Real-time Forecasting of On-Time Performance in Metrorail Systems

Robert "Randy" Soper*, Bryse Flowers‡, Sandeep Kumar Bijinemula‡, Brian Mayer*, Farnaz Khaghani†, Jordan Holt§, and Naren Ramakrishnan*
* Discovery Analytics Center, Department of Computer Science, Virginia Tech, Arlington, VA 22203
‡Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060
†Department of Civil Engineering, Virginia Tech, Blacksburg, VA 24060
§Washington Metropolitan Area Transit Authority, Washington, DC 20001
rsoper@vt.edu,brysef@vt.edu,bsk1410@vt.edu,mayer2@vt.edu,farnazk@vt.edu,JHHolt@wmata.com,naren@vt.edu

ABSTRACT

The Washington DC Metrorail system operates 1,126 railcars on six different routes over 117 miles of track to support over 500,000 passengers each day. The Washington Metropolitan Area Transit Authority relies on their On-Time Performance (OTP) metric to determine how well their system is running and identify delays in the system. This paper utilizes passenger tap-in/tap-out and train movement data to create a predictive model of OTP for current passengers in real time. These predictions can be used by WMATA to improve performance and communicate delays to passengers more effectively. Our approach goes beyond predicting OTP of current in-flight passengers and uses RNN predictions of the future network state to make OTP predictions for passengers *who have not yet entered the network*. These empirical applications can be powerful for agencies and planners to assess and improve transit service performance using big data analytics and real-time predictions.

1 INTRODUCTION

The Washington Metropolitan Area Transit Authority (WMATA) operates the Metrorail system in Washington D.C serving over half a million trips daily. The DC Metrorail system consists of 91 stations spanning 117 miles of track across 2 states and the District of Columbia (Fig. 1). It is the third-busiest rapid transit system in the United States in number of passenger trips. To support the development of new analytics for transit planning and operations, Virginia Tech's Discovery Analytics Center and WMATA formed a partnership through the US National Science Foundation(NSF)-sponsored Urban Computing Program. This paper represents one of the first problems addressed through this collaboration.

One of the primary performance measures of interest to WMATA is On-Time Performance (OTP), defined as the percentage of passengers whose trip time is less than or equal to WMATA's Travel Time Standards (Figs. 2 and 3). WMATA has compiled a database of expected trip times for riders, given Origin/Destination pairs for each Service Period. For example, during the AM peak time, it should take no longer than 15 minutes to travel from Ballston to West Falls Church. Customers who experience a trip longer than this time are said to be late and decrease OTP for the network.

Inherently, one cannot know for certain whether a customer took longer than expected until the customer exits the network. Therefore, we propose a method for predicting the percentage of in-flight passengers who will experience a late trip using information which is available in real time such as where customers are *in-flight from* and the current train movement data. This is the first step

Figure 1: Map of DC Metro Rail System

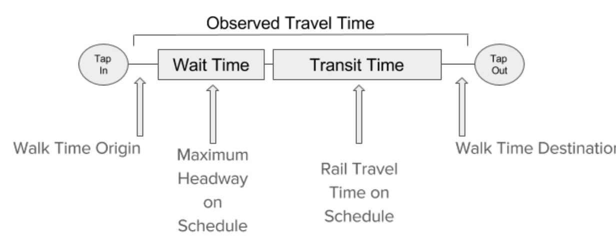

Figure 2: Components of Travel Time Standards

Origin	Destination	Service Period	Transfer	Train Travel Time	Wait Time Leg 1	Wait Time Leg 2	Walk Time Origin	Walk Time Destination	Walk Time Transfer	Travel Time Std
Ballston	West Falls Church	AM	N	7	6	0	1	1	0	15
Ballston	West Falls Church	AM_Peak	N	7	6	0	1	1	0	15
Ballston	West Falls Church	Midday	N	7	12	0	1	1	0	21
Ballston	West Falls Church	PM_Peak	N	7	6	0	1	1	0	15
Ballston	West Falls Church	PM	N	7	12	0	1	1	0	21
Ballston	West Falls Church	LateNight	N	7	20	0	1	1	0	29

Figure 3: Example Travel Time Standards

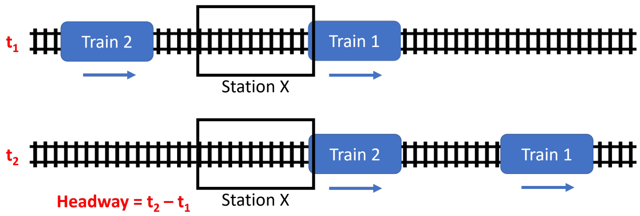

Figure 4: Defining Headway (HDW).

towards WMATA's primary goal of creating a predictive model that will identify how passengers will be impacted due to an incident.

