capacities and covering different areas (urban, rural, downtown).

Traffic prediction studies usually consider point estimation methods like Mean Absolute Percentage Error (MAPE) to analyze prediction performance [2]–[11], [13], [14]. These measures work well for overall performance comparison. However, they fail to provide any insight into underlying spatial and temporal prediction patterns. Forecasting methods may not provide uniform prediction performance across the road network. Moreover, prediction accuracy may also depend upon the time and the day. These spatial and temporal performance trends contain useful information about predictability of the network. ITS applications can provide more robust and accurate solutions by utilizing such trends.

We apply temporal window SVR to perform large-scale traffic prediction. For analysis, we consider a large road network from Outram Park to Changi in Singapore. The road network consists of more than 5000 road segments. We compare the performance of SVR with other commonly used time series prediction algorithms such as Artificial Neural Networks (ANN) and exponential smoothing. We also provide performance comparison for different road categories in this study. To extract spatial and temporal prediction patterns for large networks, we propose unsupervised learning methods such as k-means clustering and Principal Component Analysis (PCA). For link level temporal prediction patterns, we use Self Organizing Maps (SOM). We apply these data mining algorithms to extract performance trends in SVR prediction data.

The paper is structured as follows. In section II, we briefly discuss the problem of large-scale prediction. In section III, we propose different data driven algorithms for large-scale prediction. In Section IV, we explain the data set and performance measures for comparison. In section

This work has been presented in part at the IEEE Intelligent Transportation Systems Conference, Hilton, Anchorage, AK, USA, 2012.

Muhammad Tayyab Asif, Justin Dauwels, Esmail Fathi, Muye Xu, Menoth Mohan Dhanya and Nikola Mitrovic are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798; muhammad89@e.ntu.edu.sg, jdauwels@ntu.edu.sg.

Ali Oran is with Center for Future Urban Mobility, Singapore-MIT Alliance for Research and technology, Singapore, 117543; aoran@smart.mit.edu.

Patrick Jaillet and Chong Yang Goh are with Laboratory for Information and Decision Systems, MIT, Cambridge, MA, 02139; jaillet@mit.edu.

The research described in this project was funded in part by the Singapore National Research Foundation (NRF) through the Singapore MIT Alliance for Research and Technology (SMART) Center for Future Mobility (FM).

V, we compare prediction performance of SVR with other methods. In section VI, we develop unsupervised learning techniques for extracting spatial and temporal performance trends in large-scale prediction. In section VII, we evaluate the efficiency of proposed data mining methods with prediction data of SVR. In Section VIII, we summarize our contributions and suggest topics for future work.

II. Large-Scale Prediction

In this section, we briefly discuss the problem of large scale prediction. We also discuss the selection of training and test data for supervised learning algorithms.

A. Traffic Prediction

We represent the road network by a directed graph $G = (N,E)$. The set E contains p road segments (links) $\{s_i\}_{i=1}^{p}$. Space averaged speed value $z(s_i,t_j)$, represents the weight of the link s_i, during the interval $(t_j-\delta_t,t_j)$. The sampling interval δ_t is 5 minutes. Future traffic trends strongly depend on current and past behavior of that road and its neighbors [2], [3], [13]. Suppose $\{\theta_u \in \Theta_{s_i}\}_{u=1}^{l}$ is the set of road segments containing s_i and its neighbors, such that $\Theta_{s_i} \subseteq E$. Our aim will be to find the relationship function f between current/past traffic data $\{z(\theta_u,t_j-q\delta_t)|u=1,...l,q=0,...m_{\theta_u}\}$ and the future traffic variations $\hat{z}(s_i,t_j+k\delta_t)$ such that:

$$\hat{z}(s_i,t_j+k\delta_t) = f\big(z(\theta_1,t_j),...z(\theta_l,t_j-m_{\theta_l}\delta_t)\big). \qquad (1)$$

The feature set $\{m_{\theta_u}\}_{u=1}^{l}$ determines the horizon of the past speed values of link θ_u which are used for predicting k-step ahead speed values of s_i. We will refer to k-step ahead prediction as k^{th} prediction horizon.

We need to determine relevant links Θ_{s_i} (spatial features) and time lags m_{θ_u} (temporal features) to predict $\hat{z}(s_i,t_j+k\delta_t)$. Extracting spatial features is a computationally expensive task [15]. This additional computational cost severely limits the scalability of prediction algorithm for large and generic road networks. Therefore, we will not consider spatial features in this study. For large scale prediction, we consider the following variant of (1), termed as temporal window method [6]:

$$\hat{z}(s_i,t_j+k\delta_t) = f\big(z(s_i,t_j),...z(s_i,t_j-m_{s_i}\delta_t)\big). \qquad (2)$$

In (2), we only consider past historical trends of s_i to predict $\hat{z}(s_i,t_j+k\delta_t)$. Temporal window method for feature selection has been demonstrated to work effectively for data driven traffic prediction algorithms [3], [4], [6], [7], [9], [16]. Different methods have been proposed to take further advantage of inherent temporal traffic patterns for enhanced prediction accuracy [8], [10], [12], [17]–[21]. These methods employ different feature selection techniques to pre-partition the data according to temporal patterns (time of the day, weekdays/weekends etc.). The feature selection algorithms include Self Organizing Maps (SOM) [12], [18], [21], genetic algorithms [8], [19], wavelets [20] and committees of neural networks [10]. Another proposed method combines Kalman filter with Seasonal Autoregressive Integrated Moving Average (SARIMA) [12], [17]. These techniques, however, are computational expensive, which limits their scalability for large networks. Furthermore, M. Lippi et. al. showed that traditional SVR can provide similar performance to these ensemble methods without suffering from extra computational overheads [12]. Consequently, we will consider SVR with temporal window for large scale prediction. We train separate predictors for each link s_i and for each prediction horizon k. Temporal window method for feature selection allows predictors from different links and prediction horizons to run in parallel. Furthermore, these algorithms are independent of each other. Therefore, they can efficiently run on distributed platforms with minimum communication overhead.

B. Training and test data for supervised learning

Supervised learning methods such as SVR and Artificial Neural Networks (ANN) assume that the labeled training data and the test data come from the same distribution [22]–[24]. Hence, it is unnecessary to retrain the algorithm every time new data becomes available. Traffic prediction methods also follow the same assumption [2]–[9], [13], [16], [25]. Similar to other studies, we train the algorithm with 50 days of data and perform prediction for 10 days [3], [5], [13]. It is important to point out that this assumption may not hold true in the long term. Factors such as changes in transportation infrastructure, residential location, fuel prices and car ownership can significantly affect long term traffic patterns [25], [26]. Supervised learning methods may not work well in such cases. Techniques based on transfer learning might prove useful in such scenarios [24].

III. Traffic Prediction Algorithms

We apply SVR to perform large scale prediction. We briefly explain the algorithm in this section. We compare the performance of SVR with ANN and exponential smoothing. We also briefly discuss these algorithms in this section.

