Topology-regularized universal vector autoregression for traffic forecasting in large urban areas

Florin Schimbinschi[a], Luis Moreira-Matias[b], Vinh Xuan Nguyen[a], James Bailey[a]

[a] *University of Melbourne, Victoria, Australia*
[b] *NEC Laboratories Europe, Heidelberg, Germany*

Abstract

Autonomous vehicles are soon to become ubiquitous in large urban areas, encompassing cities, suburbs and vast highway networks. In turn, this will bring new challenges to the existing traffic management expert systems. Concurrently, urban development is causing growth, thus changing the network structures. As such, a new generation of adaptive algorithms are needed, ones that learn in real-time, capture the multivariate nonlinear spatio-temporal dependencies and are easily adaptable to new data (e.g. weather or crowd-sourced data) and changes in network structure, without having to retrain and/or redeploy the entire system.

We propose learning Topology-Regularized Universal Vector Autoregression (TRU-VAR) and examplify deployment with of state-of-the-art function approximators. Our expert system produces reliable forecasts in large urban areas and is best described as scalable, versatile and accurate. By introducing constraints via a topology-designed adjacency matrix (TDAM), we simultaneusly reduce computational complexity while improving accuracy by capturing the non-linear spatio-temporal dependencies between timeseries. The strength of our method also resides in its redundancy through modularity and adaptability via the TDAM, which can be altered even while the system is deployed. The large-scale network-wide empirical evaluations on two qualitatively and quantitatively different datasets show that our method scales well and can be trained efficiently with low generalization error.

Email addresses: florin.schimbinschi@unimelb.edu.au (Florin Schimbinschi), luis.matias@neclab.eu (Luis Moreira-Matias), vinh.nguyen@unimelb.edu.au (Vinh Xuan Nguyen), baileyj@unimelb.edu.au (James Bailey)

We also provide a broad review of the literature and illustrate the complex dependencies at intersections and discuss the issues of data broadcasted by road network sensors. The lowest prediction error was observed for TRU-VAR, which outperforms ARIMA in all cases and the equivalent univariate predictors in almost all cases for both datasets. We conclude that forecasting accuracy is heavily influenced by the TDAM, which should be tailored specifically for each dataset and network type. Further improvements are possible based on including additional data in the model, such as readings from different metrics.

Keywords: topology regularized universal vector autoregression, multivariate timeseries forecasting, spatiotemporal autocorrelation, traffic prediction, big data, structural risk minimization

---

1. Introduction

Expert systems are at the forefront of intelligent computing and 'soft Artificial Intelligence (soft AI)'. Typically, they are seamlessly integrated in complete business solutions, making them part of the core value. In the current work we propose a system for large-area traffic forecasting, in the context of the challenges imposed by rapidly growing urban mobility networks, which we outline in the following paragraphs. Our solution relies on the formulation of a powerful inference system which is combined with expert domain knowledge of the network topology, and that can be seamlessly integrated with a control schema.

Fully autonomous traffic implies an *omniscient* AI which is comprised of two expert sytems, since it has to be able to both perceive and efficiently control traffic in real time. This implies the observation of both the network state and the entities on the network. Therefore, sensing (perception) can be done via (i) passive sensors (e.g. induction loops, traffic cameras, radar) or (ii) mobile ones (e.g. Global Positioning Systems (GPS), Bluetooth, Radio Frequency Identification (RFID)). While the crowdsourced data from moving sensors (ii) can provide high-granularity data to fill accurate Origin-Destination (O-D) matrices, their penetration rate is still scarce to scale up (Moreira-Matias et al., 2016).

Forecasting traffic is a function of control as well, since changing traffic rules or providing route recommendations can have an impact on the network load. However, there are factors that are not a function of control, such as

human error or extreme weather conditions, which are the actual unforeseen causes of congestion. Therefore, during the transition to fully autonomous traffic control, there will be an even greater need for accurate predictions. There are also many possible intelligent applications such as a personalized copilots making real time route suggestions based on users preferences and traffic conditions, economical parking metering, agile car pooling services, all of these paving the way towards fully autonomous self driving cars. Not surprisingly, the work in simulation by Au et al. (2015) has shown that semi-autonomous intersection management can greatly decrease traffic delay in mixed traffic conditions (no autonomy, regular or adaptive cruise control, or full autonomy). This is possible by linking cars in a semi-autonomous way, thus solving the congestion 'wave' problem, if most of the vehicles are semi-autonomous.

Traffic prediction will therefore become paramount as urban population is growing and autonomous vehicles will become ubiquitous for both personal and public transport as well as for industrial automation. Currently, one may argue that automatic traffic might be a self-defeating process. A common scenario might be in the case when the recommendations from a prediction expert system are identical for all users in the network. In this case, new congestions can and will be created (most vehicles take the same route), which in turn invalidate the forecasts. This is evidently caused by *poor control policies* or a lack of adequate infrastructure. Fortunately, simple solutions for both of these issues exist, here we refer the reader to two references for each potential issue. Çolak et al. (2016) formulate the control problem as a collective travel time savings optimization problem, under a centralized routing scheme. Different quantified levels of social good (vs greedy individual) are tweaked in order to achieve significant collective benefits. A simple (but more socially challenging) way to overcome the infrastructure problem is recommendations for car pooling as suggested by Guidotti et al. (2016).

Concerning the traffic prediction literature, most research effort is focused on motorways and freeways (Ko et al., 2016; Su et al., 2016; Ahn et al., 2015; Hong et al., 2015; Asif et al., 2014; Lippi et al., 2013; Lv et al., 2009; Wang et al., 2008; Zheng et al., 2006; Wu et al., 2004; Stathopoulos & Karlaftis, 2003), while other methods are only evaluated on certain weekdays and / or at particular times of the day (Su et al., 2016; Wu et al., 2016). These methods usually deploy univariate statistical models that do not take into consideration all the properties that can lead to satisfactory generalization accuracy in the context of growth and automation in urban areas, namely:

Table 1: Comparison of TRU-VAR properties with state of the art traffic forecasting methods. Properties that couldn't be clearly defined as either present or absent were marked with '∼'.

Property	STARIMA[1]	DeepNN[2]	VSSVR[3]	STRE[4]	GBLS[5]	TRU-VAR[6]
1) Online learning	∼	✓	∼	✗	✗	✓
2) Nonlinear representation	✗	✓	✓	✓	✗	✓
3) Low complexity	✗	✗	✗	✓	∼	✓
4) Topological constraints	✗	✗	∼	✓	✓	✓
4) Non-static spatio-temporal	✗	✓	∼	✗	✓	✓
5) Infrastructure versatility	✗	∼	✗	∼	✓	✓
6) Easy to (re)deploy	✗	✗	✗	✗	✗	✓
7) Customizable design matrix	✗	✗	∼	∼	∼	✓
7) Distinct model per series	✗	✗	✓	✗	✗	✓
7) Transferable cross-series	✗	✗	∼	✓	∼	✓
8) Adaptable to multi-metrics	✗	✓	✓	∼	∼	✓

1) real-time (online) learning; 2) model nonlinearity in the spatio-temporal domain; 3) low computation complexity and scalability to large networks; 4) contextual spatio-temporal multivariable regression via topological constraints; 5) versatility towards a broad set of infrastructure types (urban, suburban, freeways); 6) adaptation to changes in network structure, without full-network redeployment; 7) redundancy and customization for each series and adjacency matrix; 8) encoding time or using multi-metric data.

In the current work we address these issues and propose a multivariate traffic forecasting method that can capture spatio-temporal correlations, is redundant (fault tolerant) through modularity, adaptable (trivial to redeploy) to changing topologies of the network via its modular topology-designed adjacency matrix (TDAM). Our method can be efficiently deployed over large networks of broad road type variety with low prediction error and therefore generalizes well across scopes and applications. We also show (Figure 12) that our method can predict within reasonable accuracy even up to two hours in the future – the error increases linearly and the increase rate depends on the function approximator, the TDAM and the quality of the data. We provide a comparison with state of the art methods in Table 1 according to properties that we believe are essential to the next generation of intelligent expert systems for traffic forecasting:

Our contributions are as follows: (i) We propose learning Topology-Regularized Universal Vector Autoregression (TRU-VAR), a novel method that can absorb spatio-temporal dependences between multiple sensor stations; (ii) The extension of TRU-VAR to nonlinear universal function approximators over the existing state of the art machine learning algorithms, resulting in an exhaustive comparison; (iii) Evaluations performed on two large scale real world datasets, one of which is novel; (iv) Comprehensive coverage of the literature, and an exploratory analysis considering data quality, preprocessing and possible heuristics for choosing the topology-designed adjacency matrix (TDAM).

Our conclusions are: TRU-VAR shows promising results, scales well and is easily deployable with new sensor installations; careful choice of the adjacency matrix is necessary according to the type of dataset used; high resolution data (temporal as well as spatial) is essential; missing data should be marked in order to distinguish it from real congestion events; given that the methods show quite different results on the two datasets we argue that a public set of large-scale benchmark datasets should be made available for testing the prediction performance of novel methods.

2. Related work

Traffic forecasting methodologies can be challenging to characterize and compare due to the lack of a common set of benchmarks. Despite the numerous methods that have been developed, there is yet none that is modular, design-flexible and adaptable to growing networks and changing scopes. The scope (e.g. freeway, arterial or city) and application can differ across methods. Therefore, it is not trivial to assess the overall performance of different approaches when the datasets and metrics differ. Often, subsets of the network are used for evaluating performance as opposed to the general case of network-wide prediction, which includes highways as well as suburban and urban regions. Furthermore, off-peak times and weekends are also sometimes excluded. For critical reviews of the literature we point the reader to (Oh et al., 2015; Vlahogianni et al., 2014; Van Lint & Van Hinsbergen, 2012;

[1] Kamarianakis & Prastacos (2005) [2] Lv et al. (2015) [3] Xu et al. (2015) [4] Wu et al. (2016) [5] Salamanis et al. (2016) [6] Nonliearity dependent on the function approximator. Careful design of the topological adjacency matrix is essential and requires domain knowledge in order to define the appropriate heuristics.

Vlahogianni et al., 2004; Smith et al., 2002; Smith & Demetsky, 1997).

Traffic metric types for sensor loops and floating car data: When it comes to metrics, speed, density and flow can be used as target prediction metrics. Flow (or volume) is the number of vehicles passing though a sensor per time unit (usually aggregated in 1, 5 or 15 minute intervals). Density is the number of vehicles per kilometre. It was shown (Clark, 2003) that multi-metric predictors can result in lower prediction error. That is, variety of input data metrics is beneficial. As to the metric being predicted, some authors argue that flow is more important due to its stability (Levin & Tsao, 1980) while others (Dougherty & Cobbett, 1997) have found that traffic conditions are best described using flow and density as opposed to speed, as output metric. Nevertheless, there is a large amount of work where speed is predicted, as opposed to flow or density (Salamanis et al., 2016; Fusco et al., 2015; Mitrovic et al., 2015; Asif et al., 2014; Kamarianakis et al., 2012; Park et al., 2011; Lee et al., 2007; Dougherty & Cobbett, 1997). This data can come from either loop sensors (two are needed) or floating car data such as that collected from mobile phones, GPS navigators, etc. For the traffic assignment problem (balancing load on the network), density is a more appropriate metric as opposed to flow, according to the recent work in (Kachroo & Sastry, 2016). The authors make the observation that vehicles travel with a speed which is consistent with the traffic density as opposed to flow. For the current work we therefore use only *flow* data for both the independent and dependent target variables, since there were no other metrics readily available for the two datasets. We would have adopted a multi-metric approach (e.g. using speed and density data as additional input metrics) had we been able to acquire such data. However, the extension is trivial and we aim to show this in future work.