The above proposed model gives an estimate of the number of passengers experiencing delays in real time using passenger and train movement data. However, knowing that the network is currently in a delayed state is not actionable. We, therefore, propose a method for predicting headway (HDW; defined as the time from the instance the tail of a train leaves the station and the instance the tail of the next train leaves the station, see Fig. 4) delays for every station on the network given the prior HDW delays using an RNN (recurrent neural network).

1.1 Research Impact

Having a predictive model for OTP allows WMATA to better estimate where delays will occur. WMATA currently makes real-time system control decisions based on the dispatcher's assessment of the network state and some alerts that are set for a certain thresholds. This process is almost entirely manual and lacks predictive modeling or recommendation algorithms. Our model adds accuracy to their expectations as well as detail (station by station). WMATA operators can then take actions to mitigate these effects in advance of the riders being impacted. These actions may include physical changes to the network by adding more trains/cars or staggering/slowing trains to close gaps in service from an incident or delay. WMATA will also communicate this real-time delay information to their customers. Giving riders this information allows the riders to adjust their trip accordingly or take another mode of transportation (which also helps to resolve overcrowding). Additionally, customers tend to be happier when you are able to give them better information.

Additionally, from a historical perspective understanding the impacts of certain events or incidents enables WMATA to better invest resources or make planning changes that can improve the performance of the network.

Furthermore, WMATA recently implemented the *Rush Hour Promise* policy for 2018 that refunds fares to customers who experience 15 minutes or more of delay during peak travel times (AM, AM Peak, and PM Peak) [24]. Being able to react to potential delays sooner can now be tied to a direct monetary impact.

This paper makes the following key contributions:

(1) We propose a discrete state model encoded in minute-by-minute intervals that uses either passenger or train movement data to represent and reason about the status of the system. This approach enables us to adopt a categorical encoding of normalized HDW state for describing and forecasting the overall state of the Metrorail network.
(2) Building upon the network state encoding approach, we propose a time shift model, based on the expected travel time of a train between stations, to better understand and represent delay propagation. This propagation method further enriches the vocabulary that is available for forecasting via machine learning algorithms.
(3) WMATA's analysis is currently limited to retrospective studies of OTP. Our approach develops two new predictive models that enables operators to forecast the experience of riders currently in network and that of riders *who have not yet entered the system.*

2 RELATED WORK

The potential of smart card and detailed train movement patterns for planning and performance management have attracted the attention of many researchers in urban computing and public transit system analysis[12, 18, 26]. Massive data being collected by smart card and railway information systems can greatly benefit traffic management and dispatching. We focus our related work discussion primarily on works that concentrate on modeling and predicting delays.

Most studies [3, 10, 11, 15] attempt to predict actual arrival, travel, or delay times and report errors based on actual times. WMATA has stated their current interest is only in identifying OTP. This relaxation allows this paper to focus simply on classifying trains as on-time or not on-time instead of attempting to predict a more precise transit time. Strathman [21] did perform an empirical assessment of factors simply affecting OTP but did not make any efforts to predict OTP.

A key problem when predicting train delays in large-scale railway networks is often studied through the lens of detecting recurrent delays. Several studies have treated this problem as one of time series forecasting using train movement records as the fundamental data source. Milinkovic et al. [15] employed a fuzzy petri net (FPN) model to estimate train delays. An alternative approach is to address delay prediction from the perspective of stochastic modeling where the duration of each activity has an associated probability distribution. Two of the most important studies using this approach, Carey et al. [2] and Meester et al. [14], proposed using approximations of delay distributions to reduce computational effort and study the error propagation for such approximations. Berger et al. [1] forecasted the delay for the whole German railway network based on directed acyclic graphs, the timetable, and its corresponding event graph.

