Real-time and Predictive Analytics for Smart Public Transportation Decision Support System

Fangzhou Sun, Yao Pan, Jules White, Abhishek Dubey
Institute of Software Integrated Systems
Department of EECS
Vanderbilt University, Nashville, TN, USA
{fangzhou.sun, yao.pan, jules.white, abhishek.dubey}@vanderbilt.edu

***Abstract*—Public bus transit plays an important role in city transportation infrastructure. However, public bus transit is often difficult to use because of lack of real-time information about bus locations and delay time, which in the presence of operational delays and service alerts makes it difficult for riders to predict when buses will arrive and plan trips. Precisely tracking vehicle and informing riders of estimated times of arrival is challenging due to a number of factors, such as traffic congestion, operational delays, varying times taken to load passengers at each stop. In this paper, we introduce a public transportation decision support system for both short-term as well as long-term prediction of arrival bus times. The system uses streaming real-time bus position data, which is updated once every minute, and historical arrival and departure data - available for select stops to predict bus arrival times. Our approach combines clustering analysis and Kalman filters with a shared route segment model in order to produce more accurate arrival time predictions. Experiments show that compared to the basic arrival time prediction model that is currently being used by the city, our system reduces arrival time prediction errors by 25% on average when predicting the arrival delay an hour ahead and 47% when predicting within a 15 minute future time window.**

I. Introduction

Emerging trends and challenges. Bus systems are the backbones of public transit services in many cities. With their high capacity and relatively low investment and operational costs, bus systems can reduce traffic congestion substantially as well as bring environmental benefits such as reducing energy consumption and air pollution [1]. However, one major issue preventing many people from choosing bus service for commuting and travelling is its unpredictability [2]. Buses can often show up late due to various reasons: traffic congestion, road construction, special events or bad weather. This uncertainty forces potential riders to opt for other modes of transportation.

Travel/arrival time prediction is one key research topic in intelligent transportation research [3], [4], [5]. Often, transit authorities use Automatic vehicle location (AVL) systems to monitor bus service status in order to provide information to city decision makers as well as commuters. The data collected provides the potential for more intelligent applications such as transit operation monitoring, smart trip planning, rough delay time estimation, at-stop displays, etc.

In this paper, we focus on delay prediction for midsize cities. Midsize cities with populations of 10,000s-100,000s [6] of citizens. According to the statistics [7], more than 280 cities in the United States fall into the midsize city category. Various statistical models have been applied to travel/arrival time prediction [3], [4], [5]. However, in most cases the system analyzed belonged to a large city with a large number of vehicle trips, generating a large dataset that was then used for analysis. Unlike large cities, midsize cities have limited public resources to invest in the transportation services, and moderate residential and employment density. As a result, the transit network is often not very dense and the vehicles are not scheduled very frequently. This reduces the amount of data that is available for creating prediction models, which can produce poor accuracy. In this paper, we shot that by using data available at the route segment level, it is possible to produce enough samples for more rigorous statistical analysis.

Contributions. This paper presents Transit-Hub, a decision support system that addresses the question of whether it is feasible to build a smart public transportation decision support system that can efficiently use utilize data from shared route segments to produce more accurate predictions. This paper's main contributions are as follows:

- We present a clustering model that learns bus performance patterns during different hours of the day and different days of the week.
- We describe a real-time vehicle schedule adherence and prediction model. This model can be also used for identifying arrival time outliers and anomalous operations.
- We empirically validate our approach using a real-world dataset and real-time transit feed from Nashville. The experiments show that data collected over a two-hour window is most suitable for real-time prediction. Our model provides a 25% reduction error in average arrival time prediction within the a 1-hour window and achieves a 47% improvement when predicting the delay for next 15 minutes compared to the model currently in use by the Nashville MTA.

Outline: Section II specifies the system model, our data sources and defines the delay. Section III provides related research. Section IV describes our key contributions to address the system's challenges, and the experimental evaluation of the

978-1-5090-0898-8/16/$31.00 ©2016 IEEE

system. Section VI describes how we integrate the models and describes the solution architecture of Transit-Hub. Section VII presents concluding remarks and future work.

II. System Model

Transit network in a city is broken up into multiple routes. Each route can be further divided into multiple transit stops. Unique paths between two transit stops are call route segments. A transit schedule consists of scheduled trips for each route. A trip is identified by the time that it starts from the origination bus stop. Given the finite number of transit vehicles, a bus often finishes one trip and is then immediately allocated to another trip. Therefore, delays in the systems can cascade across scheduled trips. Furthermore, traffic and other phenomena can add to delays, which is defined as the time between expected arrival at a stop and the time of actual arrival. The stops with a detailed record of departure and arrival time are called "time points".

A. Data Sources

We have collaborated with Nashville's Metropolitan Transit Authority (MTA) for accessing their static bus schedules, historical data set of time points all across the city and real-time transit data feeds. The data sources and their features are as follows.

