SpeedPro: A Predictive Multi-Model Approach for Urban Traffic Speed Estimation

Chinmaya Samal, Fangzhou Sun, Abhishek Dubey
Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA
{chinmaya.samal.1,fangzhou.sun,abhishek.dubey}@vanderbilt.edu

***Abstract*—Data generated by GPS-equipped probe vehicles, especially public transit vehicles can be a reliable source for traffic speed estimation. Traditionally, this estimation is done by learning the parameters of a model that describes the relationship between the speed of the probe vehicle and the actual traffic speed. However, such approaches typically suffer from data sparsity issues. Furthermore, most state of the art approaches does not consider the effect of weather and the driver of the probe vehicle on the parameters of the learned model. In this paper, we describe a multivariate predictive multi-model approach called SpeedPro that (a) first identifies similar clusters of operation from the historic data that includes the real-time position of the probe vehicle, the weather data, and anonymized driver identifier, and then (b) uses these different models to estimate the traffic speed in real-time as a function of current weather, driver and probe vehicle speed. When the real-time information is not available our approach uses a different model that uses the historical weather and traffic information for estimation. Our results show that the purely historical data is less accurate than the model that uses the real-time information.**

***Keywords*—Transit vehicles; Traffic speed; Cluster**

I. Introduction

Emerging trends and challenges. In recent years, almost every city in the United States has an increase in traffic congestion [1], which has a great impact on commuter's daily life. A report [2] in 2015 revealed that on average, urban commuters in the U.S. spent about 6.9 billion hours a year stuck in traffic jams and spent $160 billion on the extra cost of gas. Leveraging the recent advances in technology and research to create traffic monitor systems is important for both commuters and city planners. Commuters can use the traffic congestion information in real-time and change their travel plan to avoid the traffic jams. City planners can optimize the public transportation schedules, and road networks by analyzing the collected traffic data and identify patterns.

The state of the art in estimating traffic uses static sensors such as loop detectors [3], traffic cameras [4], microwave [5]. Such technologies are mostly managed by city transportation agencies and it is infeasible to install these sensors to cover the entire road network for traffic state estimation due to the cost and technical constraints [3]. Cell Phones can be used to estimate traffic state [6]. Almost everyone is equipped with cell phones, but due to privacy concerns, this data is not readily available. Data from private sources such as HERE API [7] can be used for travel time estimation on urban streets. However, resolution, the frequency of update and cost limit collection and purchase of such data for the entire transportation network every hour/day/year.

Therefore, public transportation like buses when equipped with GPS can be used as an alternate source of travel time information, especially where no traffic detector is installed [8], [9]. The use of buses as probe vehicles adds little or no financial burden to a transit agency because most buses operating on urban streets are equipped with GPS units for tracking and predicting bus arrival times. Further, a large number of buses run on the most used arterial streets and generally have higher frequencies during peak periods [10].

However, bus speeds do not entirely represent average traffic speeds. Buses must stop to pick up and drop off passengers, must follow a schedule, and have different acceleration and deceleration profiles [11]. For major road segments, the estimation gets more complicated due to traffic signals and interruption from traffic as well as other influential factors such as weather conditions, accidents, etc. Most of the road segments in a city are not covered by bus routes or frequency of the bus is low in some route segments, so there is huge data sparsity issue.

Contributions. In this paper, we explore the feasibility of using weather and historical traffic data that is often available from planning organizations to solve the data sparsity issue. We also improve the prediction accuracy when the probe GPS information is available by first classifying available training data into different clusters and then learning a separate model per cluster. Our results indicate that using this multi-model approach improves the prediction accuracy. These results are codified in the **SpeedPro** [12] toolchain.

Paper outline. Section II is an overview of the current literature on traffic speed estimation. Section III describes the data used by our models. Section IV-A describes the motivations. Section IV-B presents the details of our modeling approach. Section IV-C shows our cluster-based prediction approach. Section IV-D presents an empirical validation of our prediction model. In Section V, we discuss our approach and future work. Concluding remarks are present in Section VI.

II. Related work

Numerous studies have been conducted to develop models and algorithms to estimate traffic speed from dynamic sensors such as Global Positioning System (GPS). Cathey et al. [13] presented a framework that uses Kalman filter to estimate transit vehicle state and determine traffic speeds and times along freeways. Dailey et al. [14] also used Kalman filters to

978-1-5090-6517-2/17/$31.00 ©2017 IEEE

estimate the vehicle dynamical state and used AVL-equipped vehicles as traffic probe sensors. Aslam et al. [15] used taxi probes and followed logistic regression model for estimating traffic volumes or speeds for regular drivers. Hofleitner et al. [16] used Dynamic Bayes Network to learn the dynamics of arterial traffic from probe data.