2.1. Traffic prediction methods

A comparison between parametric and non-parametric methods for single point traffic flow forecasting based on theoretical foundations (Smith et al., 2002) argues that parametric methods are more time consuming while non-parametric methods are better suited to stochastic data. An empirical study with similar objectives (Karlaftis & Vlahogianni, 2011) provides a comparison of neural networks methods with statistical methods. The authors suggest a possible synergy in three areas: core model development, analysis of large data sets and causality investigation. While we focus on short-term

forecasting, it is evident that forecast accuracy degrades with a larger prediction horizons. We show that the prediction error increases linearly for our method in Figure 12 on page 35. The rate of increase depends on the function approximator that is being used and the design of the topology matrix. For long-term forecasting (larger prediction horizons – Sec. 3, page 14), continuous state-space models such as Kalman filters (Wang et al., 2008) or recurrent neural networks (RNN) (Dia, 2001) have outperformed traditional 'memoryless' methods.

Parametric methods: Prediction of traffic flow using linear regression models was deployed in (Low, 1972; Jensen & Nielsen, 1973; Rice & Van Zwet, 2004) while non-linear ones were applied in (Högberg, 1976). ARIMA (Box et al., 2015, ch. 3-5) are parametric linear models extensively used in time series forecasting that incorporate the unobserved (hidden) variables via the MA (moving average) component. Seasonal ARIMA (SARIMA) models can be used where seasonal effects are suspected or when the availability of data is a constraint, according to the work in (Kumar & Vanajakshi, 2015). ARIMAX use additional exogenous data. In (Williams, 2001) data from upstream traffic sensors was used for predicting traffic using ARIMAX models. The results outperformed the simpler ARIMA models at the cost doubling the computational complexity and decreased robustness to missing data. Traffic state estimation with Kalman filters was evaluated on freeways (Xie et al., 2007; Wang et al., 2008) in combination with other methods such as discrete wavelet transforms in order to compensate for noisy data.

Non-parametric methods: One of the first applications of the K Nearest Neighbours (KNN) algorithm for short-term traffic forecasting was in (Clark, 2003). KNNs have also been applied to highway incident detection (Lv et al., 2009). The latter research made use of historical accident data and sensor loop data, representing conditions between normal and hazardous traffic. Hybrid multi-metric k-nearest neighbor regression (HMMKNN) was proposed for multi-source data fusion in (Hong et al., 2015) using upstream and downstream links.

A Support Vector Regression (SVR) model was applied to travel time prediction (Wu et al., 2004) on highways. It was compared only with a naive predictor. Online SVRs with a gaussian kernel were deployed for continuous traffic flow prediction and learning in (Zeng et al., 2008). The method outperformed a simple neural network with one hidden layer. However, it is important to note that the neural network was trained on historical averages, which is not equivalent to training on online streaming data. Under

non-recurring atypical traffic flow conditions, online SVRs have been shown to outperform other methods (Castro-Neto et al., 2009). SVRs with Radial Basis Function (RBF) kernels showed a marginal improvement over Feed-forward Neural Networks (FFNN) (also known as Multilayer Perceptrons - MLP) and exponential smoothing (Asif et al., 2014). The predictions were done independently for each link, thus spatial information was not leveraged. The authors clustered links according to the prediction error using K-means and Self-Organizing Maps (SOM).

Neural networks have been extensively evaluated for short-term real-time traffic prediction (Clark et al., 1993; Blue et al., 1994; Dougherty & Cobbett, 1997; Dia, 2001; Lee et al., 2007; Park et al., 2011; Fusco et al., 2015). A comparison of neural networks and ARIMA in an urban setting found only a slight difference in their performance (Clark et al., 1993). Feed forward neural networks were also used to predict flow, occupancy and speed (Dougherty & Cobbett, 1997). While prediction of flow and occupancy was satisfactory, prediction of speed showed less prospects. In (Park et al., 2011) a neural network was used for simultaneous forecasting at multiple points along a commuter's route. Multiple FFNNs were deployed (one per station) in (Lee et al., 2007) with input data from the same day, the previous week and data from neighbouring links. The weekday was also added as an input in the form of a binary vector. This provided better spatial and temporal context to the network, thus reducing forecasting error. Time information in the form of time of day and day of week was used as additional information for traffic prediction using FFNNs in (Çetiner et al., 2010) also resulting in improved performance. Recurrent neural networks (RNN) demonstrated better forecasting performance (Dia, 2001) at larger prediction horizons compared to FFNNs, mostly due to their ability to model the unobserved variables in a continuous state space. The performance was superior when the data was aggregated into 5 minute bins (90-94%) versus 10 minutes (84%) and 15 minutes (80%). Bayesian networks were also deployed for traffic flow prediction (Castillo et al., 2008). No empirical comparison with other methods was provided.

It is important to consider that as the number of parameters increase with road network complexity and size, the parsimony of non-parametric methods becomes more evident (Domingos, 2012).

Hybrid Methods: Existing short-term traffic forecasting systems were reviewed under a Probabilistic Graphical Model (PGM) framework in (Lippi et al., 2013). The authors also propose coupling SARIMA with either SVRs

(best under congestion) and Kalman filters (best overall). This work assumes statistical independence of links. Hybrid ARIMA and FFNNs (Zhang, 2003) were also applied to univariate time series forecasting. The residuals of a suboptimal ARIMA model were used as training data for a FFNN. No evaluations on traffic flow data was provided. A hybrid SARIMA and cell transmission model for multivariate traffic prediction was evaluated in (Szeto et al., 2009). Comparisons with other univariate or multivariate models were not provided. Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and ARIMA were combined in (Chen et al., 2011). The hybrid model did not show any advantages over the standard ARIMA, although the authors argued that the method captured the traffic characteristics more comprehensively.

An online adaptive Kalman filter was combined with a FFNN via a fuzzy rule based system (FRBS) (Stathopoulos et al., 2008) where the FRBS parameters were optimized using Meta heuristics. The combined forecasts were better than the separate models. Functional nonparametric regression (FNR) was coupled with functional data analysis (FDA) in (Su et al., 2016) for long term traffic forecasting on one day and one week ahead horizons. Traffic state vectors were selected based on lag autocorrelation and used as predictor data for various types of kernels. The distance function for the kernels was computed using functional principal component analysis. The method outperformed SARIMA, FFNNs and SVRs on the selected benchmark, based on a subset of weekdays (Monday, Wednesday, Friday and Saturday) for a single expressway section. Genetic algorithms (GA) were used in (Abdulhai et al., 2002; Vlahogianni et al., 2005) to optimize neural network architecture structure. In (Abdulhai et al., 2002) it was applied to the structure of time-delayed neural network while in (Vlahogianni et al., 2005) the GAs were used to optimize the number of units in the hidden layer. The optimised version reached the same performance as a predefined one with less neurons.

2.2. Topology and spatio-temporal correlations in traffic prediction

There are various methods that model the spatial domain as opposed to solely the temporal one. The approaches can be characterised based on the number of timeseries used as inputs and outputs for a prediction model. Regarding the inputs, the models can be either single-series (univariate) or multi-series (multivariable). In the latter case additional contemporary data can be used and the selection can be based on either topology (if known) or learned. Further categorization can be defined based on the importance

of each relevant road – static or dynamic (since the dependency structure changes – see Fig. 10). According to the outputs, the algorithms can be single task, in which case one predictor is learned for each station and multi-task in which case parameters are coupled and predictions are made simultaneously at all stations.

Single task & multi-series: Most common among the multi-series methods is to take into consideration the upstream highway links. Kalman filters using additional data from upstream sensors have been found to be superior to simple univariate ARIMA models (Stathopoulos & Karlaftis, 2003). Data from only five sequential locations along a major 3 lane per direction arterial on the periphery was used (distance between sensors varied between 250 and 2500 meters). The authors also conclude that short-term traffic flow prediction at urban arterials is more challenging than on freeways. A similar conclusion is drawn in (Vlahogianni et al., 2005).

Markov random fields (MRF) were deployed to capture dependencies between adjacent sensor loops via a heat map of the spatio-temporal domain (Ko et al., 2016). Focus was on the dependencies between the query location and the first and second upstream link connections on freeways with degree higher than one. Such upstream bifurcations were referred to as 'cones'. Data between the 'cones' and the query was not considered. The authors quantized data into twelve levels. The weights for the spatial parameters were reestimated monthly. In more complex networks such as urban areas, these weights can change during the course of one day. Similarly, dependencies on the cliques are estimated in (Ahn et al., 2015) using either multiple linear regression and SVRs, the latter resulting in better accuracy. No comparisons were made with other methods. Multivariate Adaptive Regression Splines (MARS) were used for selecting the relevant links' data and SVRs as the predictor component in (Xu et al., 2015). The method was evaluated on a subarea and compared to AR, MARS, SVRs, SARIMA and ST-BMARS (spatio-temporal Bayesian MARS) showing promising results. An interesting approach was the work in (Mitrovic et al., 2015) where the core idea was to minimize execution time by transferring learned representations (on a subset of links) to the overall network. A low dimensional network representation was learned via matrix decomposition and this subset of links was used to extrapolate predictions over the entire network. The computations were sped up 10 times at the expense of increasing error by 3%.

Spatio-temporal random effects (STRE) was proposed in (Wu et al., 2016). The algorithm reduced computational complexity over spatio-temporal

kalman filter (STKF). Data around a query point was selected and weighted using a fixed precomputed set of rules, depending on the relative position (e.g. upstream, downstream, perpendicular, opposite lane, etc). Training was done using 5 minute resolution data from Tuesdays, Wednesdays and Thursdays. Only data from 6a.m. to 9p.m. (peak hour) was considered. Predictions were made on a group of query points from two separate areas (mall area and non mall area). The authors hypothesized that the mall area could have had more chaotic traffic patterns. STRE had lower error relative to ARIMA, STARIMA and FFNNs except for one case, the westbound non-mall area. It is important to note that the FFNNs were trained in univariate, one networok per station mode. Furthermore, it is also not clear whether the prediction results were for out of sample data. Similar work making use of sensor location data, spatio-temporal random fields (STRF) followed by Gaussian process regressors were proposed in (Liebig et al., 2016) for route planning.

FFNNs and Radial Basis Function Networks (RBFNs) were combined into the Bayesian Combined Neural Network (BCNN) (Zheng et al., 2006) model for traffic flow prediction on a 15 minute resolution highway dataset. Data from the immediate and two links upstream sensors as well as downstream sensors was used as input to both networks. The predictions were combined linearly and weighted according to the spread of the error in previous time steps on all relevant links. The combined model was better than the individual predictors, however the RMSE was not reported nor any comparisons with other baseline methods given. Bayesian networks with SARIMA as an a priori estimator (BN-SARIMA) and FFNNs as well as Nonlinear AutoRegressive neural network with eXogenous inputs (NARX) were compared in (Fusco et al., 2015) for floating car data. The learning architectures used data from the output and conditioning links and predictions were single task. The results were marginally different for a subsection of reliable data. For network-wide forecasting on both 5 and 15 minute intervals, NARX and FFNNs were better than BN-SARIMA.

Multi-task & multi-series: Spatio-Temporal ARIMA (STARIMA) (Kamarianakis & Prastacos, 2005) were perhaps the first successful traffic forecasting models that focused on the spatio-temporal correlation structure. However, the spatial correlations were fixed, depending solely on the distances between links. We empirically show in Figure 10 (Page 29) that the spatial correlations can change during the course of a day. STARIMA compensate for non-stationarity by differencing using the previous day values

which can bias the estimated autoregression parameters (traffic behaviour can be considerably different e.g. Monday vs. Sunday). A study on autocorrelation on spatio-temporal data (Cheng et al., 2012) concludes that ARIMA based models assume a globally stationary spatio-temporal autocorrelation structure and thus are insufficient at capturing the changing importance between prediction tasks. The work in (Min et al., 2009) addresses this problem using Dynamic Turn Ratio Prediction (DTRP) to update the normally static matrix containing the structural information of the road network. Under the hypothesis that the physical distance between road sections (tasks) does not accurately describe the importance of each task, a VARMA (Lütkepohl, 2005) model is refined by Min & Wynter (2011) to include dependency among observations from neighbouring locations by means of several spatial correlation matrices (as many as the number of lags). In one matrix, only related tasks are non-zero. The authors do not explicitly define task relatedness, however most likely all upstream connections are selected. Further sparsity is introduced by removing upstream links, under the hypothesis that such links are unlikely to influence traffic at the query task, given the average travel speed for that location and time of day. There could be one such matrix for peak times and one for off-peak times, depending on the design choice. Another ARIMA inspired algorithm (Kamarianakis et al., 2012) makes use of a parametric, space-time autoregressive threshold algorithm for forecasting velocity. The equations are independent and incorporate the MA (moving average) and a neighbourhood component that adds information from sensors in close proximity, based on the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996).