A. Support Vector Regression

SVR is a data driven prediction algorithm. It is commonly employed for time series prediction [27]. With temporal feature selection, the input feature vector $\mathbf{x}_j \in \mathbb{R}^n$ at time t_j for link s_i will be $\mathbf{x}_j = [z(s_i,t_j)...z(s_i,t_j-m_{s_i}\delta_t)]^T$. The feature vector $\mathbf{x}_j$ contains current average speed of the road $z(s_i,t_j)$ and m_{s_i} past speed values. Let $y_{jk} = z(s_i,t_j+k\delta_t)$ be the future speed value at time $t_j+k\delta_t$. We aim to find the relationship between y_{jk} and $\mathbf{x}_j$. To this end, we use historical speed data of s_i to train SVR. The training data contains r 2-tuples $\{(\mathbf{x}_j,y_{jk})\}_{j=1}^{r}$. We use SVR to infer non-linear relationships between $\mathbf{x}_j$ and y_{jk}, to find f_k in (2) for k^{th} prediction horizon.

We briefly explain the SVR algorithm here. More rigorous treatment of the topic can be found in [22], [23]. Let us consider the formulation of SVR called ε-SVR, which is formulated as [22]:

$$\begin{aligned} &\text{minimize} \;\; \frac{1}{2}\mathbf{w}\cdot\mathbf{w}+C\sum_{j=1}^{r}(\xi_j+\xi_j^*), \\ &\text{subject to} \;\; \begin{cases} y_{jk}-\mathbf{w}\cdot\mathbf{x}_j-b\leq\varepsilon+\xi_j \\ \mathbf{w}\cdot\mathbf{x}_j+b-y_{jk}\leq\varepsilon+\xi_j^* \\ \xi_j,\xi_j^*\geq 0, \end{cases} \end{aligned} \tag{3}$$

where, $\mathbf{w}$ is the required hyperplane and ξ_j, ξ_j^* are the slack variables. It uses so called insensitive loss function which imposes cost C on training points having deviation of more than $\mid \varepsilon \mid$. It is often hard to predefine the exact value of error bound ε [28]. This problem can be avoided by adopting a variant of SVR called ν-SVR [28]. Hence, we will employ ν-SVR to perform speed prediction. SVR non-linearly maps (not explicitly) the input speed data into some higher dimensional feature space Φ [22], [28]. It then finds the optimal hyperplane in that high dimensional feature space Φ. The kernel trick helps SVR to avoid this explicit mapping in Φ. Let us chose κ as the desired kernel function. Then we can replace dot products in the feature space by the relation $\kappa(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$ [22]. The function f_k will be [22], [28]:

$$f_k(\mathbf{x}) = \sum_{j=1}^{r}(\alpha_j-\alpha_j^*)\kappa(\mathbf{x},\mathbf{x}_j)+b, \tag{4}$$

where α_j, α_j^* are the Lagrange multipliers. We employ (4), to train SVR and perform speed prediction. The matlab package LIBSVM is used for SVR implementation [29].

For this study, we consider Radial Basis Function (RBF) kernel. It is highly effective in mapping non-linear relationships [30]. Consequently, it is commonly employed for performing traffic prediction [3], [4].

B. Artificial Neural Networks

Artificial Neural Networks (ANN) can perform time series prediction with high accuracy [31]. Consequently, they have been extensively used for traffic parameter forecasting in different configurations [2], [4], [6], [8], [10], [11], [16], [32]. Multi-layer feed forward neural networks is the most commonly employed configuration for traffic prediction [4], [16], [32].

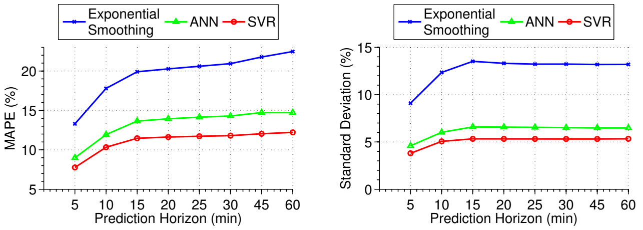
We apply feed forward neural network for large scale speed prediction of the network G across multiple prediction horizons. We consider temporal window for feature selection for ANN. We apply back-propagation to train ANN. We train separate neural networks for different links and prediction horizons.

C. Exponential Smoothing

Exponential smoothing is a commonly employed method to perform time series prediction. It is also applied for traffic parameter prediction [33]. The prediction is computed as a weighted average of past data points. Specifically, weights of past values decay exponentially with decay factor χ_k for k^{th} prediction horizon.

TABLE I: Categories of road segments

Category	CATA	CATB	CATC	Slip Roads	Other
No. of links	703	2818	841	592	70



(a) Mean error for different algorithms.

(b) Error standard deviation within the network.

Fig. 1: Performance comparison of prediction algorithms for different prediction horizons.

IV. Data Set and Performance Measures

In this section, we explain the data set for performance evaluation. We consider a large sub-network in Singapore, which covers region from Outram park to Changi. The road network G consists of a diverse set of roads having different lane count, speed limits and capacities. It includes three expressways, which are East Coast Parkway, Pan Island Expressway and Kallang-Paya Lebar Expressway. The network also includes areas carrying significant traffic volumes, such as Changi Airport and the central business district.

Overall, the network G consists of $p = 5024$ road segments. These road segments are grouped into different categories by Land Transport Authority (LTA) of Singapore. Table I shows the number of road segments for each category in G. In this study, we consider speed data provided by LTA. The data set has an averaging interval of 5 minutes. We choose speed data from the months of March and April, 2011.

We now explain the performance measures used to assess prediction accuracy of the proposed algorithms. We calculate Absolute Percentage Error (APE) for s_i at time t_j for k^{th} prediction horizon $e(s_i,k,t_j)$ as follows:

$$e(s_i,k,t_j) = \frac{\mid \hat{z}_k(s_i,t_j) - z(s_i,t_j) \mid}{z(s_i,t_j)}. \tag{5}$$

Mean Absolute Percentage Error (MAPE) for link s_i and k^{th} prediction horizon $e_s(s_i,k)$, is defined as:

$$e_s(s_i,k) = \frac{1}{d}\sum_{j=1}^{d}\frac{\mid \hat{z}_k(s_i,t_j) - z(s_i,t_j) \mid}{z(s_i,t_j)}, \tag{6}$$

where d is the number of test samples and $\hat{z}_k(s_i,t_j)$ is the predicted speed value at time t_j for k^{th} prediction horizon. MAPE is a commonly used metric to assess the accuracy of traffic prediction algorithms [3], [4], [7], [14]. For the whole network G containing p links, we calculate MAPE $e_G(k)$ for k^{th} prediction horizon as:

$$e_G(k) = \frac{1}{p}\sum_{i=1}^{p}e_s(s_i,k). \tag{7}$$

We calculate Standard Deviation (SD) of error σ_k for each prediction horizon as:

$$\sigma_k = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\left(e_s(s_i,k) - e_G(k)\right)^2}. \tag{8}$$

We use these measures to assess the performance of SVR, ANN and exponential smoothing models for large-scale prediction.