A bus can be used instead of taxis as traffic probe. However, bus speeds do not entirely represent general traffic speeds ([10], [11]) and there are studies which measure the accuracy of using a bus as probe vehicle instead of taxi ([8], [9]). Carli et al. [17] used a bus as a probe and presented an algorithm for the automated analysis and evaluation of traffic congestion in urban areas. Denaxas et al. [18] estimated traffic speed on the urban road network by solving a linear system of equations that associates the successive GPS position data for each probe vehicle with the distances traveled on the road network. Weng et al. [19] estimated the bus travel speed by making use of the real-time GPS data of bus with bus route data and matching GPS points with the Geographic Information System (GIS) map.

Widhalm et al. [20] presented a method for faulty traffic sensor detection using Floating-Car Data (FCD). A nonlinear regression model was trained using the flow data from traffic sensors and estimated speed from FCD. However, they rely on data from regular cars, not transit vehicles. Since the distribution of probe vehicles over space and time is uneven, there are some studies ([21], [22]) about solving the problem of missing data from probe vehicles. Shan et al. [23] used an online method based on multiple linear regression models, in which the information from both time and space domain is obtained to estimate missing data.

A consistent thread in these studies is that they only use probe vehicles to estimate traffic speed. For example, when they analyze a particular road segment or street, they only consider the probe vehicles and external factors such as weather, driver is not taken into account. Weather conditions have an impact on the travel time of bus [24] and speed of a vehicle varies with different drivers. Also, there is huge data sparsity issue when we use GPS-equipped vehicles. Since the number of taxis and buses in an area is limited many road segments/times have low-frequency data or no data at all. This missing or sparse data leads to inaccurate or even no measurements for these roads [25].

Our approach as discussed earlier first identifies similar clusters of operation from the historic data that includes the real-time position of the probe vehicle, the weather data, and anonymized driver identifier, and then uses these different models to estimate the traffic speed in real-time as a function of current weather, driver and probe vehicle speed. Using clusters to group bus and weather data with similar characteristics, gave a better prediction as shown in our results in Figure 3. When the real-time information is not available our approach uses a different model that uses the historical weather and traffic information for estimation.

III. System Model and Data Specification

In our model, we divide the transit network of Nashville into multiple routes. Each route can be further divided into multiple route segments and each segment can be shared by multiple transit bus having unique Trip ID. For getting a shared segment on a road we are using shared route segment network generation algorithm [26]. All the buses are GPS-enabled, so it records its location with a certain frequency.

TABLE I: Real-time and Static Datasets Collected in the System.

Bus Schedules	
Format	Static GTFS
Source	Nashville MTA
Update	Every public release
Size	78.4 MB (09/15/2016)

Real-time Transit	
Format	Real-time GTFS
Source	Nashville MTA
Update	Every minute
Size	451 GB

Weather	
Format	JSON
Source	Dark Sky API
Update	Every 5 minute
Size	40.9 MB

Shared Route Segments	
Format	JSON
Source	Transit-Hub [26]
Update	Every GTFS version
Size	37.3 MB

Traffic	
Format	JSON
Source	TDOT
Update	Every public release
Size	6.03 GB

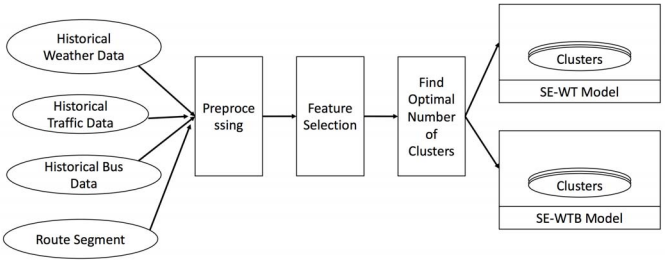

Fig. 1: Framework for SpeedPro Model Construction.

