

Evaluation of Smart-phone Performance for Real-time Traffic Prediction

Rauf Ansar¹, Paron Sarampakhul¹, Soham Ghosh¹, Nikola Mitrovic¹,
Muhammad Tayyab Asif¹, Justin Dauwels¹ and Patrick Jaillet²

Abstract—Smartphones efficiently collect traffic information, guide the drivers and inform the end-user about the current and future traffic conditions. Due to enhanced sensor technology, visualization capabilities, navigational performance and network connectivity, smartphones play an important role in advanced travel information systems (ATIS). Although they offer increasing computation power nowadays, this potential smartphone resource has not been explicitly evaluated for neither ATIS nor Intelligent Transport System (ITS) applications. In this study, we actively involve the smartphone into real-time compressed prediction of large traffic networks. More precisely, we run prediction algorithms on the central server to obtain future state for the subset of the links in the network that we refer to as compressed network state. Then, we send the predicted values for compressed network state to smartphones where network extrapolation is performed. Network extrapolation involves vector-matrix multiplication where row vector represents the compressed network state while the matrix is stored on the mobile phone and contains the relationships function between the compressed state and entire network. Such decentralized infrastructure can significantly reduce the overhead of the communication network and enhance the development of cooperative, peer to peer networks for the NextGen Intelligent Transportation Systems applications.

I. INTRODUCTION

With the recent development in sensing and communication technology, advanced travel information systems (ATIS) become the key component towards the efficient utilization of existing traffic infrastructure. ATIS contain technological framework that: (1) collects traffic variables in the large traffic networks and with the high temporal frequency [1]–[3]; (2) processes the collected data and reveal the meaningful information from it [4]–[7]; (3) disseminates the instructive real-time information to the end users (travelers) in the form of either colored

network maps or suggested trip characteristics such as departure time, mode of transportation and assigned path [3], [8]. The provided information is frequently delivered via smartphones, embedded GPS devices or desktop applications [8], [9]. Travelers utilize provided information to chose less congested paths and consequently decrease their travel time. In that way ATISs reduce the total delay, improve the user's comfort and satisfaction, decrease the pollution and noise at congestion sites and enhance the overall productivity within a city [10].

Smartphones are frequently seen as the most convenient way to deliver real-time information to the travelers. With easy-to-use characteristics, high penetration rate and powerful performance, smartphones play important role in both sensing and delivering traffic information. In the former, numerous smart sensors such as accelerometers, GPS, manometers, microphones and even cameras have found many application domains in intelligent transportation systems (ITS) [2], [11]–[15]. In the later, high visualization and navigational performance of the smart devices inform the end-users about the network condition and direct travelers to their destinations [8], [9], [16]–[18]. Although smart devices possess the significant and fast-growing computational power, this potential is not explicitly evaluated for ITS applications.

In this paper, we evaluate the capabilities of smartphones for real-time multi-horizon traffic prediction by assigning a certain computations to user's smartphone device. Real time traffic stream is provided by California Department of Transportation (Caltrans). Caltrans deliver us traffic variable (speed) for the freeway network across all major metropolitan areas of the State of California. This data is coming to our server every 5 minutes. To perform traffic prediction we use compressed method, developed in our previous study [19]. Compressed method explicitly predicts the traffic conditions for the subset of road segments (or loop stations in this study) using standard regression methods. For the rest of the network traffic conditions are obtained through straightforward vector-matrix multiplication where: row vector contains traffic conditions for the subset of road segments (e.g. 10%); the relationship matrix consists the functions between the subset of road segments and whole network. We propose to perform network extrapolation on user' device. Such a decentralized system can significantly reduce the overhead of the communication network regarding that traffic condition for few network places need to be monitored, processed and sent to end users. Such an approach also opens up the development of cooperative, peer to

¹Rauf Ansar, Paron Sarampakhul, Soham Ghosh, Nikola Mitrovic, Muhammad Tayyab Asif and Justin Dauwels are with the School of Electrical and Electronic Engineering, College of Engineering, Nanyang Technological University, Singapore, 639798. ansar002@e.ntu.edu.sg, muhammad89@e.ntu.edu.sg, jdauwels@ntu.edu.sg