V. Comparison of Different Prediction Algorithms

In this section, we compare the prediction performance of SVR with ANN and exponential smoothing. Fig. 1 provides the network level comparison of performance of the algorithms. Fig. 2 shows the distribution of prediction error across different horizons. Fig. 3 and 4 show the performance of proposed methods for different road categories.

SVR has the smallest MAPE across different prediction horizons for the whole network G (see Fig 1a). It also has the smallest SD of error between different links (see Fig. 1b). ANN provides slightly larger error as compared to SVR. This can be attributed to the problem of local minima associated with ANN training algorithms [34]. Overall, prediction performance for all three algorithms degrades as prediction horizon increases. Performance degradation tends to flatten out especially for data driven methods (SVR, ANN) for large prediction horizons (see Fig. 1a, 3).

Error distribution plots (see Fig. 2) show variations in prediction error across the road network. We observe that distribution of prediction error (MAPE) varies from one prediction horizon to another. We also observe more than one peak for each prediction horizon. This implies that there might exist different groups of links with similar prediction performance. Let us analyze the prediction behavior of different road categories.

Fig. 3 and 4 show how prediction error and standard deviation vary from one road category to another. SVR still provides lowest MAPE (see Fig. 3) and error standard deviation (Fig. 4, except CATA see Fig. 4a). As expected expressways are relatively easy to predict as compared to other road categories. This can be verified by comparing the performance of the predictors for CATA roads (expressways, see Fig. 3a and 4a) with other categories. However, we still observe high SD of error within expressways, especially for large prediction horizons. We find similar patterns for other road categories. Overall SD of error within different road categories is not that different from network wide SD (see Fig. 4 and 1b). We find that roads belonging to the same category can still have dissimilar prediction performance.

Point estimation methods work well to evaluate and compare prediction performance of different methods. However, they provide little insight into the spatial distribution of performance patterns. For instance, we fail to identify which set of links provide worse prediction performance and vice versa. Moreover, we cannot extract temporal performance patterns across the network and for individual roads.

In the next section, we propose unsupervised learning methods to analyze these trends.

VI. SPATIAL AND TEMPORAL PATTERNS in PREDICTION PERFORMANCE

In this section, we consider prediction data analysis as a data-mining problem. For this purpose, we propose unsupervised learning methods to find spatial and temporal performance patterns in large-scale prediction. We analyze the efficiency of proposed algorithms by applying them to the prediction data of SVR.

Traffic prediction studies commonly employ measures such as MAPE to evaluate the performance of algorithms [2]–[11], [13], [14], [16]. These measures are inadequate to provide any information about underlying spatial and temporal behavior of prediction algorithms. If $e_s(s_i,k)$ represents mean prediction error observed for s_i, then MAPE $e_G(k)$ across the test network G in more convenient form will be:

$$e_G(k) = \frac{1}{pd}\sum_{i=1}^{p}\sum_{j=1}^{d}\frac{\mid \hat{z}_k(s_i,t_j) - z(s_i,t_j)\mid}{z(s_i,t_j)}. \tag{9}$$

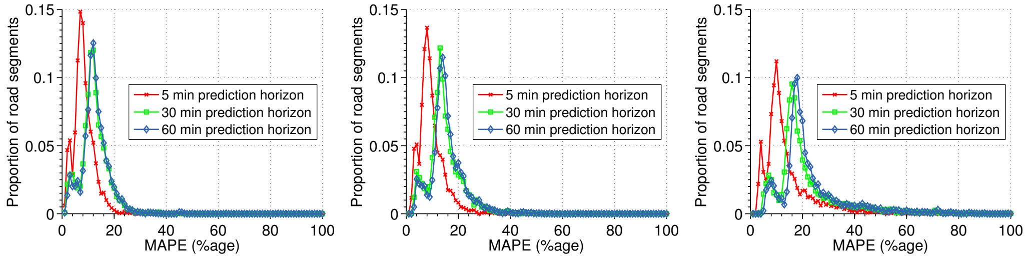
In (9), we obtain averaged out effect of errors across different links and during different time periods. Consider a large network G containing thousands of links and prediction performed for multiple prediction horizons. The prediction error of a particular link s_i at time t_j might be different from that at $t_{j'}$. It might vary from day to day or change during different hours. Similarly, prediction performance between any two links s_i and s_j may also vary significantly. However, in (9) all these trends are averaged out. We observed earlier that prediction performance may not remain uniform across large networks (see Fig. 1, 2, 3 and 4). We also observed that point estimation methods provide little detail about these spatial variations. Moreover, these measures do not give any information about temporal performance variations.

The spatial and temporal patterns provide insight about long-term and short-term predictability. Hence, they can be highly useful for ITS applications like route guidance and traffic management.

For analysis, we consider three components of (9), which are space (s_i), time (t_j) and prediction horizon (k). We perform cluster analysis to obtain spatial prediction patterns. This will help to find roads with overall similar performance across different prediction horizons. To find temporal performance patterns, we combine PCA with k-means clustering. Temporal patterns help us to identify roads with variable and consistent prediction performances during different time periods. We also analyze daily and hourly performance patterns for individual links by applying SOM.

A. Analysis of Spatial Prediction Patterns

In this section, we propose k-means clustering to find spatial prediction patterns. The method creates different groups (clusters) of road segments. We represent these clusters by labels $\{\omega_i\}_{i=1}^{w}$. Each group (cluster) contains roads that provide similar prediction performance across different prediction horizons. To compare the links, we represent each link s_i, by a vector $\mathbf{e}_{s_i} = [e_s(s_i,1)...e_s(s_i,k)]^T$, where $e_s(s_i,k)$ is the MAPE for k^{th} prediction horizon for s_i. The distance

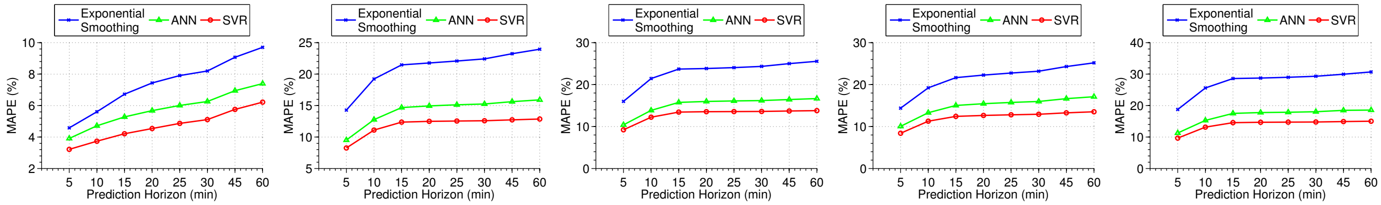


(a) Error distribution of SVR.