We have been collaborating with Nashville Metropolitan Transit Authority (MTA) to access the bus schedules, driver information and real-time bus data feed in Nashville. We are also integrating data from other sources, such as Dark Sky API for weather data and Tennessee Department of Transportation (TDOT) for getting historical traffic data in the city. It should be noted that we need historical traffic data for all road segments to label each data point while training the models. But for testing and prediction, we only rely on bus and weather data. For getting a shared segment on a road we are using shared route segment network generation algorithm [26], which uses static GTFS data [27] from Nashville MTA. Table I shows details about our data sources.

It is important to point out the key challenges that we encountered while processing the data. The most important challenge was to combine data from different sources which have different sample rates and location. Since sample rate and location of bus data, traffic data and weather data varies, we had to put certain thresholds on time and distance till which the data is acceptable. For example, if there are no weather or traffic data at a certain time stamp or location, then we consider the data available within last 5 minutes or within a radius of 0.3 miles. So, we assume that the data remains constant until

Feature	Acronym	Description
Weekday	WD	Day of the week
Hour	H	Hour of the day
Driver ID	D	Driver who was driving the bus
Maximum Speed	MS	Maximum speed of buses on the segment in last 5 minutes
Wind Speed	WS	Wind speed
Nearby Storm Distance	NSD	Nearby Storm Distance
Visibility	V	Visibility

TABLE II: Features considered in our model

a certain threshold time or distance.

Each route segment can be shared with multiple routes. So, it's difficult to merge the high volume of data needed to be handled carefully in order to match the route segments with their corresponding trips. We collected and processed data from December 12, 2016, to January 8, 2017, for multiple route segments in Nashville, which is then used by our prediction models to estimate traffic speeds for any route segment.

IV. Our Approach

In this section we present our toolchain **SpeedPro** [12], to estimate traffic speed. First, we describe our motivation for building SpeedPro. Then, we specify data used for building SpeedPro and the process used for building predictive models for SpeedPro. Finally, we validate SpeedPro and analyze its prediction results. Figure 1 shows our framework for SpeedPro and acronyms for the features used in the framework are described in Table II.

A. Motivation

To show our motivation we build two traffic speed estimation models used in SpeedPro, based on large-scale empirical data we collected in Nashville. The first model is called SE-WT (Speed Estimator-Weather with Traffic data), which consists of historical traffic data and weather data, which we collected with the help of Dark Sky API [28]. The second model is called SE-WTB (Speed Estimator-Weather with Traffic and Bus data) which consists of real-time GTFS bus data, historical traffic data along with weather conditions when the real-time bus data is recorded. Both these models are based on simple Random forests [29] and are used to infer real-time traffic speeds. In Figure 2, we compared these two models to the real-time traffic speed, from 4:30 a.m. to 11:00 p.m. on one day on West End Avenue which is a major road segment in Nashville. We chose this road segment because it's among the busiest roads in Nashville and has a high frequency of bus data available.

In general, the SE-WTB model has data sparsity issues. As shown in Figure 2, from 4:30 a.m. to 11:00 p.m, there was a total of 1251 readings, among which SE-WT have 1204 readings and SE-WTB have 47 readings, that is 96.25% and 3.75% respectively. (i) SE-WTB has a major data sparsity issue during the early morning and late night when there are no buses available. It underestimates the speed during the daytime because transit buses stop frequently at bus stops to pick up passengers which sometimes involve long wait times. As per our analysis, we get an RMSE error in the range of 4.0 to 4.5 miles/hr when we use SE-WTB model (ii) SE-WT has frequent data available, but it's not a direct model to estimate speed, because we only label historical traffic speed with weather data point at a particular time, place and hence, using only weather data to infer traffic speeds is not accurate. As per our analysis, we get an RMSE error in the range of 5.2 to 5.8 miles/hr when we use only SE-WT model.

A seemingly promising solution is to integrate these two models so that SE-WT model can complement the data sparsity issue arising from SE-WTB model. In this work, we combine both models and propose a cluster-based prediction model for traffic speed estimation.

B. Modeling Approach

In this section, we show how the model is clustered using k-means algorithm [30]. Each cluster is trained using Random Forests, which fall under the regression and tree-based family of models and finally, we provide the results of the model in terms of predictive accuracy. We provide a brief description of the features considered and their corresponding acronyms used to present the predictors in our models in Table II.