²Patrick Jaillet is with the Department of Electrical Engineering and Computer Science, School of Engineering, and also with the Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is also with the Center for Future Urban Mobility, Singapore-MIT Alliance for Research and Technology, Singapore. jaillet@mit.edu

This work was supported in part by the Singapore National Research Foundation through the SingaporeMassachusetts Institute of Technology Alliance for Research and Technology (SMART) Center for Future Mobility (FM). We would also like to acknowledge California Department of Transportation (Caltrans) for providing us the real-time traffic data feed and the GIS resources of their road networks.

peer networks for the NextGen Intelligent Transportation Systems. As a comparison, we explore: (1) traditional (uncompressed) prediction method where traffic condition for each road segment is explicitly predicted and sent to user's smartphone; (2) explicit prediction for subset of the links, followed by network extrapolations that is performed on central server computer. Like in previous case, traffic information for the entire network is sent to the user's phone. We evaluate the accuracy and computational time of the tested methods using Caltrans real-time data stream. To achieve this we have developed mobile traffic application that performs vector-matrix multiplication and contains the matrix of relationship functions. Finally mobile application overlays the background mobile map and inform the drivers about the predicted traffic conditions.

The rest of the paper is structured as follows. In section II, we describe the system infrastructure that we use for real-time traffic prediction. We also briefly explain the compressed prediction method that has been developed in our previous study. In Section III, we explain the traffic data set under consideration and experimental setup. In Section IV, we provide the accuracy and computational complexity of the tested approaches. We also briefly discuss the existing capabilities of smartphones for traffic data analytics. In Section V, we summarize our contributions and suggest topics for future work.

II. ARCHITECTURAL OVERVIEW

This section presents the general architectural design of the system for real-time traffic prediction on smartphones that we use for evaluation [16]. The prototype system consists of three components as illustrated in Fig. 1. In the following subsections we briefly explain each component.

A. Third Party Dynamic Traffic Information Provider

California Department of Transportation (Caltrans) provided us real-time stream and historical database of traffic data through their Performance Measurement System (PeMS) (see Fig.1). Real-time data is available on Caltrans FTP server and is refreshed every 5 minutes. Historical data is available for offline download on Caltrans internet website [20]. Currently, we are using only the speed data, generated from loop detectors that are installed in California highway network [20].

B. Server Side Implementation

The server is the backbone of the implemented infrastructure. Once the new Caltrans data becomes available, the server fetches the real-time traffic information. Traditional and compressed prediction approaches are deployed on the server. In the former, server takes the real-time and corresponding historical information for each road segment (loop station) to perform prediction using a standard regression model such as Support Vector Regression (SVR). Once the prediction is done, server sends the information for the entire network to user's phone. In the later, predefined subset of the links is used for explicit traffic