- Bus scheduling dataset: static transit data in General Transit Feed Specification (GTFS) [8] format that presents the physical route layout, stop locations and static schedules.
- Time point dataset: historical data of the buses at time points[1], including bus ID, route ID, trip ID, actual departure and arrival time, dwell time.
- Real-time transit feeds: real-time updates of transit fleet in GTFS real-time format for the following feed types: trip updates, service alerts and vehicle positions. These feeds are established by streaming AVL data on operating buses.
- Crowd-sourced data feeds: the collected data from mobile apps include anonymized information about the location, when they get on/off the buses, their walking distances to bus stops, etc. All data are collected anonymously. It should be noted that this data set is not used in the analysis described in this paper.

Bus scheduling dataset (static GTFS) is updated only when MTA modifies its bus routes or schedule.; Historical time point dataset is collected by MTA when each monthly period ends. So at the end of each month, time point dataset is manually imported into our MongoDB database; The real-time transit data is collected from real-time feeds and persisted by the back-end server every minute

B. Delay Definition

We consider two types of delay metrics, a delay associated with a route segment and a delay associated with a time point. Consider two adjacent stops A and B, the time interval of scheduled arrival time at B and the scheduled arrival time at A is the normal travel time without delay. The travel time delay for route segments between any two adjacent stops $t_{route\ delay}$ can be calculated as follows: $(B^{act}_{arr} - A^{sch}_{dep}) - (B^{sch}_{arr} - A^{sch}_{arr})$, if $A^{act}_{arr} \leq A^{sch}_{arr}$ or $(B^{act}_{arr} - A^{act}_{arr}) - (B^{sch}_{arr} - A^{sch}_{arr})$, if $A^{act}_{arr} > A^{sch}_{arr}$, where act and sch in superscript indicates actual/scheduled time. And dep and arr in subscript indicates departure/arrival time. For example, $B^{act}_{dep} - A^{sch}_{dep}$ refers the actual departure time of time point B minus the scheduled departure time of time point A. Specific delay for a particular stop, say B is $t_{time\ point\ delay}$ is $B^{act}_{arr} - B^{sch}_{arr}$.

[1]A time point is a preidentified transit stop which has recorded arrival and department information per trip.

III. Related Work

A. Historical Delay Analysis Models

Many researchers have conducted studies that analyze the historical data of bus service to investigate factors that cause delay and affect bus service. Abkowitz et al. [9] found that trip distance, passenger activity and signalized intersections could greatly affect the mean and variance of bus running time. Kimpel et al. [10] analyzed the bus service performance and passenger demand using Tri-Met Bus Dispatch System data at time point level. They found that the delay variation at previous time points, passenger demand variation, speed and distance contribute to delay variations. They suggested that optimizing delay at early time points could improve service reliability. El-Geneidy et al. [11] investigated how reserved bus lane affect the running time delay and arrival time delay of other parallel routes.

B. Bus Service Quality Measurement

Researchers have defined several performance measures to quantify the quality of bus service. Sterman et al. [12] tested the inverse of the standard deviation of travel times to measure service reliability. Camus et al. [13] proposed a new service measure called weighted delay index. Saberi et al. [14] evaluated the existing reliability measures and defined an alternative metric at the stop level. Other researchers have presented systematic frameworks for bus service measurement. Lin et al. [15] created a quality control framework of Data Envelopment Analysis (DEA) that uses data from AVL devices to quantify route service reliability. Gilmore et al. [16] presented the integration of quantitative analysis tools and applied to public bus systems.

C. Delay Prediction Models

Travel time and arrival time variation were found to have a great impact on commuters' satisfaction [17]. In the past decade, numerous studies have been conducted to develop models and algorithms to predict bus travel delay and arrival delay. Abdelfattah et al. [18] developed linear and nonlinear regression models for predicting bus delay under normal conditions using simulation data. Williams and Hoel [19] found that daily traffic condition patterns are consistent across the weeks. Jeong et al. [20] presented a historical average model and found that the historical model was outperformed by other models because its prediction accuracy was limited by

the reliability of traffic patterns. Regression models measure various independent variables to predict a dependent variable. Patnaik et al. [4] used distance, number of passengers at stops, stop numbers, and weather conditions for multilinear regression models to predict bus arrival time. However, since the attributes in transit services are often not independent but correlated with each other, the performance of regression models will deteriorate as the dimension of the data increases. Machine learning models can deal with complicated relationships and noisy data. Elhenawy et al. [3] presented a data clustering and genetic programming approach to predict the travel time along freeways. Artificial neural network (ANN)[21], [22], [23] and support vector machine (SVM) [5], [24], [23], [25] are two most widely used machine learning models in bus time prediction. Kalman Filtering models rely on historical data and real-time data and have been employed extensively for bus time prediction [21], [26], [25].

D. Summary

Even though there are numerous models and algorithms developed for historical data analysis and bus time prediction, only few of them focus on using data of segments that are shared by multiple routes. Yu et al. [23] investigated the models of k nearest neighbours algorithm (kNN), SVM, ANN, and linear regression (LR) on arrival time prediction that used multiple routes' segment data. However, they only studied the peak hours' data and the short-term prediction. Bai et al. [25] presented a dynamic travel time prediction model based on SVMs and Kalman filtering-based algorithm with multiple bus routes. But Bai's research did not study the model's performance for different prediction periods and training data periods. Also in their system, long-term data analytics and short-term delay prediction were not integrated together.