1) Preprocessing: After getting data from various feeds, we preprocessed it, so that it can be used in our model. Since we are using Random forests for training our model, we didn't need scaling, because the nature of Random Forests is such that convergence and numerical precision issues, which normally affects the algorithms used in logistic and linear regression, as well as neural networks, aren't so important. Because of this, we didn't need to transform variables to a common scale like we do in SVM [31], Neural Networks [32]. However, we scaled the features because if there are some features, with a large size or great variability, these kinds of features will strongly affect the clustering result of k-means. So we standardized features of datasets used for building SE-WT and SE-WTB models, by removing the mean and scaling to unit variance.

After scaling features for each model, we label the data points with the historical traffic speed. For SE-WTB model we use the bus, weather data and label them with historical traffic speed. While for SE-WT model we only use weather data and label them with historical traffic speed at that particular time of the day and location. It should be noted that we only need to label each data point with historical traffic speed while training the models. But for testing and prediction, we only rely on bus and weather data.

However, not all features are important for building clusters. So in next step, we identify the important features needed for building clusters for SE-WT and SE-WTB models of SpeedPro.

2) Feature Selection: To identify the important features of SE-WT and SE-WTB dataset for clustering, we used Recursive feature elimination (RFE) [33] using Random Forest Regression as a model. The underlying principle is that the algorithm first fits the model to all predictors. Then, each predictor is ranked using its importance to the model. Let S be a sequence

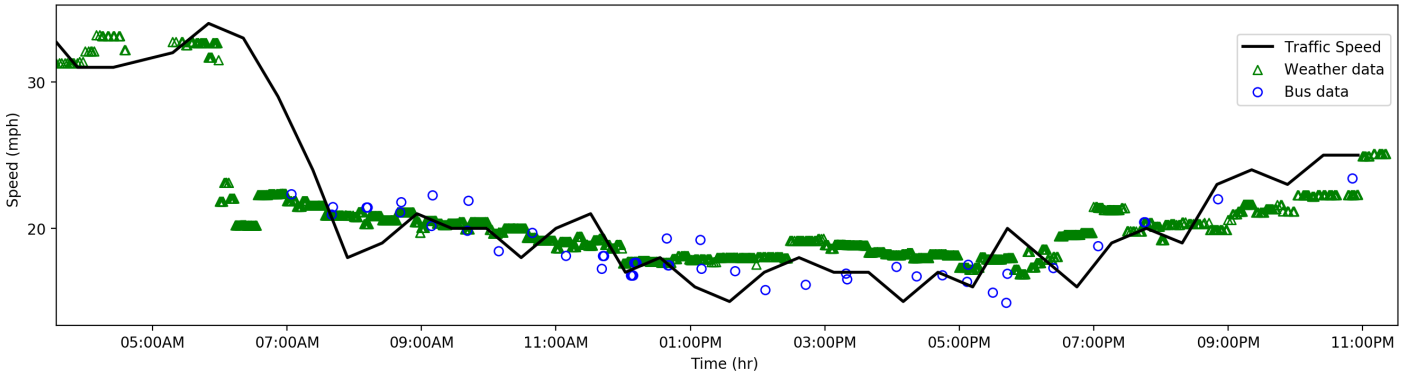

Fig. 2: Traffic Speed Estimation by SE-WT & SE-WTB models for a segment. Both these models are based on simple Random Forests.

Feature	Selected	Rank
Weekday	False	3
Hour	True	1
Driver ID	True	1
Maximum Speed	True	1
Wind Speed	True	1
Nearby Storm Distance	True	1
Visibility	False	2

Feature	Selected	Rank
Weekday	False	3
Hour	True	1
Wind Speed	True	1
Nearby Storm Distance	True	1
Visibility	False	2

TABLE III: Features considered in SE-WTB model (left) and SE-WT model (right)

of ordered numbers which are candidate values for the number of predictors to retain (S_1>S_2,...). At each iteration of feature selection, the S_i top-ranked predictors are retained, the model is refit and performance are assessed. The value of S_i with the best performance is determined and the top S_i predictors are used to fit the final model.

Table III shows the important features and ranks selected by RFE for the datasets used to build SE-WTB and SE-WT models. For SE-WTB model, Hour, Driver ID, Maximum Speed, Wind Speed and Nearby Storm Distance are considered, while for SE-WT model, Hour, Wind Speed and Nearby Storm Distance are considered by RFE. So, only these features will be used for building clusters. It should be noted that we only considered the features which have a rank of 1. Hence, Visibility and Weekday are not selected.