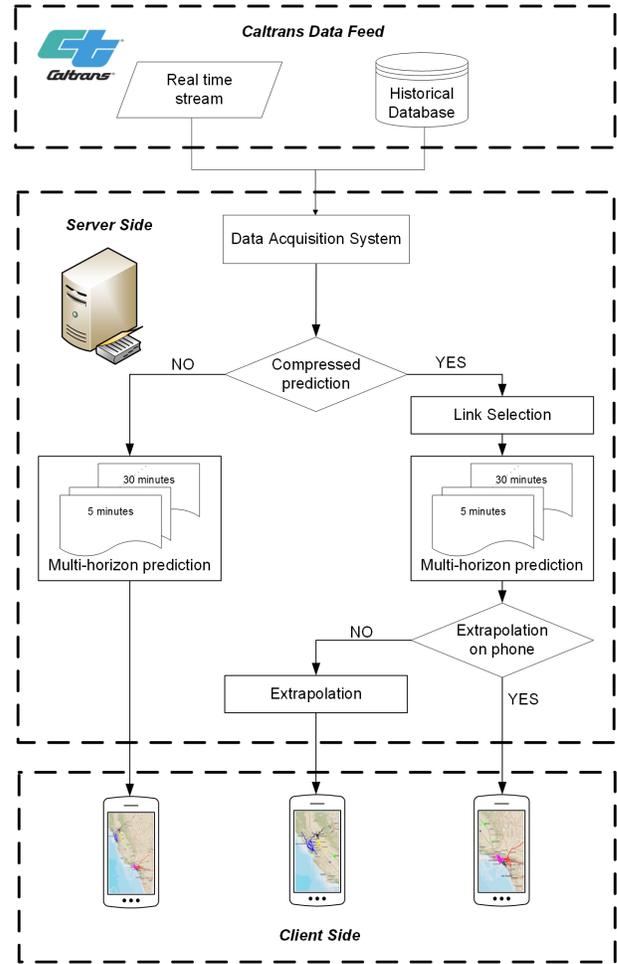


Fig. 1: Overall system architecture

prediction. In the following we briefly review the compressed prediction that is deployed on the server.

Compressed traffic prediction is fast and scalable method for traffic prediction. It explicitly predicts the traffic conditions for the representative subset of the links, which are obtained by analyzing the historical data [19]. We refer to this subset as compressed traffic state. We chose the subset of the links by: (i) assigning the importance to each road segment in the network and (ii) performing the random sampling using the assigned importance (probability) [19], [21]. Future state of the compressed network is obtained using support vector regression [22]. The prediction results for the whole network are provided through straightforward vector-matrix multiplication where row vector comprises the traffic variable for compressed network state while the matrix contains relationship functions between the compressed state and whole network [19]. These relationship functions are learned using the historical dataset.

In our analysis we consider two sub-cases of compressed prediction: (i) vector-matrix multiplication is performed on the server; (ii) vector-matrix multiplication is performed on user's phone and makes the overall system decentralized (see Fig. 2). In the former, traffic condition for each road segment in the network is sent to user's phone. Smartphone only visualizes the obtained data. In the later, traffic conditions for

the “representative” subset of the links is sent to user’s phone. In this case smartphone performs the network extrapolation, followed by data visualisation.

Server sends the traffic information to mobile phone through Microsoft Internet Information service (IIS) webserver. The predicted traffic information is currently in the form of a comma separated files (CSV) and divided into regions, thus making it platform independent with regard to the client i.e. mobile phone.

C. Mobile Application Implementation

Mobile phones receive the traffic information from the server and deliver the predicted network states to the users through developed smart-phone’s application. Smart-phone application aims to overlay traffic information on the base map once the new data comes from the server. In the case that network extrapolation is performed on user’ phones, mobile application package has to contain matrix of relationship functions.

The project is implemented for Android mobile operating system. The open source nature of Android application development is the main reason in selection of the platform. Although project is implemented in Android, the scope is not just limited to Android and using the similar approach it can be implemented for other mobile operating systems too.

Traffic information coming from server is presented to user in a geographic context. ESRI’s ArcGIS runtime SDK for Android is used to develop application and implement geographic information system (GIS) features like map layers, geocoding, location services etc [23]. The SDK enables to display and edit feature geometry and attributes which are required for traffic visualization and route display. ArcGIS World Street Map is used as a base map for overlaying traffic information [24]. GIS metadata which includes geographic locations of loop detectors for each district is provided by Caltrans and is stored on mobile phone as part of the application package.

III. EXPERIMENTAL SETUP

In this section we describe the traffic data considered in the study. We also explain how we run the experiments. At last, we introduce various measures to assess the tested methods.

We deploy the tested methods in large California freeway network (see Fig. 2 left). The California department of Transportation (Caltrans) provided us historical and real-time speed data with sampling interval of 5 minutes. The reported speed represents the average speed of all vehicles which traverse across the loop station point during the given sampling interval.