Building up on the previously mentioned work and avoiding the potentially problematic thresholds used to to identify the discrete set of regimes, a Graph Based Lag STARIMA (GBLS) (Salamanis et al., 2016) model was trained using speed data from GPS sensors, with the goal of travel time prediction. Unlike the previous work, the graph structure was initially refined using a breadth-first search based on degree (number of hops). For the selected connections, spatial weights were computed and used in the STARIMA model. The weight matrix for each lag was fixed and contained the inverse of the lag-sums Pearson correlations between relevant roads. Finally, the model took into consideration the current speed on the road, the previous two speed values and the average speed of the top 10 ranked relevant roads. The introduced sparsity reduced computational complexity. However, from our own experiments we have observed that the correlations are typically nonlinear,

which implies that this measure is not appropriate for ranking. The data used was only for the same day of the week for both training and testing for the two datasets. GBLS was compared with univariate KNNs, Random Forests (RF), SVRs and Compressed SVRs. No comparison with FFNNs were made. The behaviour of the proposed algorithm was very different over the two datasets, however the proposed method had lower prediction error then the benchmarks. It is also not clear whether the parameters were coupled or not. One could argue that this class of methods do not follow the law of parsimony (Occam‘s razor) – there are too many design choices, assumptions and parameters. Recently FFNNs with many hidden layers (deep learning) have been applied to network wide traffic prediction on highway data (Lv et al., 2015). While neural network based models were able to learn the nonlinear spatio-temporal correlations, this type of approach – as well as the similar linear STARIMA class of models (Kamarianakis et al., 2012; Min & Wynter, 2011; Min et al., 2009; Kamarianakis & Prastacos, 2005) – does not explicitly leverage the topological structure. Hence it is likely that prediction error can be further reduced by leveraging this information explicitly. Conversely to the aforementioned work, in Topological Vector Autoregression (TRU-VAR), the relative importance of the related timeseries is adjusted automatically, also accounting for the contextual time.

In summary, the following observations can help improve traffic prediction performance: (**1**) nonlinearity is important for explaining traffic behaviour; (**2**) leveraging the topological structure could result in lower errors; (**3**) the system should be flexible to changes in the adjacency matrix design; (**4**) static spatio-temporal models are not appropriate for complex urban roads; (**5**) simplicity (Occam's razor) is to be preferred in general (Domingos, 2012); (**6**) multi-metric data can increase accuracy, speed data is to be avoided as a target metric (Clark, 2003); (**7**) for certain models, explicitly encoding time can decrease error; (**8**) crowdsourced data from vehicles can help reduce sensor noise and provide redundancy to missing data; (**9**) continuous state-space models are more adequate at capturing complex patterns and therefore making predictions due to the highly dynamic nature of driver behaviour.

Considering all of the above, in the next section we introduce the theoretical underpinnings of TRU-VAR. We start with the extension of VAR to topological constraints, continue the generalization with examples of function approximators according to state of the art prediction models and conclude with the fitting process.

3. Topological vector autoregression with universal function approximators

Traffic prediction in large urban areas can be formulated as a multivariate timeseries forecasting problem. Vector AutoRegression (VAR) is a natural choice for such problems. Since in this particular case we are also provided with the precise location of each sensor station where volume (flow) recordings are made, we leverage the topological structure and use it as prior information in order to constrain the number of parameters that are to be estimated. In effect this reduces the computational complexity and improves accuracy over simple univariate forecasting methods, while learning in real time the spatio-temporal correlations between the contemporary timeseries.

From a high level perspective our idea is simple: we assume that for a particular prediction station (timeseries) it is beneficial to use data from stations in close proximity (contemporary), however we exclude stations that are distant. This induces sparsity since we do not have to use the data from all other sensor stations, while capturing the spatio-temporal dynamics. We provide empirical arguments in an exploratory analysis of what the distance heuristics can be defined as in Section 4 where we also show that the spatio-temporal correlations change throughout the day.

Given a multivariate timeseries dataset $\tilde{X} \in \mathbb{R}^{T \times S}$ with T non i.i.d. observations over S series, we denote $\boldsymbol{x}^s_{\Delta t} = \begin{bmatrix} 1 & x^s_t & x^s_{t-1} & \dots & x^s_{t-\Delta} \end{bmatrix}^\top$ as the predictor vector for sensor s consisting of the past observations of length Δ. The corresponding response variable $y^s_{t+h} = x^s_{t+h}$ is the value to be predicted h time steps ahead. Prediction can then be modelled as follows:

$$y^s_{t+h} = f(\boldsymbol{x}^s_{\Delta t}, \boldsymbol{\theta}^s) + \epsilon^s_{t+h} \quad \text{where } \epsilon \in \mathcal{N}(0, \sigma) \tag{1}$$

The prediction horizon h indicates how far in the future predictions are made. For short-term forecasting this is typically the immediate time step (e.g. $h = 1$) in the future. The first element of the input vectors is set to 1 to indicate the intercept constant. In the time series forecasting literature Δ is more commonly known as the *lag* or the number of previous observations from the past that are used to make predictions. For neural networks this is sometimes referred to as the receptive field. To satisfy the assumption that $\mu_\epsilon = 0$ and normally distributed, we later discuss differencing. If f is the dot product, then Eq. 1 describes an autoregressive model (AR(Δ)) of order Δ. AR models operate under the assumption that the errors are not autocorrelated. However, the errors are still unbiased even if they are

autocorrelated. ARIMA models also incorporate a Moving average (MA). These are identical with the difference that predictions are made based on the past prediction errors $\boldsymbol{e}^s_{\Delta t}$ where $e^s = \hat{y}^s - y^s$. It is possible to write any stationary AR(Δ) model as an MA(∞) model.

3.1. Topology Regularized Universal Vector Autoregression

In multivariable regression models, additional data can be used for the predictor data such as contemporary data and / or an encoding of the time of day $\boldsymbol{z}_{\Delta t}$. Any other source of relevant data, such as weather data, can also be used as input.

Vector AutoRegression (VAR) models (Hamilton, 1994, ch. 11) are a generalization of univariate autoregressive models for forecasting multiple contemporary timeseries. We refer the reader to (Athanasopoulos et al., 2012) for a discussion on VARs in contrast to VARMAs. VARs are not a closed class when the data is aggregated over time like VARMAs are. However, VARs have been found to be often good approximations to VARMAs, provided that the lag is sufficiently large (Lütkepohl, 2005). We start with a two dimensional VAR(1) with one lag, where $\boldsymbol{\theta}$ (Eq. 1) is the vector containing the autoregressive parameters consisting of θ^{ij}_t which weight the external spatio-temporal autoregressive variables, specifically the influence of the t-th lag of series x^j on series x^i:

$$\begin{aligned} y^1_{t+h} &= \theta^1_0 + \theta^{11}_t x^1_t + \theta^{12}_t x^2_t + \ldots + \theta^{1S}_t x^S_t + \epsilon^1_{t+h} \\ y^2_{t+h} &= \theta^2_0 + \theta^{21}_t x^1_t + \theta^{22}_t x^2_t + \ldots + \theta^{2S}_t x^S_t + \epsilon^2_{t+h} \end{aligned} \tag{2}$$

It becomes evident from the above equations that the required number of parameters to be estimated increases quadratically with the number of timeseries. Furthermore, if more lags are added, the computational complexity becomes $O((s \times l)^2)$. Therefore, introducing sparsity is both beneficial (Eq. 3) and necessary since it introduces constraints on the number parameters that are to be estimated. Sparsity refers to reducing the number contemporary time series (and therefore parameters to be estimated) that are relevant for a query location. Since the topological structure of road networks is known, introducing such priors is intuitive since the relevance of contemporary timeseries does vary. We therefore introduce sparse topology regularization for vector autoregression (TRU-VAR) based on the road geometry via a topology-designed adjacency matrix $A \in \{0, 1\}$. If G is the

graph describing road connections, then we denote $G^1(i)$ to be the set of all first order graph connections with node i, then:

$$a^{ij} = \begin{cases} 1, & i = j; \\ 0, & j \notin G^1(i); \\ 1, & j \in G^1(i). \end{cases}$$

Evidently, we could have also opted to select higher order degrees e.g. $G^2(i)$, for designing the topological adjacency matrix, however in urban settings the number of such neighbours can increase exponentially as the degree increases. However, the matrix could be heuristically adapted according to the number of first order connections, in the case of highways where the degree of a node can be low. Other means of defining A can be used such as ranking based on correlation of the timeseries. However, we show in Figure 9 (plotted using Google Maps) that correlation does not necessarily correspond to the relevant topology. In contrast, in our case the relevant importances are learned through fitting the model parameters, thus capturing the spatio-temporal dynamics. We introduce the sparsity terms and write in general form:

$$y^s_{t+h} = \theta^s_0 + \sum_{k=1}^{S} a^{sk}\, \theta^{sk}_t\, x^k_t + \epsilon^s_{t+h} \tag{3}$$

We can then generalize the previous equation for one series to multiple lags:

$$y^s_{t+h} = \theta^s_0 + \sum_{k=1}^{S} a^{sk} \sum_{\delta=0}^{\Delta} \theta^{sk}_{t-\delta}\, x^k_{t-\delta} + \epsilon^s_{t+h} \tag{4}$$

Finally we write in compact form and generalize for any function approximator:

$$\begin{aligned} y^s_{t+h} &= \boldsymbol{\theta}^{s\top} \boldsymbol{x}^s_{\Delta t} \odot [(\boldsymbol{a}^s)_{\times\Delta}] + \epsilon^s_{t+h} \\ y^s_{t+h} &= f\left(\boldsymbol{x}^s_{\Delta t} \odot [(\boldsymbol{a}^s)_{\times\Delta}], \boldsymbol{\theta}^s\right) + \epsilon^s_{t+h} \end{aligned} \tag{5}$$

For convenience, we can stack the inputs (predictor data) into a matrix with $N = T - \Delta$ rows and $M = \Delta$ columns $X^s \in \mathbb{R}^{N \times M}$ and the outputs

(response data) into the vector $\boldsymbol{y}^s \in \mathbb{R}^N$. We denote the lag constructed matrices for all series as $X = \begin{bmatrix} X^1 & X^2 & \dots & X^S \end{bmatrix}$ and $Y = \begin{bmatrix} \boldsymbol{y}^1 & \boldsymbol{y}^2 & \dots & \boldsymbol{y}^S \end{bmatrix}$. Finally, we define $A^s = \begin{bmatrix} (\boldsymbol{a}^s)_{\times\Delta} \end{bmatrix}$ and $A = \begin{bmatrix} A^1 & A^2 & \dots & A^S \end{bmatrix}$.

$$\boldsymbol{y}^s = f\left(X^s \odot A^s, \boldsymbol{\theta}^s\right) + \boldsymbol{\epsilon}^s \tag{6}$$

Note that from here on the Hadamard product with the sparsity inducing matrix A is omitted for brevity.