3) Finding Optimal number of clusters: After identifying the features needed to build clusters, we need to find the optimal number of clusters for each model. We used a weighted silhouette value to score each possible model. Silhouette analysis compares each data's similarity with its assigned cluster to its similarity to the next most similar cluster [34]. Formally, for each object i, let $a(i)$ be the average dissimilarity of i with all data within the same cluster, and $b(i)$ the lowest average dissimilarity of i to any cluster of which i isn't a member. The silhouette value of i is:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{1}$$

which produces silhouette values in the range $-1 \leq s(i) \leq 1$, where a high value indicates that the incident is well matched with its assigned cluster, while a low value indicates it is more similar to objects in its neighboring cluster. Finding the average silhouette score across all objects shows how well the objects are clustered in general:

$$\frac{1}{n} \times \sum_{i \in dataObjects} s(i) \tag{2}$$

where n is the total number of objects being clustered. After finding a set of silhouette scores for a different number of clusters for each model, we select the number of clusters with maximum silhouette score as an optimal number of clusters.

For SE-WTB model, we found the optimal number of clusters to be 5 and for SE-WT model the optimal number of clusters is 3. Then we used k-means algorithm [30] to create the optimal number of clusters for each model based on the features identified in the previous step for each model.

C. Prediction Approach

The next step is to create predictive models for each cluster and use both for traffic speed estimation. We used Random Forests [29] for our regression problem to train each cluster in each model and learn their predictive model.

Random forests [29] is an ensemble learning method for classification and regression. It is an approach where a number of decision trees are constructed and voting is done to define the best classifier. The underlying principle is that a group of weak learners can be combined to form a strong learner. In a decision tree for regression, the outcome variable is fitted for a regression model using each predictor.

Prediction: Figure 4 shows prediction approach used in SpeedPro. Given a particular route segment, for each data in real-time feed, SpeedPro server first checks if real-time probe data is available for that route segment. If true, then we proceed with SE-WTB model, or else we proceed with SE-WT model to predict using only weather data. Then, we calculate distance of data to each cluster centers of either SE-WTB or SE-WT model as found by k-means algorithm in previous step. Using the distance, we calculate the probability of data to each cluster centers such that, the greater the distance of data to a cluster

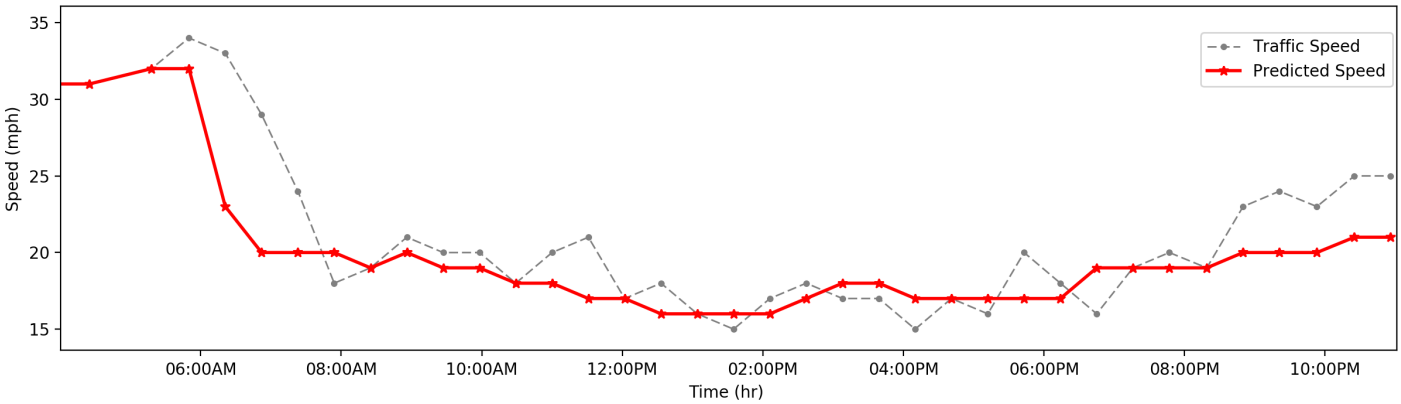

Fig. 3: Traffic Speed Estimation by cluster based model for a segment on January 9, Monday.