Due to sensor malfunction and Caltrans replacement operations, some stations do not report the traffic information all the time. In our experiments we only use those stations that are present in the historical databases for last several months. We end up with a traffic network comprising of 10355 links (road segments) divided into nine different regions (D3-D12 without D9) where the lowest and the highest number of the links in one region are 171 and 2467,

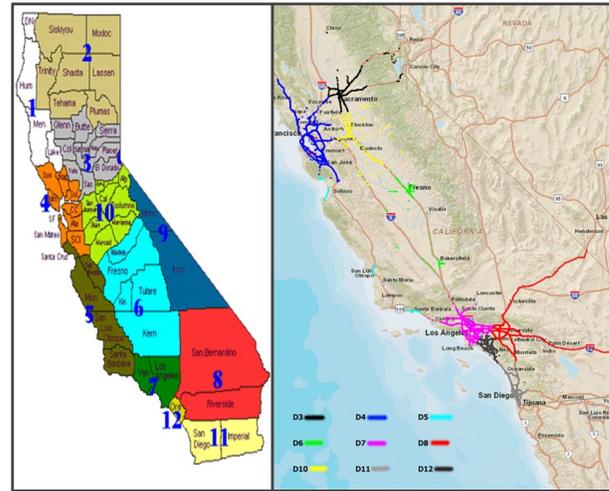


Fig. 2: California freeway network; divided into districts respectively (see Fig. 2 right). The resulting set has negligible percentage (0.1%) of missing data that is not taken in consideration.

In this study, we evaluate the prediction accuracy and computational times of three approaches. In the first approach we use server to explicitly predict traffic speed for each link in the network and utilize mobile app only for visualization purposes. In the second approach we use server to perform compressed prediction, extrapolation and send the information for the entire network to mobile phone. Similar to the first approach, mobile application only overlays the background map with the most relevant data. In the third approach we utilize server to explicitly predict the speed for subset of the links. Then, we use smartphone to perform network extrapolation followed by visualization of predicted traffic conditions.

We use three months of data (Jan-March 2014) to assess the prediction performance. We split this dataset into training and testing subsets. Training set contains data for two months, while the testing set contains remaining data. We use training data to: (i) train the predictors; (ii) learn relationships functions between the subnetwork and entire network. We use testing set to evaluate the performance of traditional and compressed prediction. To run prediction we deploy SVR algorithm that is often applied for traffic data [22], [25]. For SVR implementation we utilize matlab package LIBSVM [26]. To predict speed for k -th prediction horizon ahead we use the following feature vectors: (i) current speed and k past speed values for particular road segment; (ii) day in the week and time during the day. In our analysis, we deploy ν -SVR for $\nu=1$ and default values for other parameters [26], [27]. To evaluate performance of the tested methods, we consider percent root mean distortion (PRD) error [28].

We use real-time stream and proposed system infrastructure (see Section II) to assess the total computational time of three different approaches. Total computational time defines the budget of time from the moment that data appears on the server until the predicted data is displayed on user’s smartphone. Total computational

time is sum of: (i) time that server takes to process latest real-time data, and (ii) time that smartphone requires to process the information from the server and display it on the screen. In our analysis we use 2.76 GHz MacPro server with 32GB of random-access memory (RAM) and smartphone Nexus 4 with single processor of 1.5 GHz, and 2GB of RAM. In order to avoid randomness associated with network load and other factors we run experiments 10 times and report the average values.

IV. RESULTS

First we investigate the prediction accuracy of the traditional and compressed method for each region in the network. In traditional method, we use support vector regression (SVR) algorithm to explicitly predict the traffic speed for each road segment in the network. In compressed prediction, we use the future state of small subset of roads (obtained by SVR algorithm) in a particular region to predict future traffic conditions for the entire region. Table I and Table II show the prediction performance of traditional and compressed methods for 5 and 30 min prediction horizons, respectively. Table I shows that SVR algorithm can achieve high prediction accuracy for highway network and short prediction horizon. As expected, the compressed method has slightly larger error than the traditional approach. For each region, error of compressed method decreases if: (i) the portion of the network increases where the traffic speed is explicitly predicted; (ii) the prediction horizon increases (see Table II). Table II shows that predictions of future network states for larger horizons is difficult even with traditional approach. The magnitude of additional errors, for both 5 and 30 minutes prediction horizons are in agreement with the results of other relevant studies [19].