Multiple studies [3, 11] acknowledge the advantages to using advanced techniques, such as neural networks, to model transit travel times because times are affected by various inter-correlated and time-varied factors and neural networks do not require a specific form or function. Neural networks can capture complex relationships between the dependent variables. Pongnumkul et al. [20] modeled train delays in the network using kNN models. Using neural networks, Yaghini et al. [25] and Peters et al. [19], investigated the problem of predicting train delays based on the delay profiles of the trains on each line. These sets of data-driven models make it possible to use regression analysis and classification techniques to predict future delays. Kalaputapu [8] used ANNs to predict schedule deviation of bus routes.

Key	Value
Entry DTM	01-MAY-15 00:00:25
Entry MSTN ID	MSTN 010
Exit DTM	01-MAY-15 00:45:52
Exit MSTN ID	MSTN 056
Trip Time Min	45.45

Figure 5: Sample record of tap-in tap-out passenger data.

In recent efforts, which mainly focuses on addressing the issue of long training times of neural networks, Oneto et al. [17] develop a specialized algorithm for big data parallel architectures, to analyze past delay profiles and predict future ones accordingly.

Significant research has been performed on travel time prediction using regression as well as more advanced techniques, like neural networks, for vehicular traffic (including buses) on roadways [10, 11, 22]. These methods utilize loop data (flow, occupancy and/or velocity) to make their predictions. Similar data points could be collected in the train network but WMATA and most other transit agencies don't currently collect them. There is also a concern that even if collected they might not hold the same predictive power given the differences in the road network versus a train network.

3 AVAILABLE DATASETS

3.1 Passenger Data

WMATA provided the team with tap-in/tap-out passenger data (collected from the turnstiles gating entry to and exit from the stations of the Metrorail network) and Travel Time Standards for estimating OTP trip times. The passenger data comprises rows of passenger trips where each row contains columns describing the entry and exit stations/times and a column that subtracted those times to provide a total observed trip time for that passenger in minutes. The data spanned all of 2015 and 2016; however, we analyzed only a four-month subset (May, August, September and October of 2015) of this data to avoid major long-term maintenance efforts as well as holidays while still encompassing any potential seasonality. An example of a single row of data can be seen in Fig. 5.

3.2 Train Movement

WMATA tracks the movement of trains with a real-time system which includes the current status of all traffic in the network (i.e., including in-transit). For this paper, the research team had access to data about train interactions with station platforms. The data spanned from 4 January 2015 to 31 December 2016 and included 34,624,940 rows each accounting for a single arrival/departure of a train at a network platform. A similar four-month subset (March, May, August, and October of 2015) of train movement data was selected again to avoid major long-term maintenance efforts as well as holidays while still encompassing any potential seasonality.

The train movement data associates platforms to metro stations in a many-to-one relationship (186 platforms to 91 stations). WMATA's platform ID designation also reflects the line and direction of the train. Note that some physical platforms are shared by trains of different lines moving in the same direction over portions where the networks converge but in the data each line has separate stations and platforms even if they are physically shared.

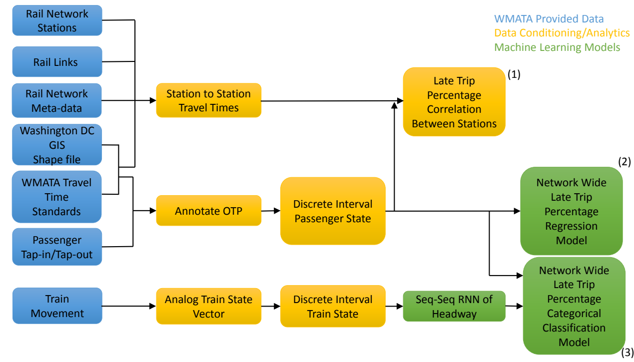

Figure 6: Flowchart for creation of passenger state, train state and percentage correlation between stations.

Arrival/departure data includes arrival time (measured by the arrival of the head of the train at the station), departure time (measured by the departure of the tail), and the number of cars in the train (referred to as length, LN). The dwell (DW) of the train is also available in some cases. WMATA defines dwell as the period of time that the doors of the train are open. Apparently, issues with door sensors prevent consistent measurement of this value.

3.3 Supplementary Datasets

In addition to the above mentioned datasets, other supplementary datasets were provided by WMATA. These are represented in Fig. 6.