3.2. Structural Risk Minimization

We assume that there is a joint probability distribution $P(x, y)$ over X^s and $\boldsymbol{y}^s$ which allows us to model uncertainty in predictions. The risk cannot be computed in general since the distribution $P(x, y)$ is unknown. The empirical risk is an approximation computed by averaging the loss function L on the training set. In Structural Risk Minimization (SRM) (Vapnik, 1999) a penalty Ω is added. Then, the learning algorithm defined by SRM consists in solving an optimization problem where the goal is to find the optimal fitting parameters $\hat{\boldsymbol{\theta}}^s$ that minimize the following objective:

$$\hat{\boldsymbol{\theta}}^s = \underset{\theta^s}{\operatorname{argmin}} \left(L\big(f(X^s, \boldsymbol{\theta}^s), \boldsymbol{y}^s\big) + \lambda^s \Omega(\boldsymbol{\theta}^s) \right) \tag{7}$$

For regression, under the assumption of normally distributed errors, the mean squared error (MSE) or quadratic loss is commonly used since it is symmetric and continuous. Thus, in least squares minimizing the MSE (Eq. 8) results in minimizing the variance of an unbiased estimator. As opposed to other applications, in our problem setting it is desirable to put a heavier weight on larger errors since this maximizes the information gain for the learner (Chai & Draxler, 2014). We take this opportunity to mention that we report the RMSE since it is linked to the loss function, however we also report the MAPE which is specific to traffic forecasting literature. For a comprehensive discussion on predictive accuracy measures we refer the reader to (Diebold & Mariano, 2012).

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{N} \sum_{n=1}^{N} (\hat{y}_n - y_n)^2 = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2 \tag{8}$$

3.3. Regularized Least Squares

Since our data consists of previous observations in time it is possible that these could have a varying degree of importance to our predictions, and furthermore these could be different for each data stream. Using the quadratic error we arrive at the least squares formulation in Eq. 9 where Ω is a penalty on the complexity of the loss function which places bounds on the vector space norm. Since the choice of the model can be arbitrary, we can use regularization to improve the generalization error of the learned model. With a lack of bounds on the real error these parameters are tuned using the surrogate error from a validation dataset or using cross-validation.

For a negative log likelihood loss function, the Maximum a Posteriori (MAP) solution to linear regression leads to regularized solutions, where the prior distribution acts as the regularizer. Thus, a Gaussian prior on $\boldsymbol{\theta}$ regularizes the L_2 norm of $\boldsymbol{\theta}$ while a Laplace prior regularizes the L_1 norm (see Murphy, 2012, ch. 5-9). ElasticNet (Zou & Hastie, 2005) is a combination of L_1 and L_2 regularization which enforces sparsity on groups of columns such as the ones we have in X where a parameter α balances the two lambdas of the two norms. In effect, ElasticNet generalizes both L_1 and L_2 regularization. We experiment with both regularization methods.

$$\sum_{s=1}^{S} \min_{\theta^1,\ldots,\theta^S} \|f(X^s, \boldsymbol{\theta}) - \boldsymbol{y}\|^2 + \lambda_2 \|\boldsymbol{\theta}\|^2 + \lambda_1 \|\boldsymbol{\theta}\|_1 \tag{9}$$

3.4. The function approximator model

Having defined the optimization problem, loss function and regularization prior types for one timeseries we can now consider the function approximation model f^s, starting with the simple linear models.

Linear least squares. If the combination of features is linear in the parameter space $\boldsymbol{\theta}$, then the regression model is linear. Typically the error ϵ is assumed to be normally distributed. Any non-linear transformation $\phi(\boldsymbol{x})$ of the input data X such as a polynomial combination of features thus still results in a linear model. In our experiments we use a simple linear combination of features ($\phi(x) = x$) for linear least squares (LLS):

$$f(X, \boldsymbol{\theta}) = \sum_{m=1}^{M} \boldsymbol{\theta}_m \phi(x)_m \tag{10}$$

By replacing f with Eq. 10 in Eq. 9, and setting $\lambda_1 = 0$, the parameters $\boldsymbol{\theta}^s$ can be found analytically for ridge regression using the normal equation:

$$\hat{\boldsymbol{\theta}} = (X^\top X + \lambda_2 I)^{-1} X^\top \boldsymbol{y} \tag{11}$$

For ordinary least squares (OLS) where regularization is absent we can simply set $\lambda_2 = 0$. It is important to note that the solution is only well defined if the columns of X are linearly independent e.g. X has full column rank and $(X^\top X)^{-1}$ exists. Furthermore, closed form solutions (offline / batch learning) are more accurate than online learning since if the solution exists we are guaranteed to converge to the global optimum, while for online methods there is no such guarantee. In real world scenarios however, data comes in streams and offline learning is not a practical option. Real-time learning is essential to incident prediction (Moreira-Matias & Alesiani, 2015). Then, $\boldsymbol{\theta}$ can be transferred to an online learner as an initialization where the learning can continue using an online method such as stochastic gradient descent (SGD).

Kernel least squares. Kernel methods such as support vector regression (SVR) (Smola & Vapnik, 1997; Suykens & Vandewalle, 1999) rely precisely on non-linear transformations of the input data, usually into higher dimensional reproducing kernel Hilbert space (RKHS), where the transformed data $\phi(\boldsymbol{x})$ allows for lower generalization error by finding a better model hypothesis. In our experiments we use a linear kernel for the support vector regression (SVR) or kernel least squares (KLS).

$$\phi : K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle \tag{12}$$

Using the kernel the model can be expressed as a kernel expansion where $\alpha(n) \in \mathbb{R}$ are the expansion coefficients. This transforms the model to:

$$f(\boldsymbol{x}, K) = \sum_{n=1}^{N} \alpha(n) K(x_n, x) \tag{13}$$

Which in turn can be formulated as kernel ridge regression:

$$\phi(\hat{\boldsymbol{\theta}}) = \underset{\phi(\theta)}{\operatorname{argmin}} \|\boldsymbol{y} - \phi(X)\phi(\boldsymbol{\theta})\|^2 + \lambda \|\phi(\boldsymbol{\theta})\|^2 \tag{14}$$

In practice this implies computing the dot product of the transformed input matrix which is very memory intensive. Instead, in our experiments

we use L-BFGS (Nocedal & Wright, 2006) for ridge or SPARSA (Wright et al., 2009) for LASSO.

Nonlinear least squares. Multi layer perceptrons with one hidden layer are a form of non-linear regression. Such models with a finite set of hidden sigmoid activation functions are universal function approximators (Cybenko, 1989). The simplest form can be defined using a single hidden layer model with H units, using a sigmoid activation function $\phi(x) = 1/(1 + e^{-x})$ The model is reminiscent of nested kernels:

$$f(X, \Theta) = \sum_{h=1}^{H} \boldsymbol{\theta}_h \phi\Big(\sum_{m=1}^{M} \boldsymbol{\theta}_m \phi(x)_m \Big) \tag{15}$$

For non-linear least squares there is no closed form solution since the derivatives are functions of both the independent variable and the parameters. Such problems can be solved using gradient descent. After specifying initial values for $\boldsymbol{\theta}$ (which can also come from an offline learner), the parameters are found iteratively through successive approximation. Multiple passes are done through the dataset. In one pass, mini-batches or single examples are shown and the parameters are adjusted slightly (according to a learning rate) in order to minimize the loss.

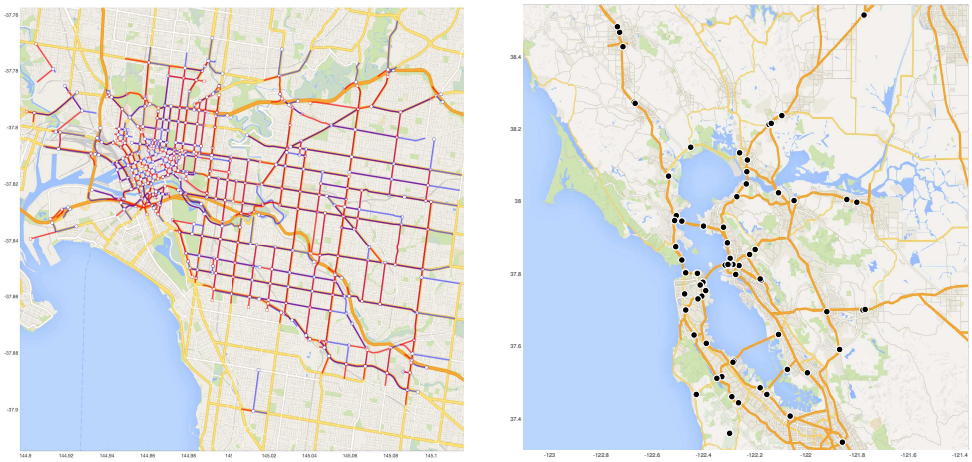
In conclusion we note that once the TDAM is defined, TRU-VAR can therefore be generalized to any type of function approximator. The modularity and flexibility results in low computational complexity thus resulting in scalable fitting of nonlinear models which capture the spatio-temporal correlations. This also provides redundance and versatility towards a broad set of road network types covering urban, suburban and freeways. Finally when the network structure changes the TDAM can be adjusted partially, which does not require redeployment of the entire prediction system.

4. An exploratory analysis of traffic data

In this section we discuss data. We make observations and apply the findings in the experimental section. We consider aspects such as trends, seasonality, outliers, stationarity, variable (sensor station) interdependency and data quality. We show that there are specific dependencies between sensors (Fig. 8) which also change as a function of time (Fig. 10).

4.1. Datasets

The VicRoads dataset (Schimbinschi et al., 2015) was recorded over 6 years in the City of Melbourne, Australia and consists of volume readings from 1084 sensors covering both urban and suburban areas as well as freeways. The frequency of recordings is 15 minutes (96 readings over 24 hours). Coverage area depicted in Figure 1a.



(a) Melbourne roads with available traffic data are highlighted in either red or blue according to direction.

(b) PeMS station points are marked with a dot.

Figure 1: Schematic illustration of the sensor location for both datasets.

The California Freeway Performance Measurement System (PeMS) dataset (Varaiya, 2001) (Fig. 1b) has been extensively used in the literature and consists of volume readings taken every 5 minutes from a network of 117 sensors over 8 months in the year 2013. As it can be seen from Figure 1b it consists of mostly freeway commuting and thus does not capture the complexities of inner city commuting.

We further describe the two datasets with an emphasis on VicRoads, since it is a new dataset and PeMS is well known and more studied in the literature (Lv et al., 2015; Lippi et al., 2013, and others).

4.2. VicRoads data quality and congestion events

Datasets recording traffic volume (flow) consist of positive integers in original format. There is no explicit distinction made (no lables) between congestion and missing data. Intuitively, volume is close to zero when traffic slows down and zero when it completely stops. While it is certainly possible to have zero volume in a period of 5 or 15 minutes, it is highly unlikely that this can happen in practice and very unlikely to happen for an entire day. We proceed to mark the missing data by assuming that if the total volume for one day is zero (no cars have passed through the sensor in 24 hours) then it does not correspond to a congestion event.

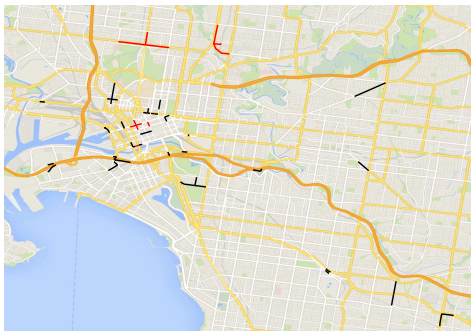


Figure 2: Sensors above the 95 (black) and 99 (red) quantile after discarding missing data, road sections with many congestion events.

We therefore use this information to discard the marked days *only from the test set*. This is important, since it allows the learner to identify congestions. Considering that in real scenarios these sensors can break down we aim to emphasize robustness towards such events or congestion and as such do *not* replace the missing values with averages or use any other method of inferring the missing values.