Let us now present the server' and smartphone' computational times. Table III and IV show the computational times that the server takes to perform traffic prediction for 5 and 30 minute prediction horizons, respectively. Each Table shows the computational times for traditional and compressed approaches. In the case of compressed approach, Tables III and IV provide the computational times for explicit prediction (for subset of the links) and for network extrapolation. Table III and IV show the advantage of using compressed prediction over traditional approach in terms of time savings. This saving is proportional to the compression ratio and it is obtained at the expense of minor reduction in prediction accuracy (see Table I and II). Table III and IV further show that network extrapolation time, if performed on server, is negligible. However, this is not the case when extrapolation is performed on tested smartphone. These times are given in Table V. Table V shows the computation times of mobile application for refreshing the user interface and (if necessary) network extrapolation. Regarding that prediction horizon does not effect the smartphone' computation times, Table V is valid for any prediction horizon. The most computationally expensive task for mobile application is the network extrapolation i.e. multiplication of row vector that contains the compressed predictions received from the

server with relationship matrix stored on mobile. This time is proportional to square of the number of links where the speed is explicitly predicted. The network extrapolation times are within the five minute interval, even for the largest region and lowest compression ratio (see Table V).

Table VI and VII show the total computational times for tested approaches and different compression ratios. Obviously, smart phones have fewer computational resources than desktop computers. However, we can expect mobile devices to have more computational power in the future. So far smartphone can easily store the relationship matrices from smallest one of 18kb (for D5 and CR10) up to largest one of 12MB (for D7 and CR2). The experiments also resulted in low battery drainage and manageable power consumption, which were comparable to the native applications on the smartphone. These parameters will be extensively explored in our next study.

V. CONCLUSIONS AND FUTURE WORK

In this paper we evaluated the performance of smartphones in performing real-time traffic predictions. More precisely, we performed prediction for subset of the links on the server and sent the predicted values to smartphone in real-time. The predicted values are sent in the form of the row vector, that we refer to as compressed network state. We utilize smartphone performance to extrapolate results for the entire network by multiplying the row vector of compressed network state with the matrix of relationship functions that is stored on the smartphone, as part of mobile map. In such a way we do not need to collect, process and send the traffic information for the entire region to user' mobile. Instead we focus on subset of "representative" road segments in the network. Such approach provides competitive accuracy with the traditional approach which explicitly predicts traffic variable for each road in the network. Furthermore, such decentralized approach opens up development towards NextGen Intelligent Transportation Systems applications.

For future work we propose to explore different methods that can help us to optimize computation cost of smartphone and reduce the size of computation matrices. Finally, for future work we propose to explore the idea of the dynamic routing that is based on multi-horizon compressed prediction.

REFERENCES

- [1] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 141–154.
- [2] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: A case study in rome," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 141–151, 2011.
- [3] C. L. Schweiger, *Use and Deployment of Mobile Device Technology for Real-time Transit Information*. Transportation Research Board, 2011, vol. 91.
- [4] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1679–1693, 2012.

Region	D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station	799	2194	147	171	2647	1434	558	803	1464
Traditional SVR	2.62	2.55	2.18	2.18	3.339	2.22	4.14	2.39	3.13
CR=2	3.36	4.34	3.18	2.52	5.75	2.94	4.64	3.34	4.28
CR=4	4.18	5.70	4.51	3.22	7.64	3.86	5.09	4.24	5.44
CR=6	4.69	6.30	5.25	3.62	8.52	4.48	5.34	4.82	6.04
CR=8	5.08	6.68	6.09	4.23	9.03	4.85	5.50	5.16	6.46
CR=10	5.42	7.00	6.12	4.47	9.31	5.07	5.69	5.49	6.89

TABLE I: PRD error of the traditional and compressed prediction methods for different regions and compression ratios (CR). Prediction performance refers to 5 min prediction horizon.