IV. Learning the Transit Performance Model

We have developed a long-term analytics model that explores the historical bus delay patterns of arrival time delay at time points and travel time delay for all route segments. Our approach uses clustering methods which allows the decision support system to provide typical delay information clustered based on time of day and eventually other features such as weather to users and city planners.

A. Clustering Analysis

For each day of week, K-means algorithm [27] is used to cluster the delay data according to the delay and time in the day by minimizing the within-cluster sum of squares (WCSS).

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{1}$$

where μ_i denotes the mean of all points in cluster S_i.

Silhouete analysis [28] is a measurement of how close each point is within one cluster.

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2}$$

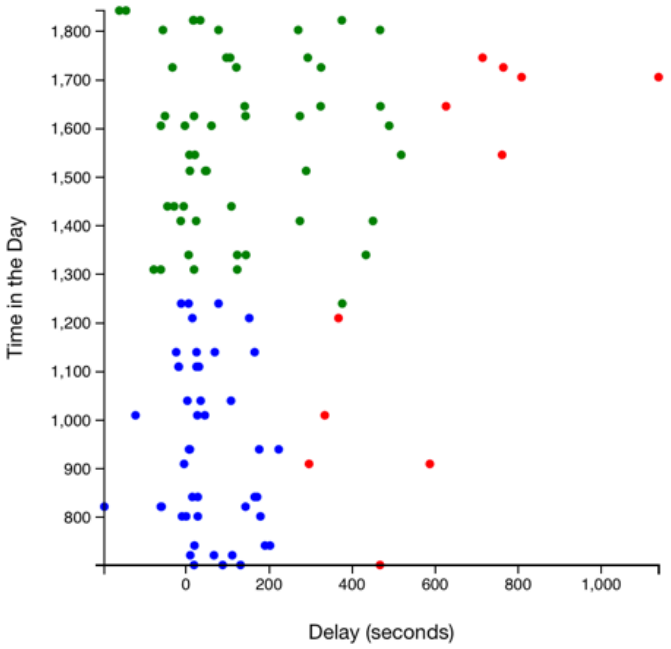

Fig. 1: Cluster historical delay data according to the delay and time in the day at time point "HRWB" on route 3. The figure shows that there are two active delay patterns. One before 1230 hours and the other after that.

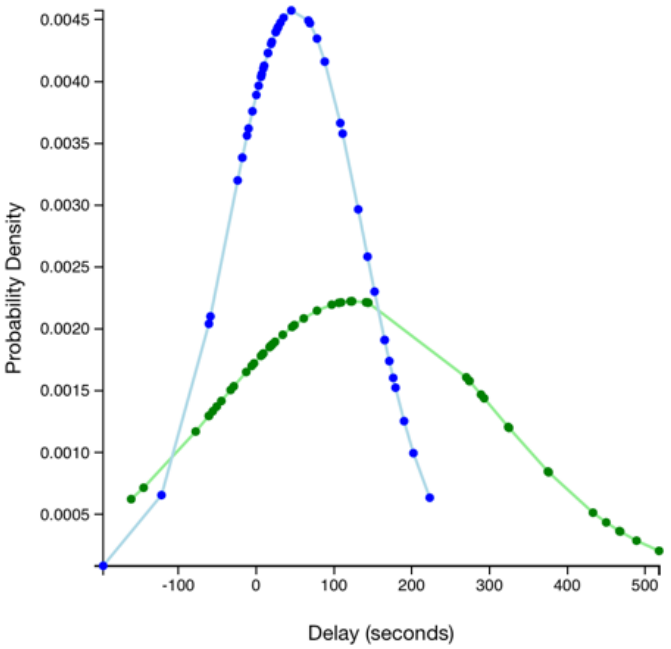

Fig. 2: Gaussian distribution of the clustered historical delay data at time point "HRWB" on route 3

where for each data point i in the cluster, a_i is the average distance between i and the rest of data points in the same cluster, b_i is the smallest average distance between data point i and every other cluster. We calculate the silhouette scores from 2 to 5 clusters using K-means algorithm to find the optimal number of clusters with the lowest silhouette score.

At the end of each month, the time point data is imported into the database. The data is then divided into 7 groups according to the day in the week. We then generate the clusters and normal distributions for all the route segments in each group. The clustered data and Gaussian distribution are then cached and persisted in the database. This ensures that we do not run clustering analysis every time we have to query the model.

Example Consider a time point "HRWB" on route 3 in Nashville. The historical data of bus arrival delay which we selected is for Tuesday, outbound direction, between August 1 2015 and August 31 2015 (107 points). Figure 1 plots the delay data in 24 hours. In the figure most of the points are roughly clustered into two groups (green and blue), one between 7 AM - 12:30 PM and another between 12:30 PM and 7 PM. This means there are two different delay patterns in the morning and in the afternoon.

B. Normality Test and Prediction.

Assuming that the historical delay data has a Gaussian distribution we perform normality test on each cluster that we get from the analysis in the previous step. From the distribution curve we can calculate long-term delay prediction confidence interval and provide the results to a mobile app and dashboard (these applications are described later in section VI-A).