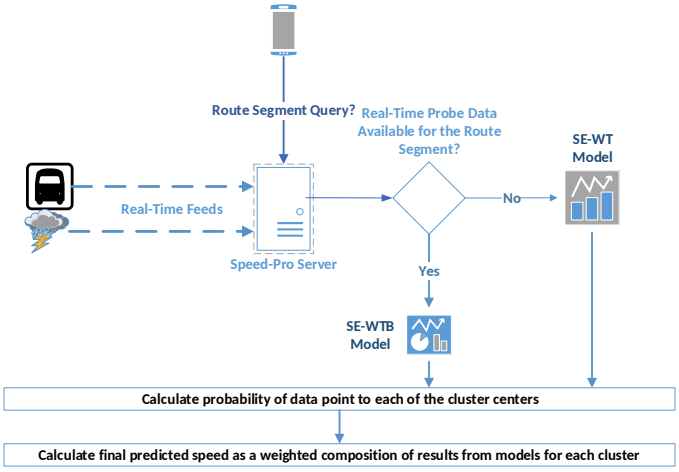

Fig. 4: SpeedPro Prediction approach.

center, the smaller is the probability. Then, using the test data we predict speed from prediction model of each cluster. For example, if p_1, p_2, p_3 are the probabilities that the data belongs to center of $cluster_1$, $cluster_2$, $cluster_3$ respectively and s_1, s_2, s_3 are the predicted speed from $cluster_1$, $cluster_2$, $cluster_3$ respectively, then the final predicted speed $\hat{s}$ is calculated as follows:

$$\hat{s} = p_1 \cdot s_1 + p_2 \cdot s_2 + p_3 \cdot s_3 \tag{3}$$

Performance: To assess the performance of the model, we consider the Root Mean Squared Error (RMSE) to assess the predictive accuracy. RMSE calculates the square root of the average of the squared differences between the predicted traffic speed ($\hat{y}_i$) and observed traffic speed (y_i) for new data points, this metric is in miles/hr, which is the unit of the estimated traffic speed.

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2} \tag{4}$$

D. Validation

To determine the accuracy of our cluster-based prediction model at estimating the traffic speed, we compared its result to validation set. The validation set consists of bus data and historical traffic data collected on January 9, Monday. We averaged the results over a 30-minute interval. It should be noted that we have weather information for all road segments used in our study, so even if we don't have bus data for a road segment, we use SE-WT model which was trained using historical weather and traffic data. We ran the clustering analysis for the given day and the result is shown in Figure 3. We got an RMSE error in the range of 2.9 to 3.3 miles/hr. Note, that we still experience RMSE error in range of 4 miles/hr on average during early morning, from 4 a.m to 8 a.m and during late night, from 8 pm to 11 p.m. This inaccuracy is expected because there is no bus and driver data available during those times, so the model relies only on SE-WT model. However, it should be noted that our model has better accuracy than what we would have got if we would have used simple Random forest model as mentioned in Section- IV-A. Our model predicted better in the daytime because there were more bus data available then.

V. Discussion

Predictive models used in SpeedPro is based on bus data from Nashville MTA and historical traffic data collected by Tennessee Department of Transportation (TDOT). However, SpeedPro is location-agnostic. If bus data, driver information of the buses, weather data and historical traffic data are available, then it can be used in any city to estimate traffic speed. Real-time traffic speed can be accessed by using HERE API and other private data sources, but time, manpower, and cost deter or limit collection and purchase of such data for the entire transportation network every hour/day/year. Our motivation for building SpeedPro was to estimate real-time traffic speed accurately for the entire transportation network without inflicting any huge cost by using private data sources.

Even though SpeedPro is built for estimating traffic speed, it can be extended for analysis of traffic state such as latency, congestion of any road segment. This can help city planners in optimizing the public transportation schedules, and road

networks by analyzing the collected traffic data and identify patterns. We can also integrate population density information to understand it's effect on the traffic speed and increase the accuracy of prediction in case of missing data, where we are currently using only weather data now.

VI. Conclusion

We have demonstrated that by considering external factors such as weather, driver information and using cluster-based approach to building predictive models, a toolchain like SpeedPro can be created that accurately predicts traffic speed in both space and time. As part of our future work, we will use SpeedPro for (a) computing latencies for any road segment and (b) understand problems in multi-modal routing games using the latency functions.

Acknowledgments

This work is sponsored in part by the National Science Foundation under the award number CNS-1647015 and CNS-1528799 and in part by the Vanderbilt Initiative in Smart City Operations and Research, a trans-institutional initiative funded by the Vanderbilt University.