Following this operation we can observe (Fig. 3) that these sensors have not all been installed at the same time. Furthermore, we can also observe network-wide missing data (vertical lines). We do not take into consideration sporadic 0 readings since these could correspond to sudden traffic congestion, although it is still highly unlikely. However, we could have considered heuristic rules where for example, 4 consecutive readings would correspond to sensor failure (or road works) since it is very unlikely that no cars would pass within one hour through a section, even in congestion conditions. We have not taken this approach in the current work.

After marking the missing data, we further compute the remaining number of 0-valued recordings for each sensor and plot in Figure 2 the roads above the 0.95 (black) and 0.99 (red) quantile on the map. It is quite probable that the error on these particular sensors will be higher than others, since half the data can be available for these sensor stations.

4.3. Seasonality, Trends, Cycles and Dependencies

It is trivial to relate to daily traffic patterns, especially peak hours. These patters are mostly stable and are a function of location and day of the week. These flows vary along the different spatio-temporal seasonalities of drivers behaviour. We adopt an initial explorative approach towards determining the statistical properties of the multivariate timeseries. We did not perform a Box-Cox transformation since upon inspection there was no evidence of changing variance. Furthermore, upon visual inspection of the series it was evident that the data is non-stationary as the volume moves up and down as a function of the time of the day.

Figure 4 depicts the summary statistics plotted per time bin in one day, where the left figure corresponds to a typical suburban region while the right one corresponds to a highway off-ramp. The right one is typical for a road with high outbound traffic in the evening peak hours when people commute back home. Comparing these two locations allows us to get insights towards the dynamics of the different types of traffic within the network. Their location is also shown on the map in Figure 7.

4.4. Autocorrelation profiles

Figure 5 depicts the autocorrelation profiles of the corresponding traffic flow time series for 2 different sensors from the VicRoads dataset. Their

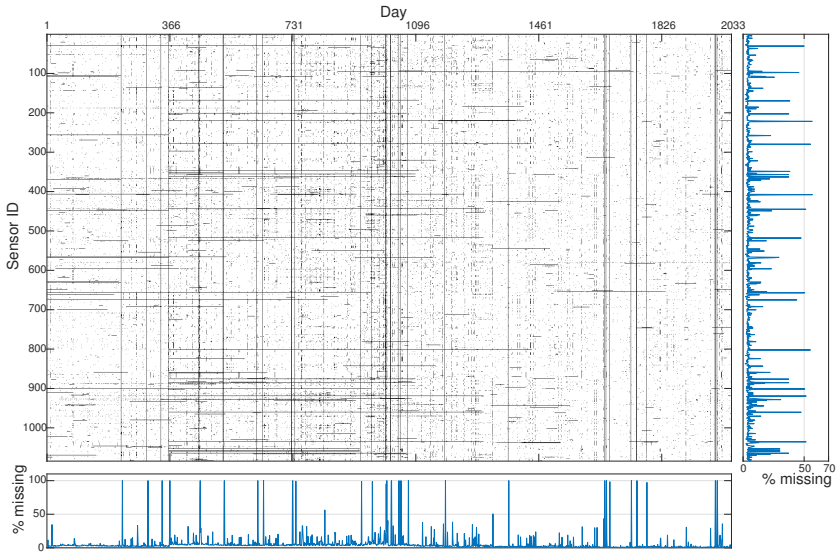
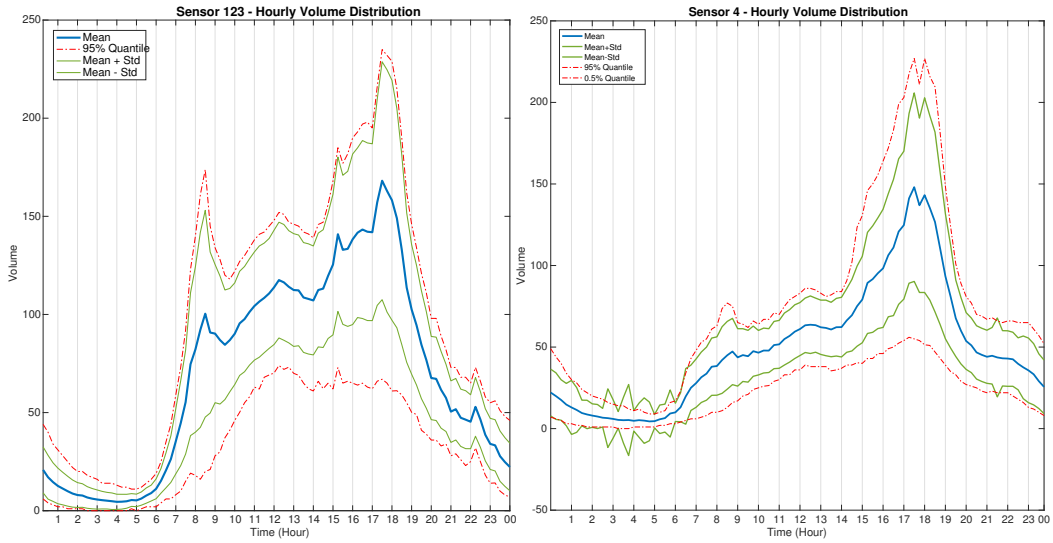


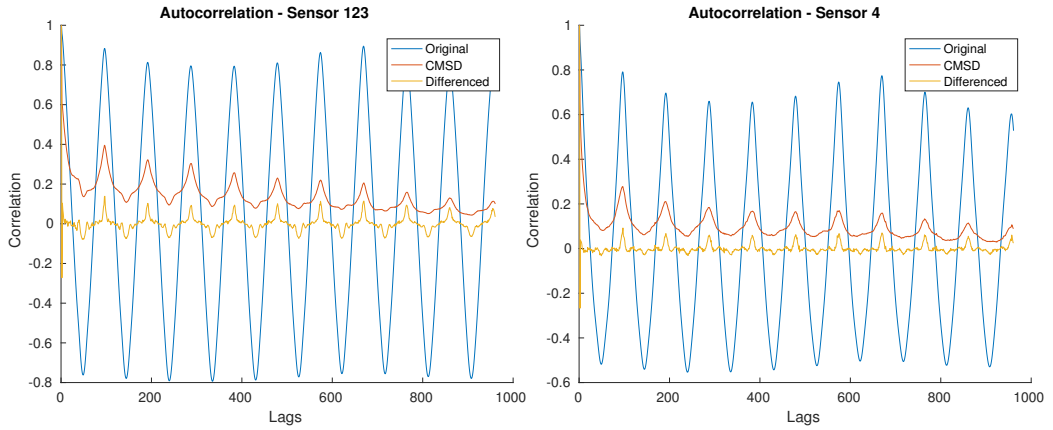
Figure 3: Black pixels indicate days with no readings.



(a) Sensor 123 **(b)** Sensor 4

Figure 4: The daily summary statistics differ for each road segment (VicRoads).

location on the map is shown in Figure 7. We chose a lag corresponding to 4 days to observe any seasonal or cyclic patterns left in the signal, after subtracting the seasonal component or differencing.



(a) Sensor 123 - Suburb **(b)** Sensor 4 - Highway off ramp

Figure 5: Autocorrelation plot for 400 lags. Differencing removes seasonality patterns. Daily seasonality is clearly observable.

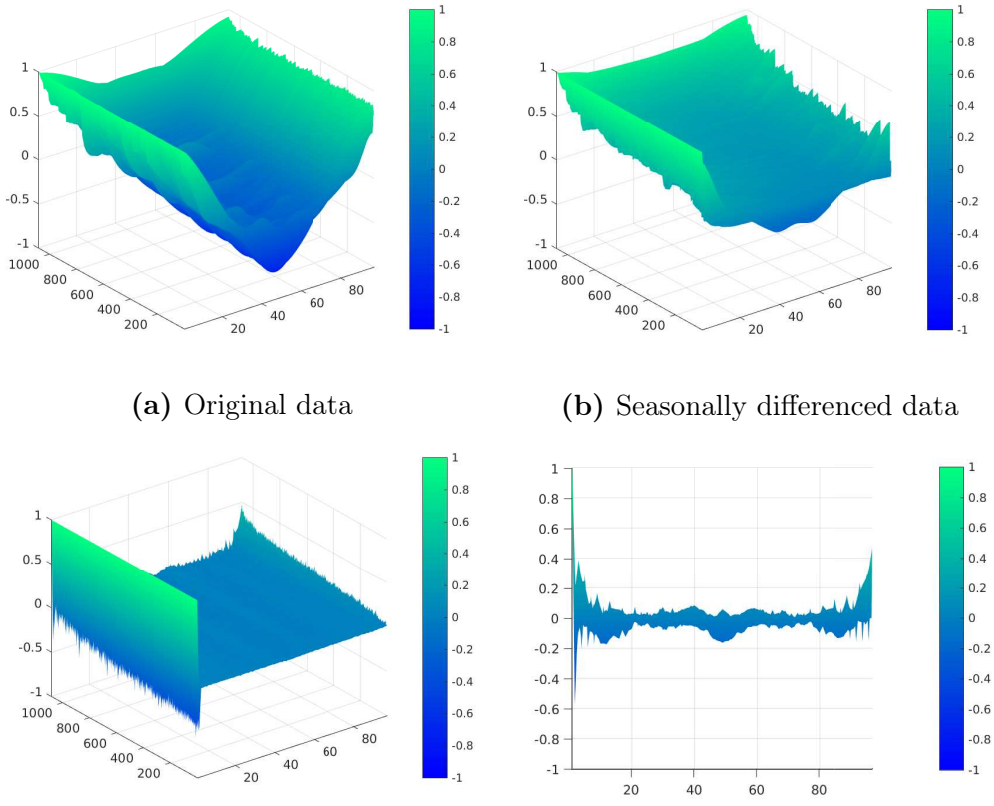
We estimate the seasonal component for each series by computing the minute / hourly averages, separately for each day of the week, for each series. In order to compute the averages we convert the time series to a tensor matrix. $X_\mu \in \mathbb{R}^{S \times D \times H}$ where $S = 1084$ is the number of road segments $D = 7$ is the number of days in a week and $H = 96$ is the 15 minute average number of observations made in one day. We then convert the matrix back to a timeseries and we subtract this from the original time series, obtaining the contextual mean seasonally differenced (CMSD) timeseries. This operation is normally performed using a moving average. However, this way we can arrive at a more precise estimation, also as a function of the day of the week. We also differentiate each sensor‘s timeseries and plot the autocorrelation again for these two sensors. This method removes most of the seasonal patterns in the data.

It can be observed from Figure 5 that while these operations largely removes the seasonality, it is quite likely that the seasonal or trend / cyclic components could come from the (non) linear interactions with the neighbouring roads. This suggests that using proximity data is more likely to lead to increased accuracy. We further inspect the overall autocorrelation over the entire network. Hence we plot the autocorrelation for all sensors for 96 lags on the same type of data transformations. It is observable from Figure 6d that the seasonal components are removed for almost the entire network data.

4.5. Intersection correlations

While we have considered the statistical properties of each individual road section we further examine the information carried between road sections, since most roads are highly dependent on the connected or neighbouring roads. In Figure 7 the roads with the same direction as the query road (solid black) are marked as black and the opposite direction are marked as red dashed lines. It is important to point out that this might not be entirely accurate and depends on the convention used when marking direction (e.g. 1 is always road heading north or west).

From Figure 8a we can observe high correlation between the red dotted segments and the black target road (123) - implying a correlation between (apparently) opposite traffic directions, which is unlikely. For prediction, this is not crucial since we select all connected sections. However, our major point is that there are complex dependencies at intersections. It is important



(a) Original data

(b) Seasonally differenced data

(c) Differenced data

(d) Differenced data - side view

Figure 6: Network Wide Autocorrelation Surface.

to note that Pearson's correlation coefficient captures only linear correlations and there could be nonlinear correlations between two road sections.