(b) Error distribution of ANN.

(c) Error distribution of Exponential Smoothing.

Fig. 2: Error distribution of algorithms for different prediction horizons.



(a) CATA.

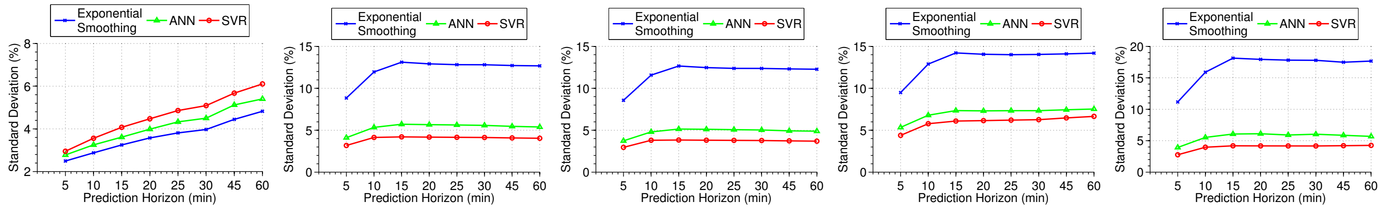
(b) CATB.

(c) CATC.

(d) Slip roads.

(e) Other categories.

Fig. 3: Road category wise performance comparison.



(a) CATA.

(b) CATB.

(c) CATC.

(d) Slip roads.

(e) Other categories.

Fig. 4: Standard deviation of error within each road category.

measure $\Delta_s(s_i, s_j)$ between the links s_i and s_j is defined as:

$$\Delta_s(s_i, s_j) = \sqrt{(\mathbf{e}_{s_i} - \mathbf{e}_{s_j})^T(\mathbf{e}_{s_i} - \mathbf{e}_{s_j})}. \tag{10}$$

We apply unsupervised clustering method as no prior knowledge about the groups of links $\{\omega_i\}_{i=1}^{w}$ is available. Unsupervised learning approach creates clusters of links depending on their predictability (mean prediction error).

We use k-means clustering to find the roads with similar performance [35]. For k-means clustering we need to specify the number of clusters w beforehand. However, this information is not available for the network G. We require the clusters to be composed of roads with similar performance. Moreover, performance of different clusters should be different from one another. This way, we expect to have a cluster of roads with high prediction accuracy and vice versa. This problem is usually referred as cluster validation [36], [37].

We consider commonly applied cluster validation techniques such as Silhouette index [38], Hartigan index [39] and homogeneity and separation index [36], [37] in this study.

Silhouette index $\Psi_{\text{sil}}(s_i)$ for link s_i belonging to cluster ω_j is defined as:

$$\Psi_{\text{sil}}(s_i) = \frac{\beta_2(s_i, \omega_j') - \beta_1(s_i)}{\max\big(\beta_1(s_i), \beta_2(s_i, \omega_j')\big)}, \tag{11}$$

where $\beta_1(s_i)$ is the mean distance (in sense of (10)) of s_i with other links in the cluster ω_j. In (11), $\beta_2(s_i, \omega_j')$ is the mean distance of s_i with links in the nearest cluster ω_j'. We chose the clustering structure with the highest mean value $\zeta_{\text{sil}}(w)$ such that:

$$\zeta_{\text{sil}}(w) = \frac{1}{p}\sum\nolimits_{s_i \in G} \Psi_{\text{sil}}(s_i). \tag{12}$$

Hartigan index $\zeta_{\text{har}}(w)$ for data size N is calculated as [40]:

$$\zeta_{\text{har}}(w) = (N - w - 1)\frac{\Omega(w) - \Omega(w+1)}{\Omega(w+1)}. \tag{13}$$

It considers change in mean intra-cluster dispersion $\Omega(w)$ due to change in the number of clusters w [39], [40]. Consider a clustering structure with w clusters and $\{g_i\}_{i=1}^{w}$ links in each cluster. Intra-cluster dispersion for the structure will be:

$$\Omega(w) = \sum_{j=1}^{w}\sum_{i=1}^{g_j} \Delta_s(s_i, c_j)^2, \tag{14}$$

where $\{c_i\}_{i=1}^{w}$ are the cluster centroids.

We use these indices to select the optimal number of clusters. To this end, we require that the indices agree upon on a certain w^*. We will treat the corresponding clustering structure $\{\omega_i\}_{i=1}^{w^*}$ as the best model for the network G.

B. Analysis of Temporal Prediction Patterns

In this section, we propose methods to infer variations in prediction performance during different time intervals.

Prediction error for a certain set of links τ_i may not change significantly during different times of the day and across different days. For the other group τ_j prediction performance might vary significantly from one period to another. We refer to these as consistent and inconsistent clusters respectively. We combine PCA and k-means clustering to identify consistent and inconsistent clusters.

We also analyze the temporal performance trends for individual roads. For a given link s_i some days (hours) will have similar performance patterns and vice versa. We apply SOM to extract these trends.

We now explain our proposed methods to infer network level and link level temporal prediction patterns.

1) Network level temporal prediction patterns: We consider variations in prediction error of the links during different days and hours to group them together. We use Principal Component Analysis (PCA) to deduce these daily and hourly performance patterns.

We define daily $\{\mathbf{d}_j(s_i) \in \mathbb{R}^{n_d}\}_{j=1}^{m_d}$ and hourly $\{\mathbf{h}_l(s_i) \in \mathbb{R}^{n_h}\}_{l=1}^{m_h}$ performance patterns as follows. The vector $\{\mathbf{d}_j(s_i)\}_{j=1}^{m_d}$ comprises the APE for all the time periods and prediction horizons for that day $\{j\}_{j=1}^{m_d}$ for link s_i. The vector $\{\mathbf{h}_l(s_i)\}_{l=1}^{m_h}$ contains APE across all the days and prediction horizons for the link s_i during the hour $\{l\}_{l=1}^{m_h}$.