Example: Doing the normality test on the clusters generated from the data describe in the previous example, we get the two Gaussian distributions in Figure 2. The cluster for the delay in the morning has a lower mean value (85.8s v.s. 196.7s) and a narrower Gaussian distribution curve, which indicates that for this time point on Tuesday, the route 3 buses are more likely to be on time in the morning than in the afternoon. In the morning the 95% confidence interval of delay is between 45.6s to 126s while in the afternoon the 95% confidence interval of delay is between 123.5s to 269.8s.

C. Outlier Analysis.

Outlier analysis is important for transit data analysis because it provides cleaner data for the normal distribution analysis and prediction. Furthermore, it helps to identify major sporting and special events, hazardous weather conditions, peak hour congestion, all of which could cause abnormal delays. By analyzing outliers, city decision makers can understand the factors that unusually affect the delay time.

To identify the outliers, Mahalanobis distances [29] are computed for the points in each Gaussian distribution. Mahalanobis distance measures the distance between a point x_i and a distribution by the following formula:

$$d_{\mu,\Sigma}x_i^2 = (x_i - \mu)^{'}\Sigma^{-1}(x_i - \mu) \tag{3}$$

where μ is the mean value and Σ is the covariance matrix for the delay vector.

Example: For the dataset discussed in the previous two examples, there are some outliers (red points) identified in Figure 1. These outliers happened mostly during the morning and evening rush hour. Our hypothesis is that in rush hour there are typically more passengers and more traffic congestion on the road. Since our back-end server is monitoring the real-time transit feeds, it also records in real-time about which trips have severe outliers and do not fit in the typical delay pattern, which can then be investigated.

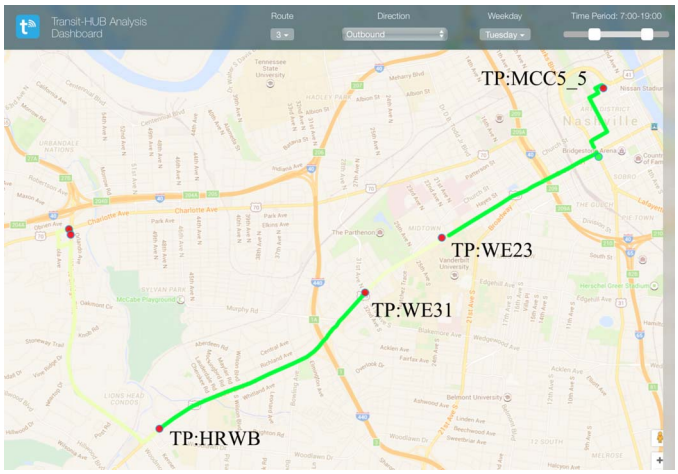

Fig. 3: Time points on route 3 in Nashville: the colors at time points indicate the arrival delay; the colors on route segments between time points indicate the travel delay. The bus in this route travels from MCC5 towards HRWB.

D. Bottleneck Identification

Once we have the mean delay patterns for all time points and all route segments, we can use them to identify the bottlenecks along the routes and take actions to optimize the route performance. As shown in Figure 6, there are three time points "MCC5_5", "WE23", "WE31" preceding to time point "HRWB" on route 3. We perform analytics on time point "WE23" and "WE31". The typical arrival delay for "WE23" on Tuesday afternoon is 114 seconds while for "WE31" is 195 seconds, considering the fact that typical delay for the succeeding time point "HRWB" on Tuesday afternoon is 196.7 seconds, which is statistically closer to 195 seconds. Therefore, we can conclude that the bus stops between "WE23" and "WE31" are the bottlenecks of route 3 which generate the delay. In Figure 3 the color of segments between adjacent time points shows the travel delay. Green means the delay is less than 100 seconds while yellow means delay is larger.

V. Integrating Real-time Data

As described earlier, the automated vehicle locators provide time stamped position for each vehicle in real-time. It also provides a basic additive arrival estimate using the the delay accrued till the previous stop. The resolution of the provided estimate is in minutes. However, unlike the time point data it does not include the actual arrival and departure time at each bus stop.

We use a two-staged Kalman filter model to integrate the data and analyze it. The first stage filter is the location filter, which uses the real-time feed data to better estimate the current

position of the vehicle. Then, we use a shared segment delay filter to update the current delay for the route segment.

A. Time Window Configuration.

We have two parameters for using the real-time data to predict bus delay time in the future, T_{past} and T_{future}. T_{past} is the length of the time window from which the real-time data is used; T_{future} is the upper bound of duration for which we predict the arrival and delay parameter. For example, if T_{past} is 3 hours and T_{future} is 30 minutes, it means the model is using real-time dataset of the past 3 hours to predict the delay up to 30 minutes in the future. Experiments that explore the relation between prediction accuracy and configuration of T_{past} and T_{future} is shown in Section V-F. By default, we use 2 hours as T_{past} and 15 minutes as T_{future}.