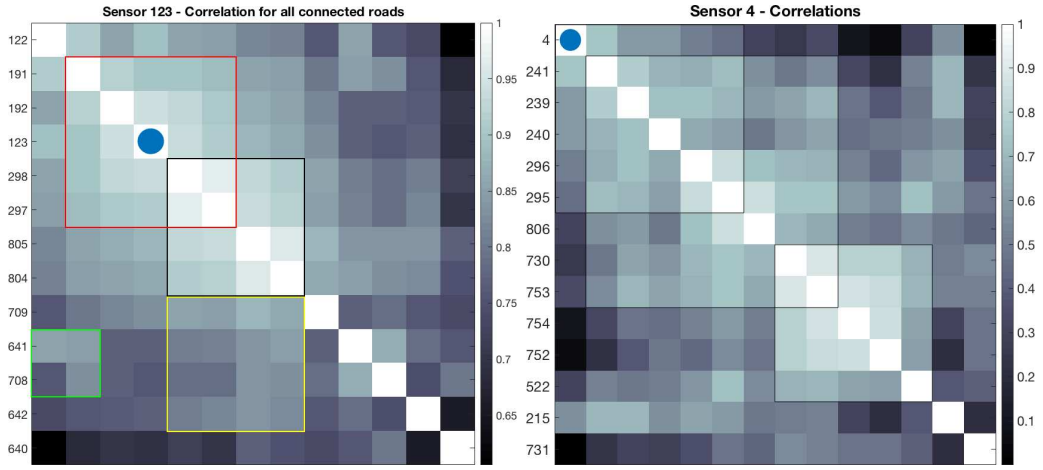
From Figure 8a we can observe that a cluster (red square) is formed with the perpendicular query road (123) at both ends, namely 191 and 192 at one end and 298 and 297 at the other end. While these consecutive sections are correlated with themselves. We do not know at which end of the road section the sensors are placed unfortunately for section 123 or other sections, hence we can not even attempt to make causality assumptions.

We can further observe that in turn these last two form a *cluster* with the roads on the opposite traffic (black square). The yellow circle depicts the correlations on the other side of the road and their correlation with the parallel side of the road. The green square shows that the opposite direction section (641) from the query road is more correlated with traffic from the top,

(a) Sensor 123 - directly connected roads

(b) Sensor 4 - nearest via euclidean distance

Figure 7: Road sections are marked for either direction (unreliable).



(a) Sensor 123

(b) Sensor 4

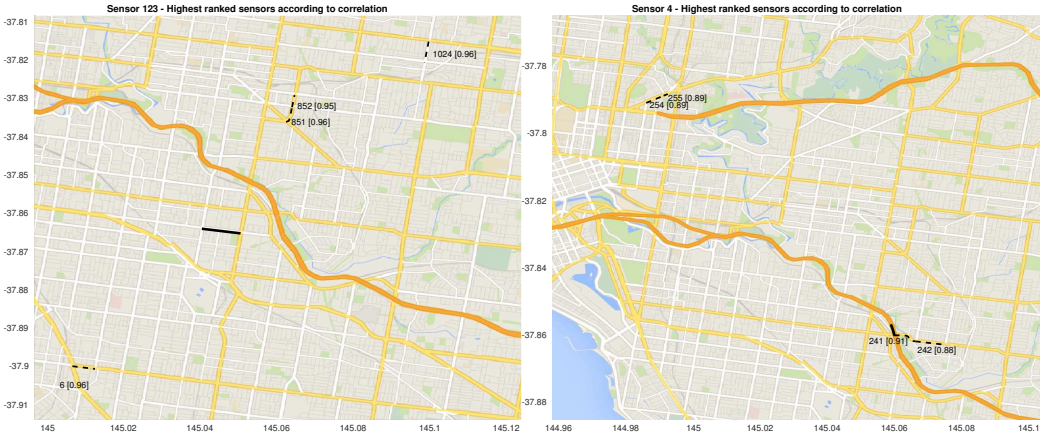
Figure 8: Correlation matrix for two different road sections

bottom and left side. Upon even more careful inspection, these correlations reveal the behaviour of traffic despite the fact that we are unaware of the actual direction of traffic – we are looking at an undirected graph.

Conversely, in Figure 8b we can observe that there are *clusters* formed for the other road segments while segment 4 is only slightly correlated with section 241. Nevertheless, there are still weak correlations even at this rel-

atively isolated location. Figure 7 shows that it is not necessary for a road section to have direct adjacent road sections. However, the traffic volume on this road is still influenced by the nearby on and off ramps. Hence, we plot in Figure 7b the closest road sections using the euclidean distance. Recent work optimize only for individual series forecasting and hence does not take proximity data into consideration towards making predictions (Lippi et al., 2013), an observation also made by the authors. It is evident that there are dependencies between sensors in a traffic network, especially at intersections as we show in Figure 8.

Thus far we have observed proximity operators based on direct adjacent connections or euclidean distance. We use the correlation as a metric for selecting road sections as opposed to using the map coordinates for each sensor. In Figure 9 we show that the top most correlated roads for these two sensors are *not* necessarily in the immediate neighbourhood. Therefore, this is not a reliable means of selecting additional relevant data for the predictors, since it is quite unlikely that there are dependencies between road sections that are far apart. Perhaps a better way of selection is to compute the cross-correlation over a fixed window interval, which accounts for shifts in the traffic signal.



(a) Sensor 123 **(b)** Sensor 4

Figure 9: Ranked road sections by correlation. Query is solid black. The highest ranked are shown in dotted black. There is a large distance between the query and the highest correlated over the entire dataset.

The correlations for all sensors in one area are shown in Figure 8. Clearly,

the road section 123 has much more correlated traffic than the off highway ramp (section 4). This is due to the fact that the ramp has no actual connected roads and we selected these roads using the euclidean distance matrix. The implications are evident: there is valuable information carried within the neighbourhood of each query road section where predictions are to be made.

Previous research has shown that separating prediction on working and non-working (weekends) days can improve performance if independent predictors are deployed separately for weekends or each day of the week. Additionally, both a larger temporal context through a larger lag and including proximity data can increase prediction accuracy (Schimbinschi et al., 2015).

4.6. Pair-wise correlations with query station as a function of time

We further investigate whether the interactions depicted in Figure 8 change during the day. In the following figure we can observe that these indeed change. There is an overall pattern, however most importantly there are deviations from the standard pattern such as the sudden spikes at 4 am for the blue sensor station and the spike at 6pm for the green sensor station.

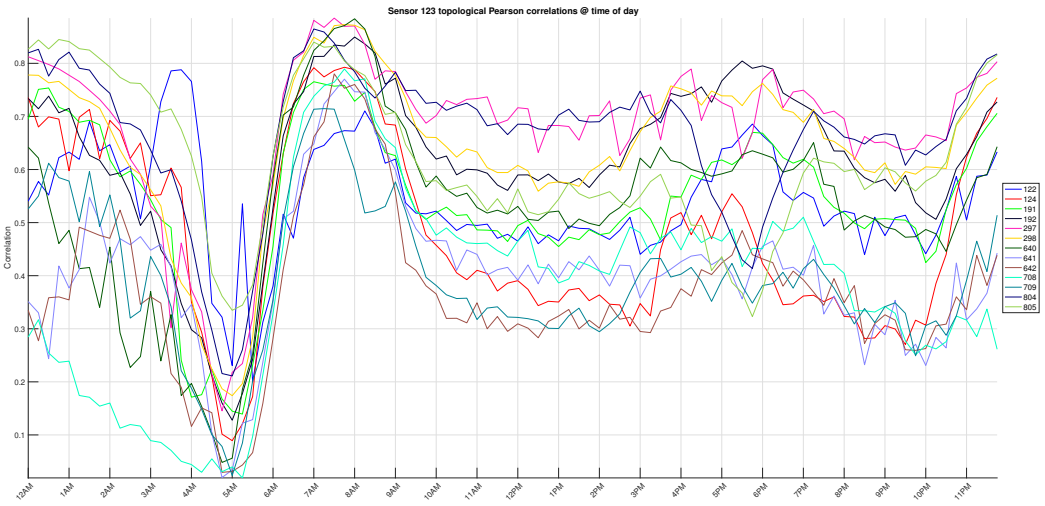


Figure 10: Sensor 123 correlation at time of day with neighboring sensors

5. Results and discussion

In this section we evaluate the network wide prediction performance for the two datasets and compare univariate methods to Topology Regularized

Vector Autoregression (TRU-VAR) generalized to state of the art function approximators. In all experiments we only report the error of ex-ante point forecasts, in other words, no information that would not be available at the time of prediction was used in any of the experiments.

5.1. Experimental setup

For all experiments the data was split sequentially into 70% training 15% validation and 15% testing. Preparation of data is discussed in Section 4. In the case of ARIMA, the models were fit on the training plus validation data. The regularization parameters and the fitting of the ARIMA (using the forecast package in R (Hyndman et al., 2007)) were found independently for each sensor station. The MLPs are trained using the gauss-newton approximation for bayesian L_2 regularized backpropagation (Foresee & Hagan, 1997). In the case of the LLS and SVR we use L-BFGS (Nocedal & Wright, 2006, ch. 9) for ridge and SPARSA (Wright et al., 2009) for LASSO regularization.

As a follow-up on the observations made in Figure 8a and Figure 8b we propose learning Topology Regularized Universal Vector Autoregression by inducing sparsity in the VAR model, in effect including only relevant data from the nearby roads to each autoregressor. However, not all road sections have direct connections. For the VicRoads dataset, we only use data from the directly connected roads, if the road section has direct connections. In Figure 7b the nearest roads for road section 4 are plotted, however there are no directly connected roads, a case where no additional data is added. In this case, the topological adjacency matrix can be refined according to other heuristics for selecting relevant roads, as discussed in Section 3. PeMS is very sparse given the area covered, we take the closest $K = 6$ roads as additional data, computed using the graph adjacency matrix from the map GPS coordinate of each sensor. We empirically selected K based on the out of sample (test set) mean RMSE using univariate OLS fitting.

We set a naive baseline using the CMSD as specified in Section 4, which is the average specific to the time of day, sensor and day of the week. We furthermore compare these methods with ARIMA which is also an intuitive baseline. Note that we do not provide comparisons with VAR since this would be computationally prohibitive, expecially for the VicRoads dataset where the input space would have almost 10k ($\Delta = 10 \times 1084$) dimensions, just for one vector. We do not perform direct comparisons with recent methods (Ko et al., 2016; Salamanis et al., 2016; Wu et al., 2016; Ahn et al., 2015; Fusco et al., 2015; Lv et al., 2015; Xu et al., 2015) since: 1) this would

be computationally prohibitive – in the original articles, most authors do not perform network-wide experiments and we have a dataset with 1000+ sensor stations; 2) these methods do not have the properties described in the introduction (Table 1 on 4); 3) we aim for a direct comparison over the univariate equivalent of the function approximator used within TRU-VAR.

Univariate models and TRU-VAR are compared via the average RMSE and MAPE for both datasets.

5.2. Choosing the lag order

We train a linear TRU-VAR over increasing lag windows $\Delta \in \{1, 2, 3, 5 \ldots, 31\}$ and plot the validation set RMSE and computation time, towards making an empirical choice for the lag value. A larger lag decreases prediction error, while putting a heavier load on processing time. As a tradeoff we set $\Delta = 10$ and use the same lag value for the VicRoads dataset.