The daily variation matrix $\mathbf{D}(s_i) = [\mathbf{d}_1(s_i)...\mathbf{d}_{m_d}(s_i)]$ and the hourly variation matrix $\mathbf{H}(s_i) = [\mathbf{h}_1(s_i)...\mathbf{h}_{m_h}(s_i)]$ contain all such patterns for the link s_i. To quantify performance variations within different days $\{\mathbf{d}_j(s_i)\}_{j=1}^{m_d}$ and hours $\{\mathbf{h}_l(s_i)\}_{l=1}^{m_h}$, we construct corresponding covariance matrices $\Sigma_d(s_i)$ and $\Sigma_h(s_i)$. By centralizing $\{\mathbf{d}_j(s_i)\}_{j=1}^{m_d}$ and $\{\mathbf{h}_l(s_i)\}_{l=1}^{m_h}$ about their means we obtain $\mathbf{D}'(s_i)$ and $\mathbf{H}'(s_i)$ respectively. The covariance matrices $\Sigma_d(s_i)$ and $\Sigma_h(s_i)$ are calculated as follows:

$$\Sigma_d(s_i) = \frac{1}{n_d}\mathbf{D}'(s_i)^T\mathbf{D}'(s_i), \tag{15}$$

$$\Sigma_h(s_i) = \frac{1}{n_h}\mathbf{H}'(s_i)^T\mathbf{H}'(s_i). \tag{16}$$

Eigenvalue decomposition of covariance matrices will yield:

$$\Sigma_d(s_i) = \mathbf{U}_d(s_i)\Lambda_d(s_i)\mathbf{U}_d(s_i)^T, \tag{17}$$

$$\Sigma_h(s_i) = \mathbf{U}_h(s_i)\Lambda_h(s_i)\mathbf{U}_h(s_i)^T, \tag{18}$$

where matrices $\{\mathbf{U}_j(s_i) = [\varphi_{j1}(s_i)...\varphi_{jm_j}(s_i)]\}_{j\in\{d,h\}}$ and $\{\Lambda_j(s_i)\}_{j\in\{d,h\}}$ contain the normalized eigenvectors and the corresponding eigenvalues of $\{\Sigma_j(s_i)\}_{j\in\{d,h\}}$ respectively. We calculate Principal Components (PC) by rotating the data along the direction of eigenvectors (direction of maximum variance) of the covariance matrix [41]:

$$\mathbf{P}_d(s_i) = \mathbf{D}'(s_i)\mathbf{U}_d(s_i), \tag{19}$$

$$\mathbf{P}_h(s_i) = \mathbf{H}'(s_i)\mathbf{U}_h(s_i). \tag{20}$$

Each eigenvalue $\lambda_j(s_i)$ represents the amount of variance in the data explained by the corresponding PC $\mathbf{p}_j(s_i)$. For instance, let us consider daily performance patterns. Strongly correlated (pointing in the similar direction) error profiles $\{\mathbf{d}_j(s_i)\}_{j=1}^{m_d}$ for link s_i imply that prediction errors across different days $\{j\}_{j=1}^{m_d}$ follow similar patterns. In this case, few PC $f_d(s_i)$ can cover most of the variance in the daily error performance data $\mathbf{D}'(s_i)$ of s_i [41]. If most days show independent behavior, then we would require more PC $f'_d(s_i)$ to explain same percentage of variance in data.

The same goes for hourly error patterns. For a link s_i with similar performance across different hours, we require a small number of hourly PC $f_h(s_i)$. In case of large hourly performance variations, we will need a large number of hourly PC to explain the same amount of variance.

The number of PC $\{f_j(s_i)\}_{j\in\{d,h\}}$ are chosen using a certain threshold of total variance η_σ (typically 80%) in the data [41]. We define following distance measure to compare consistency in prediction performance of two links:

$$\Delta_t(s_i,s_j) = \sqrt{\big(f_d(s_i) - f_d(s_j)\big)^2 + \big(f_h(s_i) - f_h(s_j)\big)^2}. \tag{21}$$

We find the clusters of consistent τ_1 and inconsistent τ_2 links by applying (21) and k-means clustering. Consistent (inconsistent) links will have similar (variable) performance patterns across days and during different hours.

2) Link level temporal prediction patterns: In the previous subsection, we proposed a method to find consistent and inconsistent links. In this section, we propose an algorithm to cluster days/hours with similar performance for each road segment s_i. The algorithm also conserves the topological relation between the clusters. Topological relations are considered in the sense of mean prediction performance of different clusters [42]. To this end, we use Self Organizing Maps (SOM). Self Organizing Maps belong to category of neural networks that can perform unsupervised clustering. In SOM, each cluster is represented by a neuron. Neurons are organized in a grid pattern $\mathfrak{M}$. The weight $\{\mathbf{q}_\rho\}_{\rho\in\mathfrak{M}}$ of the neuron represents the center of the cluster ρ. We use Kohonen rule [42] to find the optimal weights (cluster centers).

Consider a road segment s_i with prediction performance matrix $\mathbf{D}(s_i)$. The matrix $\mathbf{D}(s_i)$ is composed of daily prediction error profiles $\{\mathbf{d}_j(s_i)\}_{j=1}^{m_d}$ for m_d days. Let us represent each day by index $\{j\}_{j=1}^{m_d}$. We aim to identify subset (cluster) of days $\rho \subseteq \{j\}_{j=1}^{m_d}$ with similar performance patterns. Secondly, we aim to find a 2-D grid $\mathfrak{M}$ for clusters. In the grid, clusters $\rho \in \mathfrak{M}$ with similar behavior (daily prediction performance) will be placed adjacent to each other. However, each daily performance profile contains data for multiple prediction horizons and time instances. It is hard to visualize the data in such high dimensional representation. We apply SOM to visualize and map daily performance patterns on a 2-D clustering grid $\mathfrak{M}$ [43].

We apply the same procedure to find different groups of hourly patterns with similar prediction performance for each road segment s_i. To this end, SOM performs clustering by considering hourly profile matrix $\mathbf{H}(s_i)$ for each link s_i.

In this section, we have proposed unsupervised learning methods to find spatial and temporal performance patterns. In the next section, we apply these proposed performance analysis methods to prediction data of SVR and provide results.

VII. Results and Discussion

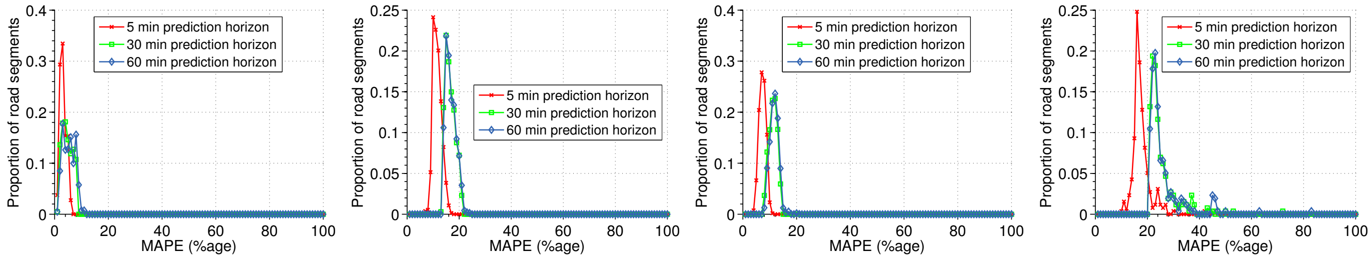
Let us start with the spatial performance patterns. We apply k-means clustering to find road segments with similar

TABLE II: Distribution of links in temporal clusters

Temporal	Cluster Center		Spatial Cluster (ω)				Total
Cluster (τ)	f_d	f_h	cluster 1 (ω_1)	cluster 2 (ω_2)	cluster 3 (ω_4)	cluster 4 (ω_4)	links
Cluster 1 (τ_1)	5	6	735	610	125	30	1500
Cluster 2 (τ_2)	13	12	66	2054	1176	228	3524