B. Location Filter.

Since rate at which the vehicle location is updated is not fixed and varies from several seconds to several minutes, we aggregate the collected data - the timestamped vehicle position array $[(t_1, d_1), ..., (t_{k-1}, d_{k-1}), ...]$ and then use it to estimate the bus locations. We assume that the following state transition model

$$\begin{pmatrix} \hat{d_k} \\ \hat{v_k} \end{pmatrix} = \phi_{k-1} \begin{pmatrix} d_{k-1} \\ v_{k-1} \end{pmatrix} + \omega_k \tag{4}$$

$$\phi_k = \begin{pmatrix} 1 & \Delta_t \\ 0 & 1 \end{pmatrix} \tag{5}$$

where the state variable v_k is the velocity at time step k. ω_k denotes the zero mean normal distribution noise with covariance Q_k; $\Delta_t = t_k - t_{k-1}$ is the update time interval.

The observation equation can be modeled as:

$$\begin{pmatrix} \tilde{d_k} \\ \tilde{v_k} \end{pmatrix} = \begin{pmatrix} d_k \\ v_k \end{pmatrix} + \nu_k \tag{6}$$

where variable z_k represents the observation of distance at time step k. ν_k represents the zero mean Gaussian distribution observation noise with covariance R_k. ω_k and ν_k are assumed to be independent.

C. Smoothing the Arrival Data.

The actual time that a bus arrives at a stop is not available in the real-time GTFS feed. Therefore, we estimate it using the data generated from the location filter.

From the static bus scheduling data, we get the distance array $[d_{stop_1}, ..., d_{stop_n}]$ for the bus stops along each route from its origination point. Then we use the timestamped vehicle position array $[(t_1, d_1), ..., (t_k, d_k), ...]$ to estimate the bus's arrival time at each bus stop.

$$t_{stop} = t_{k-1} + (t_k - t_{k-1}) \frac{d_{stop} - d_{k-1}}{d_k - d_{k-1}} \tag{7}$$

where $d_{k-1} <= d_{stop}$, $d_k > d_{stop}$, and t_{stop} denotes the estimated arrival time at the destination stop. This equation uses the average speed between d_{k-1} and d_k to estimate the time the bus reached the stop.

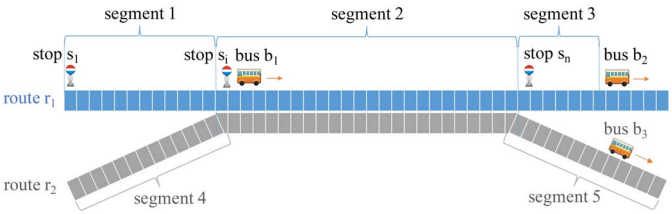

Fig. 4: Example of Using the Shared Route Segment's Delay Data to Predict a trip's delay on one route.

D. Shared Segment Delay Filter.

Once we know the estimated arrival time of the bus at a stop, we can use it to update the delay time for that stop. By using the estimated arrival time at the previous stop and the scheduled time for that route segment we can calculate the estimated delay per route segment. This estimate is then used using a Kalman filter which uses observations generated from the estimates of all buses traveling on that route segment. The state transition equations is modeled as:

$$x_k = x_{k-1} + \omega_{k-1} \tag{8}$$

where the state variable x_k denotes the delay time at time step k that needs to be predicted, ω_k denotes the zero mean normal distribution noise with covariance Q_k.

The observation equation can be modeled as:

$$z_k = x_k + \nu_k \tag{9}$$

where variable z_k represents the observation of delay at time step k. ν_k represents the zero mean Gaussian distribution observation noise with covariance R_k. ω_k and ν_k are assumed to be independent.

E. Using the Filters

An example of how the real-time prediction engine works is shown in Figure 4. Suppose a bus b_1 is running on route r_1 and it has just passed a bus stop s_i, now a user requests to predict the bus b_1's arrival time delay at stop s_n. From the real-time GTFS data feed we also know there are two preceding buses b_2 on route r_1 and b_3 on route r_2 that have traveled through the segment between stop s_i and stop s_n. Then the corresponding delay prediction work flow will be:

- Since the bus b_1 has already traveled from stop s_0 to stop s_i, the actual delay within this segment can be calculated from the result of bus b_1's position update Kalman filter.
- For the route segment between stop s_i and stop s_n, since this segment is shared by route r_1 and r_2, the preceding bus b_1's actual travel delay from b_1's position update Kalman filter and bus b_2's actual travel delay from b_2's position update Kalman filter should be inputted into the delay update Kalman filter of the route segment between stop s_i and stop s_n.
- The final arrival time delay for bus b_1 at stop s_n should be the sum of bus b_1's actual delay from stop s_0 to stop

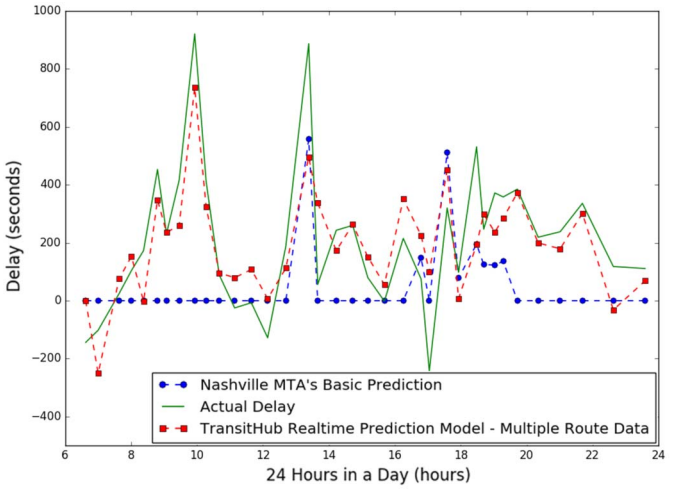

Fig. 5: The delay signal (actual arrival delay), prediction of MTA's basic model and our real-time prediction model.