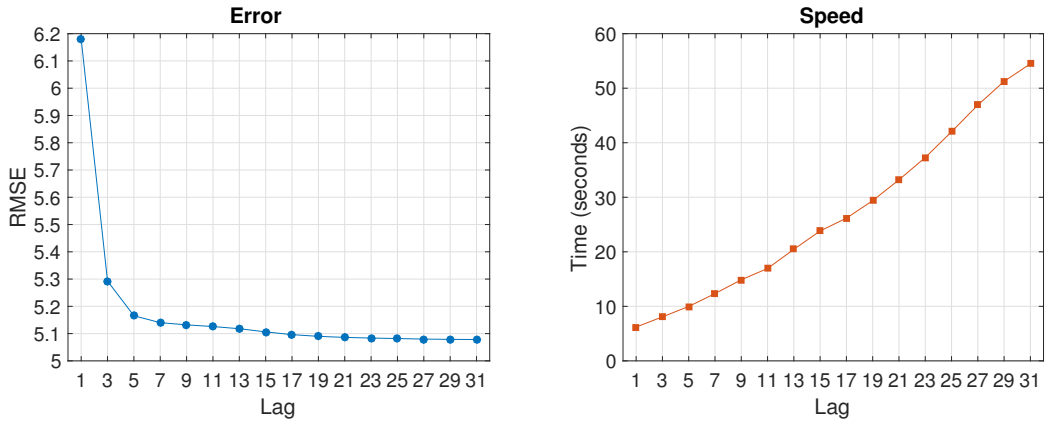


Figure 11: RMSE and prediction time as a function of lag (PeMS).

Using this procedure we aim to get an estimate of the out of sample error. We also mention that using this lag value the error on the test set is lower for models with the same lag value as oppossed to the ARIMA forecasts where the lags were selected automatically for each timeseries based on the AIC. It is however likely that if optimizing the lag values in the same way for TRU-VAR, the error could be further lowered.

5.3. TRU-VAR vs. Univariate

It can be observed from Table 2 that TRU-VAR outperformed the univariate methods in all cases except for SVR-L_1.

Table 2: VicRoads dataset - Average **RMSE**
Topology regularized universal vector autoregression (TRU-VAR)
outperforms univariate models for all f except SVR-L_1.

$f(x)$	OLS	LLS-L_1	LLS-L_2	SVR-L_1	SVR-L_2	MLP-L_2
Univariate	24.11	24.13	24.37	24.94	24.51	**22.09**
	± 15.4	± 15.4	± 15.5	± 15.9	± 15.9	± 15.6
TRU-VAR	**22.64**	**23.14**	**22.93**	26.68	**22.78**	**<u>21.36</u>**
	± 15.6	± 15.7	± 15.4	± 18.5	± 15.1	± 15.3
Baselines	CMSD: 35.29 ± 29.0			ARIMA: 24.32 ± 15.6		

Table 3: VicRoads dataset - Average **MAPE** Topology regularized universal vector autoregression (TRU-VAR)
outperforms univariate models for all f except SVR-L_1.

$f(x)$	OLS	LLS-L_1	LLS-L_2	SVR-L_1	SVR-L_2	MLP-L_2
Univariate	26.94	27.14	28.42	28.22	27.56	**21.27**
	± 12.4	± 12.6	± 14.2	± 14.6	± 14.6	± 8.5
TRU-VAR	**22.96**	**24.53**	**24.16**	31.91	**24.28**	**<u>21.03</u>**
	± 9.9	± 13.8	± 12.5	± 26.3	± 13.7	± 11.7
Baselines	CMSD: 32.86 ± 53.4			ARIMA: 27.91 ± 13.4		

For PeMS there are usually just two adjacent sensors (upstream and downstream) which can contribute to the traffic volume. We were unable to pinpoint the start and end location of the road section covered by the sensor station unlike VicRoads. Since we only had the GPS coordinates of stations themselves, we defined the TDAM based on the euclidean distance to the query sensor station. This usually resulted in includin the sensor stations on the opposite direction of traffic. While this helps in the case of VicRoads, for freeways the traffic is strictly separated between traffic directions, hence it is not relevant and adds unecessary complexity. However, it is interesting that we were able to lower the prediction error by defining the adjacency matrix based on the K roads furtherest apart from the prediction location

for the OLS and MLP. The results are shown in Table 4 in the last row. From the same table it can be seen that for higher temporal resolutions such as in the case of PeMS, the lowest errors are recorded via OLS and MLP while ARIMA, LLS and SVR show higher error.

Table 4: PeMS dataset - Average **RMSE**
Topology regularized universal vector autoregression (TRU-VAR) outperforms univariate models for all f except LLS-L_1.

$f(x)$	OLS	LLS-L_1	LLS-L_2	SVR-L_1	SVR-L_2	MLP-L_2
Univariate	4.61	4.64	4.97	5.20	4.72	**4.55**
	± 2.1	± 2.0	± 2.4	± 2.6	± 2.0	± 2.0
TRU-VAR	**4.53**	4.64	**4.92**	**5.03**	**4.70**	**4.45**
	± 2.0	± 2.1	± 2.5	± 2.7	± 2.0	± 1.9
Baselines	CMSD: 5.16 ± 3.1			ARIMA: 4.67 ± 1.7		

Table 5: PeMS dataset - Average **MAPE**
Topology regularized universal vector autoregression (TRU-VAR) outperforms univariate models for all f except LLS-L_1.

$f(x)$	OLS	LLS-L_1	LLS-L_2	SVR-L_1	SVR-L_2	MLP-L_2
Univariate	37.77	38.37	45.12	42.80	38.53	**37.48**
	± 9.4	± 11.0	± 15.9	± 20.4	± 13.5	± 9.8
TRU-VAR	**36.95**	38.37	**41.59**	**39.74**	**37.69**	**37.42**
	± 10.3	± 12.5	± 14.7	± 14.5	± 12.3	± 12.6
Baselines	CMSD: 41.9 ± 19.2			ARIMA: 38.09 ± 15.76		

5.4. Long-term forecasting: increasing the prediction horizon

In the previous section we shoed that TRU-VAR outperforms univariate methods across different machine learning function approximation models. We now ask if this holds for larger prediction horizons for to up to two hours. Consequently, we compare the prediction performance of TRU-VAR

and univariate models with the two best performing models from the previous section (OLS and MLP). The experiment is identical to the one in the previous section, with the exception that now, instead of predicting at the immediate step in the future (e.g. $h = 1$) we set the target variable to be further in time. In other words, methods are idential data split is identical, input data is the same while the target variable changes.

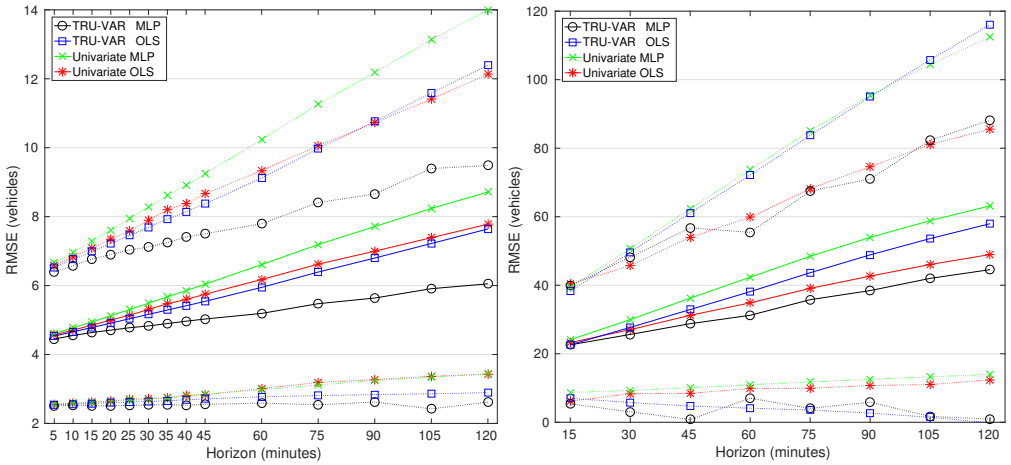
Therefore, in the following section we show how the overall network error rises as the prediction horizon is increased to up to two hours. We now refer the reader to Eq. 1 on page 14 for the definition of the prediction horizon. The horizon temporal resolution for $t+h$ where $h = 1$ was initially 5 minutes for PeMS and 15 minutes for VicRoads which correspond to the temporal resolution of the dataset.

Instead of predicting the volume of traffic at $t+1$ (for all series) we instead train and subsequently evaluate the prediction error at horizons for up to two hours. For PeMS, which has a resolution of 5 minutes per observation we plot the error for eight 5-minute distanced horizons and then add 15-minute horizons up to two hours (we skip a few horizons) – this is visible in Figure 12a. For VicRoads the resolution is 15 minutes thus we only plot 8 forecasts to reach the two hour goal (Figure 12b). Therefore, to evaluate all horizons for up to two hours we require to fit $h \in \{1, 2, \ldots, 8\}$ models for VicRoads and $h \in \{1, 2, \ldots, 24\}$ for PeMS. However, for PeMS we skip a few evaluations and take $h \in \{1, 2, 3, \ldots, 8\} \cup \{9, 12, \ldots, 24\}$.

Results are displayed in Figure 12 where the network-mean RMSE is depicted with solid lines and the standard deviation with dotted lines as an indication of the network-spread of the error. It is evident that it is more challenging to make predictions further in time, and it can be seen that the error increases linearly as the prediction horizon is increased.

From both figures we can observe that TRU-VAR MLP predicts with the lowest error even as the forecasting horizon is increased, while the Univariate MLP (one simple neural network per timeseries) is the worst for both datasets. As the prediction horizon is moved further ahead in time not only the mean RMSE increases, but the spread of the error over the network also increases. For PeMS the error increases by approximately 36% when the horizon is moved from five minutes to two hours (for TRU-VAR MLP) while for VicRoads, the error increases by 97%. This is to be expected for a much larger network, with a more complex topology, a greater variety of road types and a lower temporal resolution.

Overall, the error spread appears to be much more stable for PeMS and



(a) PeMS **(b)** VicRoads

Figure 12: Behavior of error when prediction horizon is increased. μ RMSE with solid lines and spread of $\mu \pm \sigma$ RMSE with dotted lines. Lower is better, error increases linearly with the prediction horizon.

the OLS approximators. This is for two reasons: 1) VicRoads has 10 times more sensor stations, which can cause a much larger error spread; 2) for the MLP experiments we trained the neural networks using first order gradient methods which was faster, however less stable than when using bayesian regularization as in the previous experiments in Table 2.

6. Conclusions and future work

In the current article we defined necessary properties for large scale network wide traffic forecasting in the context of growing urban road networks and (semi)autonomous vehicles. We performed a broad review of the literature and after drawing conclusions, we discussed data quality, preprocessing and provided suggestions for defining a topology-designed adjacency matrix (TDAM).

We consequently proposed topology-regularized universal vector autoregression (TRU-VAR). We compared the network-wide prediction error of the univariate and TRU-VAR prediction models over two quantitatively and qualitatively different datasets. TRU-VAR outperforms the CMSD baseline and ARIMA in all cases and the regularized univariate models in almost all cases. For VicRoads, which has high spatial but relatively lower temporal

resolution, TRU-VAR outperformed the univariate method in all cases except for SVR-L_1. For the PeMS dataset, TRU-VAR showed lower error in all cases except for LLS-L_1. From the prediction horizon experiments (Figure 12) we concluded that the TRU-VAR MLP approximator has the lowest error even when the forecast horizon is increased up to two hours, for both datasets.

PeMS was easier to predict and the RMSE is lower than for VicRoads. We would like to remind the reader that approximately half of the sensor stations in the VicRoads dataset can have more than 50% data missing. We did not remove these from the model. This evidently increases the overall error. At the same time, the PeMS dataset has higher temporal resolution (5 vs 15 minutes) however it is much smaller and has 10 times less sensor stations. With continuous state-space models and accounting for the spatial sparsity (large distance between sensor stations on highways causes shifts in the signals) the error could be further decreased.