TABLE III: Performance for different spatial clusters

Cluster	Prediction Horizon							
	5 min	10 min	15 min	20 min	25 min	30 min	45 min	60 min
Cluster 1 (ω_1)	2.69%	3.24%	3.66%	3.86%	4.05%	4.20%	4.61%	4.89%
Cluster 2 (ω_2)	6.79%	9.16%	10.38%	10.54%	10.64%	10.72%	10.95%	11.14%
Cluster 3 (ω_3)	11.06%	14.60%	15.90%	16.01%	16.05%	16.08%	16.18%	16.28%
Cluster 4 (ω_4)	17.18%	23.04%	24.58%	24.81%	24.98%	25.10%	25.41%	25.62%



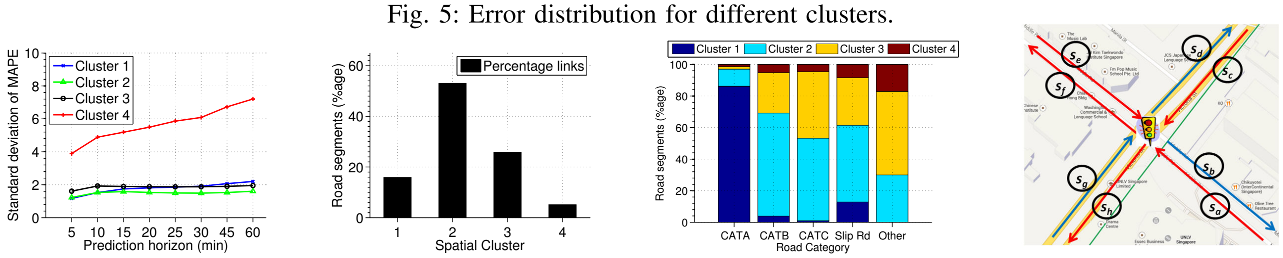
(a) Error distribution for cluster 1.

(b) Error distribution for cluster 2.

(c) Error distribution for cluster 3.

(d) Error distribution for cluster 4.

Fig. 5: Error distribution for different clusters.



(a) Intra Cluster error standard deviation.

(b) Percentage Distribution of road segments.

(c) Road categories and spatial clusters.

(d) Sections belonging to different clusters at a particular intersection.

Fig. 6: Properties of spatial clusters. In Fig. 6d, blue links (s_b, s_d, s_g) and red links (s_a, s_c, s_e, s_f, s_h) correspond to spatial clusters ω_2 and ω_3 respectively.

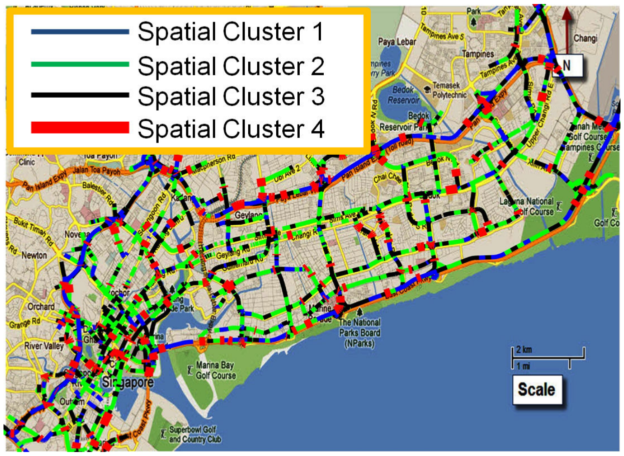


Fig. 7: Road segments belonging to different spatial clusters.

performance across the prediction horizons. We consider three different validation indices to obtain the optimal number of clusters for the test network. All three validation methods yield 4 clusters as the optimal structure. Spatial distribution of different clusters is shown in Fig. 7. Error distributions for each cluster across different prediction horizons are shown in Fig. 5.

The first cluster (ω_1) consists of roads with high prediction accuracy (see Table III). We refer to this group of links as best performing cluster (cluster 1, ω_1). Most of the roads in the network (around 75%, see Fig. 6b) belong to clusters 2 (ω_2) and 3 (ω_3). These two clusters represent the performance trends of majority of roads in the network. It is interesting to note that combined prediction performance of these two clusters (see Table III) is worse than mean prediction accuracy of the whole network (see Fig. 1a). In this case, network wise MAPE (see Fig. 1a) provides a slightly inflated depiction of overall prediction accuracy. This is due to the high prediction accuracy of best performing cluster. Finally, roads belonging to cluster 4 (ω_4) have the highest prediction error from each road category (see Table III). The proposed prediction algorithm performs poorly for this set of road segments. We refer to this cluster as the worst performing cluster.

The spatial clusters also have some intuitive sense. For instance, most of the expressways belong to best performing cluster (around 80%, see Fig. 6c). However, a small group of roads from other categories also belong to this cluster. Interestingly, a small percentage of expressways also appear in other clusters (see Fig. 6c). Some expressway sections even appear in cluster 4 (ω_4). The expressway sections belonging to ω_4 are mostly situated at busy exits. Naturally, it is relatively hard to predict traffic on such sections. Moreover, a higher ratio of CATC roads belong to cluster ω_3, as compared to roads in CATB (see Fig. 6c). Likewise, majority of CATD and CATE roads (referred as others in Fig. 6c) also belong to spatial cluster 3.

Overall, roads within each spatial cluster show similar performance (see Fig. 5). Consequently, we find small SD of error within each cluster across the prediction horizons (see Fig. 6a). Behavior of worst performing cluster is an exception in this case.

Spatial clusters can also provide useful information about the relative predictability of different road segments. Consider the intersection shown in Fig. 6d. It shows that roads carrying inbound traffic may have different prediction performance as compared to roads carrying outbound traffic. In this case, links carrying traffic towards downtown area (s_a, s_c, s_f, s_h) tend to show degraded prediction performance (cluster ω_3).

We apply PCA and k-means clustering to find temporal performance patterns. We create two clusters for the roads segments in this regard. We refer to these clusters as consistent cluster (τ_1) and inconsistent cluster (τ_2). Table II summarizes the properties of these two clusters. Prediction performance of road segments in consistent cluster remains uniform across days and during different hours. Links in the inconsistent cluster have variable prediction performance during different time periods.

We observe that roads with similar mean prediction performance (see Fig. 6a) can still have different temporal performance patterns (see Table II). All these spatial clusters $(1,2,3)$ have small intra-cluster SD (see Fig. 6a). Majority of roads in spatial cluster 1 (best performing cluster) are part of consistent cluster. However, a small proportion of roads from best performing cluster are also part of inconsistent cluster. We observe this trend in other spatial clusters as well. Although majority of the links in spatial clusters 2 and 3 are part of inconsistent cluster. Still they both have a sizable proportion of consistent links (see Table II). Even in the case of worst performing cluster, a small subset of roads are part of consistent cluster. These road segments report high prediction error consistently during most of the time periods.