4 DATA CONDITIONING

One approach to associating the train movement data (platform arrival/departure events) and passenger data (entry/exit at stations) to the overall network is to establish a vector related to the network state. As we have more information about network nodes (stations) than network edges (rail line), we chose to model the network from a node perspective. Creation of state for train and passenger data is defined in subsections below.

4.1 Passenger Data

A flowchart depicting the workflow for creating passenger state is shown in Fig. 6. The passenger data was pre-processed and an extra column depicting the *ServicePeriod* was added. The data was then annotated with the appropriate *OTP* standard. Late trips were normalized by dividing the number of late trips by the total number of trips. We then create a state vector for each minute in the time period studied. Each minute will capture how many users are on the network (*In-flight*), where they are coming from (*In-flightFrom-MSTN-ddd*), where they are heading to (*In-flightTo-MSTN-ddd*), and how many of these users will experience a late trip (*Late Trip Percentage*).

The total network usage for May 1st, 2015 is shown in Fig. 7. As expected, there are peaks in users on the network during the AM Peak and PM Peak periods. In between, the network stabilizes around 500 entries/exits per minute and thus the number of passengers in-flight remains constant.

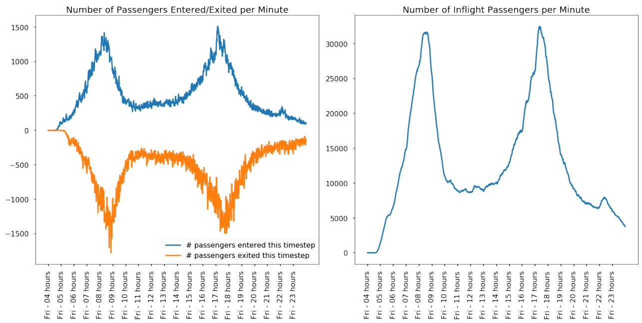

Figure 7: Left: Plot of entries and exits per minute on May 1st, 2015. Right: Plot of in-flight passengers on network per minute on May 1st, 2015. In-flight is defined as passengers which have entered the subway system but have not yet exited during that minute.

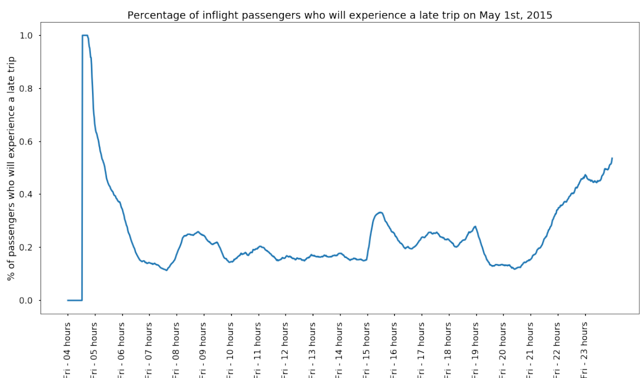

Figure 8: Percentage of passengers/minute currently in-flight who will experience a late trip on May 1, 2015.

The percentage of in-flight passengers who will experience a late trip is shown in Fig. 8. Spikes in late trip percentages at the beginning and end of service seem to be characteristic of this data. It is believed that passengers enter the station/network before the first train. Therefore, they end up waiting longer than the Travel Time Standards predict for the first train and experience a late trip. WMATA also confirmed that maintenance is typically scheduled at these times of the day. The fact that there is a lower number of in-flight passengers during these periods results in higher noise and also requires more time for in-flight OTP to return to “normal.”

4.2 Train Movement

The team used four variables to identify the state of the train movement network: HDW, DW, LN and Transit Time (TT, which is calculated as a trains arrival at station 2 minus its departure from station 1). The values of these variables are determined by recorded events that change the continuous state of the overall network. The team had to make a design decision regarding how to associate these discrete events and the associated state data to the timeline of the network. For example, in the case of HDW, a change in the HDW could be written to the network state when Train 1, from Fig. 4, departs Station X (customer-focused), when Train 2 departs Station X (sensor-focused), or anytime in between. We chose to take a network-focused approach. Changes to the network state occur when changes initially occur in the network, but when in reality the final value of the change may not be fully realized in the system. This choice was made in order to simplify network state creation and to avoid potential cause-effect sequence inconsistencies that might be introduced by over points of view (like customer-focused).