We conclude that the TDAM is a key component to our proposed traffic forecasting expert system since it has a great impact on prediction accuracy and should be tailored to the type of road network using expert domain knowledge heuristics. Therefore, the weak points of our method also reside in its strengths, namely: the design of the topological adjacency matrix is very important and is subject to domain knowledge; the model customization flexibility allows for a large search space of possibilities which can be overwhelming to fine-tune, if required. When designing the adjacency matrix for the VicRoads dataset, we could have opted to rank the most correlated connected ones and select the top K ranked ones, in this way having an equal number of additional streams for each prediction point. For the roads without direct connections, this would require computing the distance to the closest roads based on the euclidean distance and performing a correlation ranking. This could further increase accuracy since for some sensors (for example highway off ramps) there are no directly connected roads, hence simple regression is performed. An alternative would have been to select second order connections for the stations with a low graph node degree, such that of highway off ramps. We leave this for future work, also aiming to investigate continuous state space models, multi-metric data and other multi-task learning architectures. Moreover, we also aim to show that including temporal features and adding multi-metric data (such as vehicle-crowdsourced data) can further increase prediction performance.

Finally, we would like to point out that our method can be trained online,

can have a nonlinear reporesentation, has a low complexity due to the topological constraints, is non-static and robust towards changes (data sources or structure) thus scales well, is efficient and easy to redeploy. Given that the error increases linearly within reasonable bounds (Fig. 12) for up to two hours, our method may be useful in other contexts and applications other than road traffic, such as natural disaster prevention (e.g. river flood forecasting), telecommunications (e.g. antenna load forecasts), networking (e.g. forecasts for routing), finance to name a few. In general, the method can be applied to any type of dataset consisting of multivariate timeseries, where the constraints are known a priori or can be inferred from the data to construct the topology matrix.

References

Abdulhai, B., Porwal, H., & Recker, W. (2002). Short-term traffic flow prediction using neuro-genetic algorithms. *ITS Journal-Intelligent Transportation Systems Journal*, *7*, 3–41.

Ahn, J. Y., Ko, E., & Kim, E. (2015). Predicting spatiotemporal traffic flow based on support vector regression and bayesian classifier. In *Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on* (pp. 125–130). IEEE.

Asif, M. T., Dauwels, J., Goh, C. Y., Oran, A., Fathi, E., Xu, M., Dhanya, M. M., Mitrovic, N., & Jaillet, P. (2014). Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, *15*, 794–804.

Athanasopoulos, G., Poskitt, D., & Vahid, F. (2012). Two canonical varma forms: Scalar component models vis-à-vis the echelon form. *Econometric Reviews*, *31*, 60–83.

Au, T.-C., Zhang, S., & Stone, P. (2015). Autonomous intersection management for semi-autonomous vehicles. *Handbook of Transportation. Routledge, Taylor & Francis Group*, .

Blue, V., List, G. F., & Embrechts, M. J. (1994). Neural net freeway travel time estimation. In *Intelligent Engineering Systems through Artificial Neural Networks (Conference: 1994: Saint Louis, Mo.). Proceedings. Vol. 4*.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Castillo, E., Menéndez, J. M., & Sánchez-Cambronero, S. (2008). Predicting traffic flow using bayesian networks. *Transportation Research Part B: Methodological*, *42*, 482–509.

Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., & Han, L. D. (2009). Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert systems with applications*, *36*, 6164–6173.

Çetiner, B. G., Sari, M., & Borat, O. (2010). A neural network based traffic-flow prediction model. *Mathematical and Computational Applications*, *15*, 269–278.

Chai, T., & Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific Model Development*, *7*, 1247–1250.

Chen, C., Hu, J., Meng, Q., & Zhang, Y. (2011). Short-time traffic flow prediction with arima-garch model. In *Intelligent Vehicles Symposium (IV), 2011 IEEE* (pp. 607–612). IEEE.

Cheng, T., Haworth, J., & Wang, J. (2012). Spatio-temporal autocorrelation of road network data. *Journal of Geographical Systems*, *14*, 389–413.

Clark, S. (2003). Traffic prediction using multivariate nonparametric regression. *Journal of transportation engineering*, *129*, 161–168.

Clark, S. D., Dougherty, M. S., & Kirby, H. R. (1993). The use of neural networks and time series models for short term traffic forecasting: a comparative study. In *Transportation Planning Methods. Proceedings Of Seminar D Held At The Ptrc European Transport, Highways And Planning 21st Summer Annual Meeting (September 13-17, 1993), Umist. Volume P363*.

Çolak, S., Lima, A., & González, M. C. (2016). Understanding congested travel in urban areas. *Nature communications*, *7*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*, 303–314.

Dia, H. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, *131*, 253–261.

Diebold, F. X., & Mariano, R. S. (2012). Comparing predictive accuracy. *Journal of Business & economic statistics*, .

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*, 78–87.

Dougherty, M. S., & Cobbett, M. R. (1997). Short-term inter-urban traffic forecasts using neural networks. *International journal of forecasting*, *13*, 21–31.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. In *Neural Networks, 1997., International Conference on* (pp. 1930–1935). IEEE volume 3.

Fusco, G., Colombaroni, C., Comelli, L., & Isaenko, N. (2015). Short-term traffic predictions on large urban traffic networks: applications of network-based machine learning models and dynamic traffic assignment models. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on* (pp. 93–101). IEEE.

Guidotti, R., Nanni, M., Rinzivillo, S., Pedreschi, D., & Giannotti, F. (2016). Never drive alone: Boosting carpooling with network analysis. *Information Systems*, .

Hamilton, J. D. (1994). *Time series analysis* volume 2. Princeton university press Princeton.

Högberg, P. (1976). Estimation of parameters in models for traffic prediction: a non-linear regression approach. *Transportation Research*, *10*, 263–265.

Hong, H., Huang, W., Xing, X., Zhou, X., Lu, H., Bian, K., & Xie, K. (2015). Hybrid multi-metric k-nearest neighbor regression for traffic flow prediction. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems* (pp. 2262–2267). IEEE.

Hyndman, R. J., Khandakar, Y. et al. (2007). *Automatic time series for forecasting: the forecast package for R*. Technical Report Monash University, Department of Econometrics and Business Statistics.

Jensen, T., & Nielsen, S. (1973). Calibrating a gravity model and estimating its parameters using traffic volume counts. In *5th Conference of Universities' Transport Study Groups, University College, London*.

Kachroo, P., & Sastry, S. (2016). Traffic assignment using a density-based travel-time function for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, *17*, 1438–1447.

Kamarianakis, Y., & Prastacos, P. (2005). Space–time modeling of traffic flow. *Computers & Geosciences*, *31*, 119–133.

Kamarianakis, Y., Shen, W., & Wynter, L. (2012). Real-time road traffic forecasting using regime-switching space-time models and adaptive lasso. *Applied Stochastic Models in Business and Industry*, *28*, 297–315.

Karlaftis, M. G., & Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, *19*, 387–399.

Ko, E., Ahn, J., & Kim, E. Y. (2016). 3d markov process for traffic flow prediction in real-time. *Sensors*, *16*, 147.

Kumar, S. V., & Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, *7*, 1–9.

Lee, E.-M., Kim, J.-H., & Yoon, W.-S. (2007). Traffic speed prediction under weekday, time, and neighboring links' speed: back propagation neural network approach. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues* (pp. 626–635). Springer.

Levin, M., & Tsao, Y.-D. (1980). On forecasting freeway occupancies and volumes (abridgment). *Transportation Research Record*, .

Liebig, T., Piatkowski, N., Bockermann, C., & Morik, K. (2016). Dynamic route planning with real-time traffic predictions. *Information Systems*, .

Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised

learning. *Intelligent Transportation Systems, IEEE Transactions on*, *14*, 871–882.

Low, D. E. (1972). A new approach to transportation systems modeling. *Traffic quarterly*, *26*.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: A deep learning approach. *Intelligent Transportation Systems, IEEE Transactions on*, *16*, 865–873.

Lv, Y., Tang, S., & Zhao, H. (2009). Real-time highway traffic accident prediction based on the k-nearest neighbor method. In *2009 International Conference on Measuring Technology and Mechatronics Automation* (pp. 547–550). IEEE volume 3.

Min, W., & Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, *19*, 606–616.

Min, X., Hu, J., Chen, Q., Zhang, T., & Zhang, Y. (2009). Short-term traffic flow forecasting of urban network based on dynamic starima model. In *2009 12th International IEEE Conference on Intelligent Transportation Systems* (pp. 1–6). IEEE.

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

Moreira-Matias, L., & Alesiani, F. (2015). Drift3flow: Freeway-incident prediction using real-time learning. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems* (pp. 566–571). IEEE.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2016). Time-evolving od matrix estimation using high-speed gps data streams. *Expert Systems With Applications*, *44*, 275–288.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*.

Nocedal, J., & Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Oh, S., Byon, Y.-J., Jang, K., & Yeo, H. (2015). Short-term travel-time prediction on highway: a review of the data-driven approach. *Transport Reviews*, *35*, 4–32.

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

Rice, J., & Van Zwet, E. (2004). A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, *5*, 200–207.

Salamanis, A., Kehagias, D. D., Filelis-Papadopoulos, C. K., Tzovaras, D., & Gravvanis, G. A. (2016). Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction. *IEEE Transactions on Intelligent Transportation Systems*, *17*, 1678–1687.

Schimbinschi, F., Nguyen, X. V., Bailey, J., Leckie, C., Vu, H., & Kotagiri, R. (2015). Traffic forecasting in complex urban networks: Leveraging big data and machine learning. In *Big Data (Big Data), 2015 IEEE International Conference on* (pp. 1019–1024). IEEE.

Smith, B. L., & Demetsky, M. J. (1997). Traffic flow forecasting: comparison of modeling approaches. *Journal of transportation engineering*, *123*, 261–266.

Smith, B. L., Williams, B. M., & Oswald, R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, *10*, 303–321.

Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in neural information processing systems*, *9*, 155–161.

Stathopoulos, A., Dimitriou, L., & Tsekeris, T. (2008). Fuzzy modeling approach for combined forecasting of urban traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, *23*, 521–535.

Stathopoulos, A., & Karlaftis, M. G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C: Emerging Technologies*, *11*, 121–135.

Su, F., Dong, H., Jia, L., Qin, Y., & Tian, Z. (2016). Long-term forecasting oriented to urban expressway traffic situation. *Advances in Mechanical Engineering*, *8*, 1687814016628397.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, *9*, 293–300.

Szeto, W., Ghosh, B., Basu, B., & O'Mahony, M. (2009). Multivariate traffic forecasting technique using cell transmission model and sarima model. *Journal of Transportation Engineering*, *135*, 658–667.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, (pp. 267–288).

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

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*, 988–999.

Varaiya, P. P. (2001). *Freeway performance measurement system: Final report*. Citeseer.

Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. *Transport reviews*, *24*, 533–557.

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transportation Research Part C: Emerging Technologies*, *13*, 211–234.

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

Wang, Y., Papageorgiou, M., & Messmer, A. (2008). Real-time freeway traffic state estimation based on extended kalman filter: Adaptive capabilities and real data testing. *Transportation Research Part A: Policy and Practice*, *42*, 1340–1358.

Williams, B. (2001). Multivariate vehicular traffic flow prediction: evaluation of arimax modeling. *Transportation Research Record: Journal of the Transportation Research Board*, (pp. 194–200).

Wright, S. J., Nowak, R. D., & Figueiredo, M. A. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, *57*, 2479–2493.

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

Wu, Y.-J., Chen, F., Lu, C.-T., & Yang, S. (2016). Urban traffic flow prediction using a spatio-temporal random effects model. *Journal of Intelligent Transportation Systems*, *20*, 282–293.

Xie, Y., Zhang, Y., & Ye, Z. (2007). Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *22*, 326–334.

Xu, Y., Chen, H., Kong, Q.-J., Zhai, X., & Liu, Y. (2015). Urban traffic flow prediction: a spatio-temporal variable selection-based approach. *Journal of Advanced Transportation*, .

Zeng, D., Xu, J., Gu, J., Liu, L., & Xu, G. (2008). Short term traffic flow prediction based on online learning svr. In *Power Electronics and Intelligent Transportation System, 2008. PEITS'08. Workshop on* (pp. 616–620). IEEE.

Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, *50*, 159–175.

Zheng, W., Lee, D.-H., & Shi, Q. (2006). Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering*, *132*, 114–121.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*, 301–320.