Temporal performance analysis shows that links with similar overall prediction behavior can still have variable temporal performance. To analyze such trends in details, we focus on two specific links s_1 and s_2 in the network which have the following properties. They both belong to the same road category (CATA). Furthermore, both of them are from the best performing cluster. However, road segment s_1 is part of consistent cluster and s_2 is from inconsistent cluster. We apply SOM to analyze variations in daily performance patterns of these two links.

Fig. 8a and 8b show different properties of daily performance patterns for the consistent link s_1. In Fig. 8a, we present the composition of each cluster for s_1. The entry within each hexagon denotes the number of days belonging to that cluster. Fig. 8b shows relative similarity of each cluster and its neighbors. For the consistent link, we find that most of the days fall into four clusters (see Fig. 8a). SOM helps us to conserve the topological relations of these clusters. The four main clusters are positioned adjacent to each other (see Fig. 8a). This implies that these clusters represent days with similar daily performance patterns (see Fig. 8b). For this road segment, we observe similar performance patterns for most of the days (see Fig. 8b).

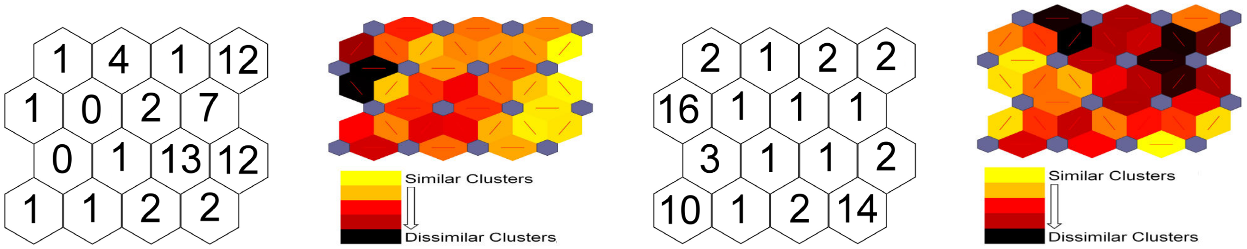
Now let us consider the behavior of inconsistent link s_2. Fig. 8c and 8d show the prediction patterns for the road segment. In this case, we find three major clusters. The rest of the days are scattered into other small clusters (see Fig. 8c). Even these three clusters represent quite different daily performance patterns (see Fig. 8d).

Both of these links belong to same road category and spatial cluster. However, their daily performance patterns are quite different from each other. In case of consistent link, we observe that prediction error patterns do not vary significantly on daily basis. For inconsistent link, performance patterns vary significantly, from one day to another.

ITS applications such as route guidance, which rely on prediction data, are vulnerable to variations in prediction error. Spatial and temporal performance patterns provide insight into prediction behavior of different road segments. Consider the example of a route guidance application. The route guidance algorithm can assign large penalties to links belonging to clusters with large prediction errors (e.g. clusters ω_3, ω_4). For instance, with spatial clustering, we can see that planning routes by incorporating expressway sections from cluster ω_4 (worst performing links) may not be a good idea. The spatial clusters also serve another important purpose. They provide information about the relative predictability of different road segments in a particular network. Furthermore, temporal clusters can help the algorithm to avoid inconsistent links. These links might have low average prediction error. However, their prediction performance may vary widely from one time instance to another. Again, consider the example of a route guidance algorithm. It would be better to plan a route by incorporating roads with known performance patterns, even if they have slightly larger prediction error than inconsistent roads. The application can utilize these spatial and temporal markers to provide routes which are more robust to variations in prediction performance.

VIII. Conclusion

In this paper, we proposed unsupervised learning methods to analyze spatial and temporal performance trends in SVR



(a) Distribution of different clusters (road segment s_1).

(b) Similarity between different clusters (road segment s_1).

(c) Distribution of different clusters (road segment s_2).

(d) Similarity between different clusters (road segment s_2).

Fig. 8: Daily performance patterns for road segments s_1 (consistent link) and s_2 (inconsistent link). Hexagons represent the clusters and the entry within each hexagon denotes the number of days in that cluster. Color bar represents cluster similarity in terms of Euclidean distance.

based large-scale prediction. We performed large-scale traffic prediction for multiple prediction horizons with SVR. Our analysis focused on a large and heterogeneous road network. SVR produced better prediction accuracy in comparison with other forecasting algorithms. We also observed that traditional performance evaluation indices such as MAPE fail to provide any insight about performance patterns. For instance, traffic speeds on some roads were found to be more predictable than others, and their performances remained uniform across different time periods. For some other roads, such patterns varied significantly across time. We proposed unsupervised learning methods to infer these patterns in large scale prediction. We used prediction data from SVR to assess the efficiency of these unsupervised learning algorithms. These methods provide a systematic approach for evaluating the predictability and performance consistency of different road segments. Such insights may be useful for ITS applications that use prediction data to achieve time-sensitive objectives.

For future work, we propose to incorporate these performance patterns into predictive route guidance algorithms. This can lead to the development of route recommendation algorithms which are more robust to variations in the future state of road networks.

References

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

[2] J. Park, D. Li, Y. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips, "Real time vehicle speed prediction using a neural network traffic model," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2011, pp. 2991–2996.

[3] C. Wu, J. Ho, and D. Lee, "Travel-time prediction with support vector regression," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 276–281, 2004.

[4] L. Vanajakshi and L. Rilett, "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed," in *Intelligent Vehicles Symposium, 2004 IEEE*. Ieee, 2004, pp. 194–199.

[5] Y. Zhang and Y. Liu, "Traffic forecasting using least squares support vector machines," *Transportmetrica*, vol. 5, no. 3, pp. 193–213, 2009.

[6] C. Quek, M. Pasquier, and B. Lim, "Pop-traffic: A novel fuzzy neural approach to road traffic analysis and prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 133–146, 2006.

[7] L. Vanajakshi and L. Rilett, "Support vector machine technique for the short term prediction of travel time," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 600–605.

[8] E. Vlahogianni, M. Karlaftis, and J. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 211–234, 2005.

[9] A. Ding, X. Zhao, and L. Jiao, "Traffic flow time series prediction based on statistics learning theory," in *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*. IEEE, 2002, pp. 727–730.

[10] C. van Hinsbergen, J. Van Lint, and H. Van Zuylen, "Bayesian committee of neural networks to predict travel times with confidence intervals," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 5, pp. 498–509, 2009.

[11] D. Park and L. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999.

[12] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 871–882, 2013.

[13] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[14] B. Smith, B. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.

[15] Z. Sun, Y. Wang, and J. Pan, "Short-term traffic flow forecasting based on clustering and feature selection," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 577–583.