Fig. 6: Route 3 and route 5 both start from downtown and have several shared route segments along the routes.

s_i and the predicted travel delay for segment between stop s_i and stop s_n. The system will use the delay from multiple segments, if required.

It should be noted that we are not only using the data of the buses on the same route that we are predicting, but also the data from buses from other routes that have traveled along th route segment. This allows to collect more delay data points and improve the prediction accuracy.

F. Results

We divide all the data into two subsets: the training set includes the static bus scheduling data and real-time vehicle position feed; the validating set includes the vehicle position feed data that has been continuously recorded by our back-end server. We use the training dataset to simulate the delay prediction for each trip in the month, and then validate the difference between the actual delay of the trip and the delay prediction made by our model and Nashville MTA's basic model. Nashville MTA's existing basic delay prediction model

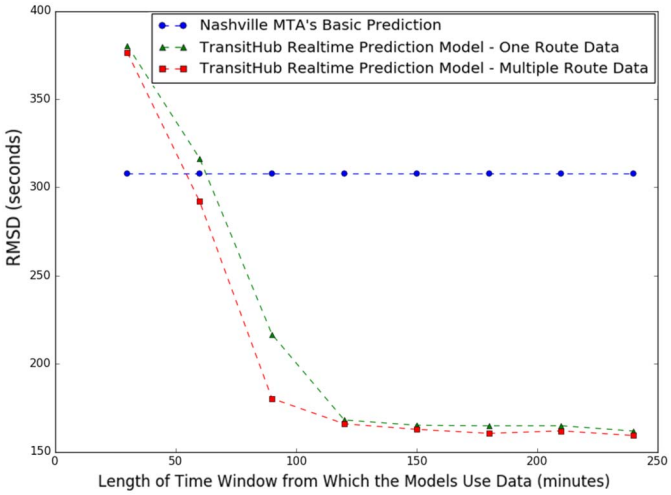

Fig. 7: RMSD of predicted delay vs length of window from which past data is used. In this experiment, T_{future} is configured to be 15 minutes and T_{past} varies.

periodically measures the difference between a bus's actual arrival time and scheduled arrival time at the latest time point that the bus passed, and use it for predicting delay time of all the rest bus stops along the route.

Route 3 and Route 5 are two of the main bus routes in Nashville. As shown in Figure 6, more than half of their route segments (MCC5_5-WE23, WE23-WE31 and WE31-HRWB) are shared by each other. In this experiment, we analyzed the arrival estimate filter for the bus stop 'HARBOSWM' (time point 'HRWB') of Route 3 on November 2 2015. Results are shown in Figure 5. We used the data collected from buses of both route 3 and route 5.

Figure 5 shows that our model's prediction curve roughly follows the actual delay signal while the basic model is only able to provide prediction between 1PM - 2PM and 4PM - 8PM. Since MTA's model only considers the delay of previous time points, the it predicts delay to be 0 when there is no delay data from the same route at the time that prediction is made. Our model collects data from Route 3 and Route 5. So when there is not enough data from Route 3, our model can still utilize Route 5's data of the shared segments for prediction.

G. Effect of length of past time window ($\mathbf{T_{past}}$) from which data is used

Figure 7 shows the root-mean-square deviation (RMSD) of delay prediction as the length of time window in the past from which we use the AVL data for real-time prediction. RMSD of delay prediction is calculated as:

$$RMSD = \sqrt{\frac{\sum(t_{arr}^{act} - t_{arr}^{pred})^2}{n}} \tag{10}$$

where variable t_{arr}^{act} represents the actual arrival time and t_{arr}^{pred} represents the predicted arrival time at the time point, n is the number of bus trips in the dataset.

The prediction performance of the model is expected to improve as more training data is included because Kalman Filtering model's accuracy relies on historical data. Since the

shared segments of multiple routes should provide more data than a single route in the same time period, we expect the model which uses shared segment's dataset to outperform the one using single route's data set.

The experiment results validate our hypothesis. (i) The basic prediction model's curve is a flat line because the model only considers current trip's delay at preceding time point. (ii) For the same length of time window, the filter that uses the shared segments' data has better performance than using single route's data. (iii) We can see from figure 7 that as the T_{past} increases from 30 minutes to 90 minutes, our model has a vast improvement in accuracy. However, the performance increase begins to taper out as the length of time window is further increased. If length of the time window is larger than 120 minutes, the curves of our model almost become flat lines. This indicates that only data within 120 minutes before the current time is important for real-time delay prediction. This is important because calculating and smoothing the delays from previous trips using Kalman Filters is expensive, especially when the computation is done for all the stops in the transit network. Using a smaller time window reduces the overhead.