Region	D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station	799	2194	147	171	2647	1434	558	803	1464
Traditional SVR	5.17	5.75	5.08	3.77	8.07	4.82	5.20	5.84	6.67
CR=2	5.42	6.53	5.38	3.90	8.90	5.11	5.49	6.02	7.06
CR=4	5.71	7.17	5.79	4.14	9.70	5.49	5.74	6.22	7.45
CR=6	5.89	7.48	6.10	4.27	10.02	5.76	5.88	6.69	7.64
CR=8	6.08	7.66	6.51	4.57	10.27	5.90	6.00	6.51	7.83
CR=10	6.22	7.83	6.52	4.63	10.41	5.99	6.11	6.60	8.02

TABLE II: PRD error of the traditional and compressed prediction methods for different regions and compression ratios (CR). Prediction performance refers to 30 min prediction horizon.

Region		D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station		799	2194	147	171	2647	1434	558	803	1464
Traditional SVR		2.78	6.82	0.55	0.66	8.45	5.3	2.07	3.06	5.82
CR=2	SVR	1.51	3.53	0.31	0.35	4.66	2.89	1.08	1.6	2.96
	Network extrapolation	1.79E-04	1.00E-03	2.93E-05	3.45E-05	1.60E-03	4.66E-04	1.11E-04	1.81E-04	8.66E-04
CR=4	SVR	0.77	1.76	0.17	0.19	2.26	1.41	0.53	0.8	1.46
	Network extrapolation	1.16E-04	5.87E-04	2.43E-05	2.81E-05	7.48E-04	2.70E-04	7.81E-05	3.89E-04	2.71E-04
CR=6	SVR	0.51	1.18	0.11	0.13	1.49	0.92	0.36	0.54	0.97
	Network extrapolation	9.07E-05	4.01E-04	2.07E-05	2.27E-05	4.96E-04	1.93E-04	2.64E-04	8.86E-05	2.00E-04
CR=8	SVR	0.39	0.88	0.09	0.1	1.12	0.7	0.27	0.41	0.73
	Network extrapolation	7.73E-05	3.05E-04	1.83E-05	2.14E-05	3.87E-04	1.63E-04	6.19E-05	7.77E-05	1.66E-04
CR=10	SVR	0.32	0.71	0.08	0.08	0.9	0.56	0.22	0.33	0.59
	Network extrapolation	7.22E-05	2.56E-04	1.80E-05	1.96E-05	3.19E-04	1.20E-04	5.93E-05	1.33E-04	1.31E-04

TABLE III: Time (in seconds) taken by server to perform predictions (5 minute prediction horizon).

Region		D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station		799	2194	147	171	2647	1434	558	803	1464
Traditional SVR		4.85	13.75	0.93	1.11	16.69	8.98	3.48	5.09	9.33
CR=2	SVR	2.48	6.9	0.49	0.57	8.56	4.59	1.76	2.56	4.56
	Network extrapolation	1.70E-04	9.83E-04	2.88E-05	3.31E-05	0.002	4.29E-04	1.05E-04	1.68E-04	4.62E-04
CR=4	SVR	1.27	3.43	0.26	0.31	4.26	2.27	0.89	1.28	2.33
	Network extrapolation	1.07E-04	5.27E-04	2.42E-05	2.77E-05	7.48E-04	2.54E-04	7.73E-05	1.05E-04	2.46E-04
CR=6	SVR	0.86	2.3	0.18	0.21	2.84	1.51	0.58	0.86	1.55
	Network extrapolation	8.60E-05	3.62E-04	2.01E-05	2.20E-05	4.96E-04	1.69E-04	6.36E-05	8.44E-05	1.74E-04
CR=8	SVR	0.65	1.72	0.14	0.16	2.13	1.14	0.45	0.65	1.18
	Network extrapolation	7.25E-05	2.69E-04	1.79E-05	2.03E-05	3.87E-04	1.41E-04	5.93E-05	7.31E-05	1.44E-04
CR=10	SVR	0.53	1.38	0.12	0.13	1.71	0.91	0.36	0.53	0.95
	Network extrapolation	6.93E-05	2.24E-04	1.81E-05	1.90E-05	3.19E-04	1.20E-04	5.93E-05	1.33E-04	1.23E-04

TABLE IV: Time (in seconds) taken by server to perform predictions (30 minute prediction horizon).