References

[1] TomTom, "Tomtom traffic index measuring congestion worldwide," 2017, [Online; accessed Apr-6-2017]. [Online]. Available: http://www.tomtom.com/en_gb/trafficindex/#/list

[2] I. Texas A&M Transportation Institute, "2015 urban mobility scorecard," 2015.

[3] S. Tang and F.-Y. Wang, "A pci-based evaluation method for level of services for traffic operational systems," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 4, pp. 494–499, 2006.

[4] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 425–437, 2008.

[5] J. Van Lint and S. P. Hoogendoorn, "A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, no. 8, pp. 596–612, 2010.

[6] K. Sohn and K. Hwang, "Space-based passing time estimation on a freeway using cell phones as traffic probes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 559–568, 2008.

[7] "Here traffic api," https://developer.here.com/rest-apis/documentation/traffic/topics_v6.1/flow.html.

[8] S. Tantiyanugulchai and R. L. Bertini, "Arterial performance measurement using transit buses as probe vehicles," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1. IEEE, 2003, pp. 102–107.

[9] S. S. Pulugurtha, R. K. Puvvala, R. C. Pinnamaneni, V. R. Duddu, and P. Najaf, "Buses as probe vehicles for travel time data collection on urban arterials," in *T&DI Congress 2014: Planes, Trains, and Automobiles*, 2014, pp. 785–793.

[10] P. Chakroborty and S. Kikuchi, "Using bus travel time data to estimate travel times on urban corridors," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1870, pp. 18–25, 2004.

[11] R. Hall and N. Vyas, "Buses as a traffic probe: Demonstration project," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1731, pp. 96–103, 2000.

[12] C. Samal, "Speedpro-cluster based predictive model to estimate traffic speed using bus as a probe vehicle," 2017, [Online; accessed April-09-2017]. [Online]. Available: https://github.com/visor-vu/thub-traffic-probe-sensor

[13] F. Cathey and D. Dailey, "Transit vehicles as traffic probe sensors," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1804, pp. 23–30, 2002.

[14] D. J. Dailey and F. W. Cathey, "Avl-equipped vehicles as traffic probe sensors," Washington State Department of Transportation, Tech. Rep., 2002.

[15] 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.

[16] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1679–1693, 2012.

[17] R. Carli, M. Dotoli, N. Epicoco, B. Angelico, and A. Vinciullo, "Automated evaluation of urban traffic congestion using bus as a probe," in *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 967–972.

[18] E. Denaxas, S. Mpollas, D. Vitsios, C. Zolotas, D. G. Bleris, G. M. Spanos, and N. P. Pitsianis, "Real-time urban traffic information extraction from gps tracking of a bus fleet," in *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2013 IEEE Symposium on*. IEEE, 2013, pp. 58–63.

[19] J. Weng, C. Wang, H. Huang, Y. Wang, and L. Zhang, "Real-time bus travel speed estimation model based on bus gps data," *Advances in Mechanical Engineering*, vol. 8, no. 11, p. 1687814016678162, 2016.

[20] P. Widhalm, H. Koller, and W. Ponweiser, "Identifying faulty traffic detectors with floating car data," in *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*. IEEE, 2011, pp. 103–108.

[21] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2289–2302, 2013.

[22] Z. Shan, Y. Xia, P. Hou, and J. He, "Fusing incomplete multisensor heterogeneous data to estimate urban traffic," *IEEE MultiMedia*, vol. 23, no. 3, pp. 56–63, 2016.

[23] Z. Shan, D. Zhao, and Y. Xia, "Urban road traffic speed estimation for missing probe vehicle data based on multiple linear regression model," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 118–123.

[24] A. Oruganti, F. Sun, H. Baroud, and A. Dubey, "Delayradar: A multivariate predictive model for transit systems," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1799–1806.

[25] P. Widhalm, M. Piff, N. Brändle, H. Koller, and M. Reinthaler, "Robust road link speed estimates for sparse or missing probe vehicle data," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 1693–1697.

[26] F. Sun, A. Dubey, J. White, and A. Gokhale, "Transit-hub: A smart public transportation decision support system with multi-timescale analytical services," *Cluster Computing*, 2017.

[27] Wikipedia, "General transit feed specification," 2015, [Online; accessed 31-January-2016]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=General_Transit_Feed_Specification&oldid=693322749

[28] "Dark sky api," https://darksky.net/dev/.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[30] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[31] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[32] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural network design*. Martin Hagan, 2014.

[33] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products," *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, pp. 83–90, 2006.

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