H. Effect of length of prediction horizon ($\mathbf{T_{future}}$)

Figure 8 shows the delay prediction's RMSD as the prediction horizon increases to a future of 5 minutes to 120 minutes after current time. The RMSD is calculated the same as Equation 10. It illustrates that generally the prediction performances of both models worsen as the prediction is applied further in the future. For instance, the RMSD of our model using data from route 3 and 5 (red line with square markers) increases from 110 seconds to 372 seconds when prediction horizon is increased from 5 minutes to 120 minutes in the future. Compared to the RSMD of the basic model, w there is a 147s improvement for the 15 minute horizon, which is a 48% improvement. For the future horizon of an hour, our real-time prediction model reduces error by 25% on average compared to the basic model. Note that in this experiment, we used a T_{past} of 3 hours and 30 minutes. It should be noted that because the T_{past} is greater than 120 minutes, the prediction using multiple routes' data only slightly outperforms the one using single route's data, which is seen in the result.

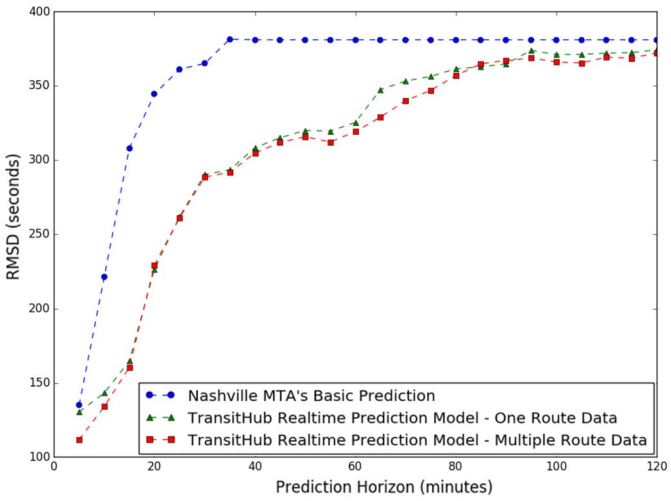

Fig. 8: RMSD of delay prediction as the prediction is applied further in the future time. In this experiment, T_{past} is configured to be 2 hours and T_{future} varies.

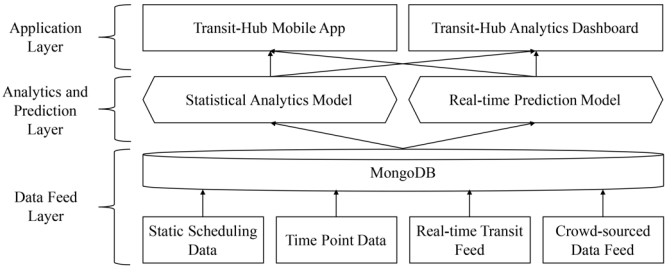

Fig. 9: The high-level architecture of Transit-Hub consists of three layers: (a) data feed layer; (b) analytics and prediction layer; (c) application layer

VI. System Integration

A. Applications

The Transit-Hub system is accessible to users by a mobile application that can be deployed on individual user's smart phones and an analysis dashboard that can be deployed online and accessed by local MTA analysts.

B. Deployment

Currently the analytics and prediction models are all deployed on a private cloud. The deployed system consists of three layers: data feed layer, analytics and prediction layer, and application layer, as shown in Figure 9. Data feed layer provides a reliable data feed mechanism for Transit-Hub system by integrating multi-source data and persisting them into MongoDB. At a global level, Data Feed Layer integrates static transit schedules, historical time point data set and real-time transit feeds. At the user level, Data Feed Layer anonymously collects planned trips, real-time locations, walking and transit distances.

C. Overall Workflow

When predicting the travel or arrival time delay for a specific trip on the same day, the system will first run the real-time prediction model to get the predicted delay at the stop, and then compare the difference between current time and the predicted time for the bus, if the time difference is smaller than the configured T_{future} (see section V-A), then the system will provide the predicted time to the user; if the time difference is larger than T_{future}, which means the predicted time is too far away in the future, then the system will return the historical delay information calculated by the clustering model described in Section IV-A to the user. Prediction for a future day always uses the model generated by the clustering analysis.

VII. Conclusion and Future Work

This paper investigates the integration of real-time and predictive analytics in a smart decision support system. To

evaluate the proposed model, we use real-world historical data of two routes of Nashville's bus system. The results show that our real-time prediction model outperformed MTA's current basic model and using a 2-hour time window produces the best trade-off between overhead and performance. Our future work will be focused in two directions: integrating more data characteristics and adaptive system deployment.

A. Data Features

In the current Transit-Hub system, we are using data that is directly related to the transit system, such as static and real-time transit feeds, historical time point datasets, etc. One possible improvement in the future would be integrating more feature vectors into the analysis and prediction models. We have begun to collect data from other sources that can potentially affect bus arrival time. The features we plan to integrate are traffic flow, weather conditions and special events in the city. Analyzing these additional data sources would be helpful to answer the questions like which feature plays the most important role in causing bus delays and how to predict delay in the future when there is no real-time transit data available. Furthermore, the work can be expanded to other cities to compare the differences in delay factors across cities.