Region	D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station	799	2194	147	171	2647	1434	558	803	1464
UI refresh time	0.18	0.29	0.10	0.09	0.32	0.25	0.17	0.15	0.23
CR=2	22	170	0.65	0.71	173	80	20.7	22.5	85
CR=4	12.5	83	0.82	0.84	89	40.8	11.5	12.8	42.9
CR=6	8.5	56	0.56	0.6	60	28.9	7.5	8.9	30.8
CR=8	6.2	42.5	0.45	0.48	47	21.2	5.5	6.9	22.3
CR=10	5.64	34	0.13	0.14	38	17	4.6	5.9	18

TABLE V: Time (in seconds) taken by mobile application to perform network extrapolation and refresh User Interface.

Region		D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station		799	2194	147	171	2647	1434	558	803	1464
Traditional SVR		2.96	7.11	0.65	0.75	8.77	5.55	2.24	3.21	6.05
CR=2	Extrapolation on Server	1.69	3.82	0.41	0.44	4.98	3.14	1.25	1.75	3.19
	Extrapolation on Mobile	23.69	173.82	1.11	1.12	177.98	83.14	21.95	24.25	88.19
CR=4	Extrapolation on Server	0.95	2.05	0.27	0.28	2.58	1.66	0.7	0.95	1.69
	Extrapolation on Mobile	13.45	85.05	1.091	1.12	91.58	42.46	12.2	13.75	44.59
CR=6	Extrapolation on Server	0.69	1.47	0.21	0.22	1.81	1.17	0.53	0.69	1.2
	Extrapolation on Mobile	9.19	57.47	0.772	0.82	61.81	30.07	8.03	9.59	32
CR=8	Extrapolation on Server	0.57	1.17	0.19	0.19	1.44	0.95	0.44	0.56	0.96
	Extrapolation on Mobile	6.77	43.67	0.64	0.67	48.44	22.15	5.94	7.46	23.26
CR=10	Extrapolation on Server	0.5	1	0.18	0.17	1.22	0.81	0.39	0.48	0.82
	Extrapolation on Mobile	6.14	35	0.31	0.31	39.22	17.81	4.99	6.38	18.82

TABLE VI: Aggregate Time (in seconds) of all three proposed approaches(5 minute prediction horizon).

Region		D3	D4	D5	D6	D7	D8	D10	D11	D12
Number of station		799	2194	147	171	2647	1434	558	803	1464
Traditional SVR		5.03	14.04	1.03	1.2	17.01	9.23	3.65	5.24	9.56
CR=2	Extrapolation on Server	2.66	7.19	0.59	0.66	8.88	4.84	1.93	2.71	4.79
	Extrapolation on Mobile	24.66	177.19	1.29	1.34	181.88	84.84	22.63	25.21	89.79
CR=4	Extrapolation on Server	1.45	3.72	0.36	0.4	4.58	2.52	1.06	1.43	2.56
	Extrapolation on Mobile	13.95	86.72	1.181	1.24	93.58	43.32	12.56	14.23	45.46
CR=6	Extrapolation on Server	1.04	2.59	0.28	0.3	3.16	1.76	0.75	1.01	1.78
	Extrapolation on Mobile	9.54	58.59	0.842	0.9	63.16	30.66	8.25	9.91	32.58
CR=8	Extrapolation on Server	0.83	2.01	0.24	0.25	2.45	1.39	0.62	0.8	1.41
	Extrapolation on Mobile	7.03	58.01	0.802	0.85	62.45	30.29	8.12	9.7	32.21
CR=10	Extrapolation on Server	0.71	1.67	0.22	0.22	2.03	1.16	0.53	0.68	1.18
	Extrapolation on Mobile	6.35	35.67	0.35	0.36	40.03	18.16	5.13	6.58	19.18

TABLE VII: Aggregate Time (in seconds) of all three proposed approaches (30 minute prediction horizon).