[16] J. McFadden, W. Yang, and S. Durrans, "Application of artificial neural networks to predict speeds on two-lane rural highways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1751, no. -1, pp. 9–17, 2001.

[17] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

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

[19] P. Lingras and P. Mountford, "Time delay neural networks designed using genetic algorithms for short term inter-city traffic forecasting," in *Engineering of Intelligent Systems*. Springer, 2001, pp. 290–299.

[20] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of Transportation Engineering*, vol. 131, no. 10, pp. 771–779, 2005.

[21] J. Zhu and T. Zhang, "A layered neural network competitive algorithm for short-term traffic forecasting," in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*. IEEE, 2009, pp. 1–4.

[22] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[23] C. Chang and C. Lin, "Training v-support vector regression: theory and algorithms," *Neural Computation*, vol. 14, no. 8, pp. 1959–1977, 2002.

[24] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[25] S. Sundaram, H. N. Koutsopoulos, M. Ben-Akiva, C. Antoniou, and R. Balakrishna, “Simulation-based dynamic traffic assignment for short-term planning applications,” *Simulation Modelling Practice and Theory*, vol. 19, no. 1, pp. 450–462, 2011.

[26] P. Moriarty and D. Honnery, “Low-mobility: The future of transport,” *Futures*, vol. 40, no. 10, pp. 865–872, 2008.

[27] K. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, “Predicting time series with support vector machines,” *Artificial Neural NetworksICANN’97*, pp. 999–1004, 1997.

[28] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, “Shrinking the tube: a new support vector regression algorithm,” *Advances in neural information processing systems*, pp. 330–336, 1999.

[29] C. Chang and C. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[30] S. Keerthi and C. Lin, “Asymptotic behaviors of support vector machines with gaussian kernel,” *Neural computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[31] R. Frank, N. Davey, and S. Hunt, “Time series prediction and neural networks,” *Journal of Intelligent & Robotic Systems*, vol. 31, no. 1, pp. 91–103, 2001.

[32] H. Chen and S. Grant-Muller, “Use of sequential learning for short-term traffic flow forecasting,” *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 5, pp. 319–336, 2001.

[33] B. Williams, P. Durvasula, and D. Brown, “Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1644, no. -1, pp. 132–141, 1998.

[34] M. Gori and A. Tesi, “On the problem of local minima in backpropagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 76–86, 1992.

[35] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002.

[36] K. Wang, B. Wang, and L. Peng, “Cvap: Validation for cluster analyses,” *Data Science Journal*, no. 0, p. 904220071, 2009.

[37] R. Sharan, A. Maron-Katz, and R. Shamir, “Click and expander: a system for clustering and visualizing gene expression data,” *Bioinformatics*, vol. 19, no. 14, pp. 1787–1799, 2003.

[38] P. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[39] J. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[40] I. Pelczer and H. Cisneros-Iturbe, “Identification of rainfall patterns over the valley of mexico,” in *Proceedings of 11th International Conference of Urban Drainage, Edinburgh, Scotland, UK*, 2008.

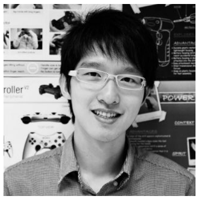
[41] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[42] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[43] B. S. Penn, “Using self-organizing maps to visualize high-dimensional data,” *Computers & Geosciences*, vol. 31, no. 5, pp. 531–544, 2005.

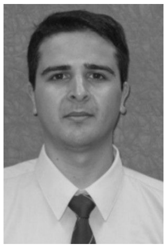
Muhammad Tayyab Asif is a Ph.D. student at the Nanyang Technological University, Singapore. He received his Bachelor of Engineering degree from University of Engineering and Technology, Lahore. Previously, he worked at Ericsson as design engineer in the domain of IP-networks. His research interests include sensor fusion, network optimization, and modeling of large-scale networks.

Justin Dauwels is an Assistant Professor with School of Electrical & Electronic Engineering at the Nanyang Technological University (NTU) in Singapore. His research interests are in Bayesian statistics, iterative signal processing, and computational neuroscience. He obtained the PhD degree in electrical engineering at the Swiss Polytechnical Institute of Technology (ETH) in Zurich in December 2005. He has been a JSPS postdoctoral fellow (2007), a BAEF fellow (2008), a Henri-Benedictus Fellow of the King Baudouin Foundation (2008), and a JSPS invited fellow (2010).



Chong Yang Goh is a Ph.D. student at the Massachusetts Institute of Technology in Cambridge, MA. Previously, he received his B.Eng. degree from Nanyang Technological University, Singapore. His recent works focus on the development of real-time decision models and algorithms that incorporate large-scale, asynchronous data sources, with applications in transportation.

Ali Oran received his masters and doctoral degrees from University of California, San Diego in 2005, and 2010, respectively. Shortly after his graduation, he moved to Singapore to work as a Postdoctoral Associate at Future Urban Mobility (FM) group of Singapore-MIT Alliance for Research Technology (SMART) Center. His research interests include vehicle localization, vehicle routing, and decision making under uncertainty.

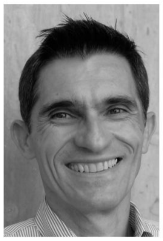


Esmail Fathi was born in Loshan, Iran. He got his masters degree from Shahid Beheshti University in 2007. He worked as an associate researcher at Nanyang Technological University (NTU). His research topic was Tracking and Predicting Traffic Flow in Dynamic Urban Transportation Networks. Currently he works at Siemens Singapore as a Project Engineer.

Muye Xu received the BEng Honors degree in Electrical and Electronic Engineering from Nanyang Technological University in 2012. His research interest includes machine learning, transportation system and operations research. Currently Muye Xu is a technology analyst with Deutsche Bank’s Trading Technology Group.

Menoth Mohan Dhanya is a Research Associate at Nanyang Technological University, Singapore. She received her Bachelor’s degree in Mechatronics Engineering from Anna University of Technology, India and her Master’s degree in Computer Control and Automation from Nanyang Technological University, Singapore. Her research interests cover machine learning, signal processing and automation.

Nikola Mitrovic received the bachelor degree in traffic engineering from the University of Belgrade, Serbia, in 2009. He obtained masters degree at Department of civil engineering at Florida Atlantic University, USA, in 2010. He is currently a PhD student with the department of Electrical and Electronic engineering at Nanyang Technological University. His research topics are traffic modeling, intelligent transportation systems, and transportation planning.



Patrick Jaillet received the Ph.D. degree in operations research from the Massachusetts Institute of Technology, Cambridge, USA, in 1985. He is currently the Dugald C. Jackson Professor in the Department of Electrical Engineering and Computer Science, and the co-director of the Operations Research Center, Massachusetts Institute of Technology, Cambridge, USA. His research interests include algorithm design and analysis for on-line problems; real-time and dynamic optimization; network design and optimization; and probabilistic combinatorial optimization.