B. System Deployment

Currently the analytics and prediction models are all deployed on a private cloud, the performance may decrease when the number of users and computation scale increase in the future. In the next step, different modules in the system can be deployed in different cluster groups in the cloud according to the module's time requirement, data scale and computation latency. For example, the long-term transit analytics engine has the largest latency but its tasks are not time sensitive, therefore it can be deployed on a public cloud that is far away from the users. Real-time delay prediction engine has smaller latency but its response is expected to be real-time, so it can be deployed in the cloudlet which is closer to the users.

ACKNOWLEDGMENTS

This work is sponsored by National Science Foundation under the award number CNS-1528799. We acknowledge the support and suggestions provided by our partners from Nashville Metropolitan Transport Authority.

REFERENCES

[1] P. Poudenx, "The effect of transportation policies on energy consumption and greenhouse gas emission from urban passenger transportation," *Transportation Research Part A: Policy and Practice*, vol. 42, no. 6, pp. 901–909, 2008.

[2] "Top reasons people stop using public transit," http://www.governing.com/blogs/view/gov-reasons-riders-abandon-public-transit.html.

[3] M. Elhenawy, H. Chen, and H. A. Rakha, "Dynamic travel time prediction using data clustering and genetic programming," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 82–98, 2014.

[4] J. Patnaik, S. Chien, and A. Bladikas, "Estimation of bus arrival times using apc data," *Journal of public transportation*, vol. 7, no. 1, p. 1, 2004.

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

[6] R. America, "Midsize cities on the move: A look at the next generation of rapid bus, bus rapid transit, and streetcar projects in the united states," 2012.

[7] "2010 united states census," http://2010.census.gov/2010census/.

[8] 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

[9] M. D. Abkowitz and I. Engelstein, "Factors affecting running time on transit routes," *Transportation Research Part A: General*, vol. 17, no. 2, pp. 107–113, 1983.

[10] T. Kimpel, "Time point-level analysis of transit service reliability and passenger demand, urban studies and planning," *Portland, OR: Portland State University*, p. 154, 2001.

[11] J. Surprenant-Legault and A. El-Geneidy, "Introduction of reserved bus lane: Impact on bus running time and on-time performance," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2218, pp. 10–18, 2011.

[12] B. P. Sterman and J. L. Schofer, "Factors affecting reliability of urban bus services," *Journal of Transportation Engineering*, vol. 102, no. ASCE# 11930, 1976.

[13] R. Camus, G. Longo, and C. Macorini, "Estimation of transit reliability level-of-service based on automatic vehicle location data," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1927, pp. 277–286, 2005.

[14] M. Saberi, K. Ali Zockaie, W. Feng, and A. El-Geneidy, "Definition and properties of alternative bus service reliability measures at the stop level," *Journal of Public Transportation*, vol. 16, no. 1, 2013.

[15] J. Lin, P. Wang, and D. T. Barnum, "A quality control framework for bus schedule reliability," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 6, pp. 1086–1098, 2008.

[16] S. Gilmore, M. Tribastone, and A. Vandin, "An analysis pathway for the quantitative evaluation of public transport systems," in *Integrated Formal Methods*. Springer, 2014, pp. 71–86.

[17] J. Bates, J. Polak, P. Jones, and A. Cook, "The valuation of reliability for personal travel," *Transportation Research Part E: Logistics and Transportation Review*, vol. 37, no. 2, pp. 191–229, 2001.

[18] A. Abdelfattah and A. Khan, "Models for predicting bus delays," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1623, pp. 8–15, 1998.

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

[20] R. Jeong and L. R. Rilett, "Bus arrival time prediction using artificial neural network model," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004, pp. 988–993.

[21] M. Chen, X. Liu, J. Xia, and S. I. Chien, "A dynamic bus-arrival time prediction model based on apc data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, no. 5, pp. 364–376, 2004.

[22] R. H. Jeong, "The prediction of bus arrival time using automatic vehicle location systems data," Ph.D. dissertation, Texas A&M University, 2005.

[23] B. Yu, W. H. Lam, and M. L. Tam, "Bus arrival time prediction at bus stop with multiple routes," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1157–1170, 2011.

[24] Y. Bin, Y. Zhongzhen, and Y. Baozhen, "Bus arrival time prediction using support vector machines," *Journal of Intelligent Transportation Systems*, vol. 10, no. 4, pp. 151–158, 2006.

[25] C. Bai, Z.-R. Peng, Q.-C. Lu, and J. Sun, "Dynamic bus travel time prediction models on road with multiple bus routes," *Computational intelligence and neuroscience*, vol. 2015, p. 63, 2015.

[26] L. Vanajakshi, S. C. Subramanian, and R. Sivanandan, "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses," *IET intelligent transport systems*, vol. 3, no. 1, pp. 1–9, 2009.

[27] S. P. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[28] 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.

[29] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.