- [5] Z. Liang and Y. Wakahara, "City traffic prediction based on real-time traffic information for intelligent transport systems," in *ITS Telecommunications (ITST), 2013 13th International Conference on*. IEEE, 2013, pp. 378–383.
- [6] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Route travel time estimation using low-frequency floating car data," in *2013 16th International IEEE Conference on Intelligent Transportation Systems*, vol. 8, 2013.
- [7] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 929–936.
- [8] E. J. Horvitz, J. Apacible, R. Sarin, and L. Liao, "Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service," *arXiv preprint arXiv:1207.1352*, 2012.
- [9] S. Diewald, A. Möller, L. Roalter, and M. Kranz, "Driveassist-a v2x-based driver assistance system for android," in *Mensch & Computer Workshopband*, 2012, pp. 373–380.
- [10] T. Toledo and R. Beinaker, "Evaluation of the potential benefits of advanced traveler information systems," *Journal of Intelligent Transportation Systems*, vol. 10, no. 4, pp. 173–183, 2006.
- [11] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1462–1468, 2012.
- [12] S. Amin, S. Andrews, S. Apte, J. Arnold, J. Ban, M. Benko, R. M. Bayen, B. Chiou, C. Claudel, C. Claudel *et al.*, "Mobile century using gps mobile phones as traffic sensors: A field experiment," 2008.
- [13] J. Steenbruggen, M. T. Borzacchiello, P. Nijkamp, and H. Scholten, "Mobile phone data from gsm networks for traffic parameter and urban spatial pattern assessment: a review of applications and opportunities," *GeoJournal*, vol. 78, no. 2, pp. 223–243, 2013.
- [14] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.
- [15] J. Readles, F. Calabrese, A. Sevtsuk, and C. Ratti, "Cellular census: Explorations in urban data collection," *Pervasive Computing, IEEE*, vol. 6, no. 3, pp. 30–38, 2007.
- [16] J. Jariyasunant, B. Kerkez, R. Sengupta, S. Glaser, and A. Bayen, "Mobile transit trip planning with real-time data," 2011.
- [17] K. Erhardt, "Development of a navigation solution for the android based driver assistance system driveassist," 2012.
- [18] A. Anwar, "Changinow: a mobile application for efficient taxi allocation at airports," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.
- [19] N. Mitrovic, M. T. Asif, J. Dauwels, and P. Jaillet, "Compressed prediction of large-scale urban traffic," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, oct. 2014.
- [20] C. D. of Transportation. (2014) Caltrans performance measurement system. [Online]. Available: <http://pems.dot.ca.gov/>
- [21] N. Mitrovic, M. T. Asif, U. Rasheed, J. Dauwels, and P. Jaillet, "CUR decomposition for compression and compressed sensing of large-scale traffic data," in *Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on*, oct. 2013, pp. 1475–1480.
- [22] M. T. Asif, J. Dauwels, C. Goh, A. Oran, E. Fathi, M. Xu, M. Dhanya, N. Mitrovic, and P. Jaillet, "Spatiotemporal patterns in large-scale traffic speed prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–11, 2013.
- [23] Esri. (2014) Arcgis runtime sdk for android. [Online]. Available: <https://developers.arcgis.com/android/>
- [24] Arcgis world street mapserver. [Online]. Available: http://server.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer
- [25] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 871–882, 2013.
- [26] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [27] B. Scholkopf, P. Bartlett, A. Smola, and R. Williamson, "Shrinking the tube: a new support vector regression algorithm," *Advances in neural information processing systems*, pp. 330–336, 1999.
- [28] J. Dauwels, S. Kannan, R. Ramasubba, and A. Cichocki, "Near-lossless multi-channel EEG compression based on matrix and tensor decompositions," *Biomedical and Health Informatics, IEEE Journal of*, vol. PP, no. 99, p. 1, 2012.