Although the analog state vectors describe the network, they do not reflect any normalization to expected network conditions, like rush-hour versus weekend schedules and relative scale of transit delays versus normal transit time. WMATA has specific day-of-week, time-of-day, Metro line, and platform-specific normalization values appropriate to the state variables. We normalize the resultant state vector using the Travel Time Standards data. For the purposes of this research, we thereafter seek a binary encoding of the state that captures the general characteristics of the information. The encoding of the normalized network state might then be used as a feature set for regression, classification, or other analysis.

Using our 4 month subset of data we constructed 113,464 minute-by-minute values of the four states for the 171 network platforms (684-dimension analog state vector). Each day’s data was bounded by the Metro service hours and each morning the state vector was initially undefined. Upon daily state vector initialization, a few platforms retain undefined status much longer than the rest of the network, reflecting little-used platforms. While these platforms could be discarded from the state vector entirely, their use might indicate the kinds of significant event that we wish to track and understand. On further analysis, we found that the cost to the volume of available data was, in the trade-off, worth including additional encoding to designate a "NaN" state at all platforms. Therefore, we chose to encode each state with three bits retaining the zero value to indicate "no defined value" associated with daily start-up and rarely used stations. The three bits enabled us to encode three ranges for each state.

In encoding the normalized state variables, it is important to note their long-tail distribution (according to the scale-free property of power laws, Fig. 9). WMATA is is concerned with staying on, or ahead of, schedule, as well as in degraded network performance that may be impacting customer experience. Therefore, for a three bit encoding, we decided to use one bit to designate on, or ahead of, schedule, leaving us two bits to separate the remainder of the range. In the absence of other data to appropriately segregate the remainder of the data, the median normalized value above zero would divide the remainder of the data into two equal volumes. Encoding of HDW is shown in Fig. 10. The normalized HDW, DW, and TT states are more directly and dynamically related to network incidents; due to WMATA’s data capture system, of these three states the HDW data is of the highest quality. For the purposes of the current effort, we have restricted analysis and classification efforts to the HDW component of the network state hereafter.

5 RESEARCH METHODOLOGY

Here is a list of the research questions addressed by the team:

(1) Can delays be correlated in time and space? (Section 6) (Box 1 in Fig. 6)
(2) Can passenger information, which is available in real time, be used as an indicator of the delays that customers are

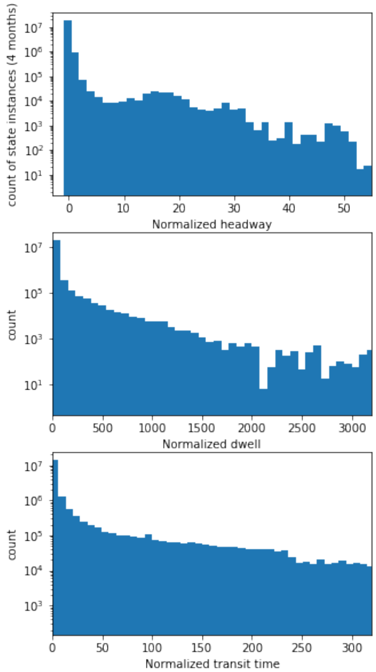

Figure 9: Histograms of Normalized Network State Variables

	Normalized HDW
[0,0,0]	NaN
[1,0,0]	≤ 0
[0,1,0]	> 0 and ≤ 0.4
[0,0,1]	> 0.4

Figure 10: Binary Encoding of HDW Network State

currently experiencing on the network? (Section 7) (Box 2 in Fig. 6)

(3) Can the current state of the network be used to forecast delays that will be seen by riders who have not yet entered the network? (Section 8) (Box 3 in Fig. 6)

6 CORRELATING LATE TRIP PERCENTAGES ACROSS STATIONS

Because passengers at different stations rely on the same train and rail network to transport them to their destination, it is expected that when large percentages of passengers at one station experience delay, that large percentage of passengers at neighboring stations will also experience delay. Therefore, we analyze in-flight OTP To and From derivations (described in Fig. 11) as continuous signals and compute the Pearson product-moment correlation coefficient for all station pairs.

Our first cut, as shown at the top of Fig. 11 simply correlates the continuous signals. We first drop out any portions of the signals where there are no passengers on the network (overnight when the network is closed) and then take the Pearson product-moment correlation of all in-flight OTP for all red line stations. We note that communities appear where stations that are closer together on the red line are more highly correlated to each other. We noticed that Judiciary Square does not appear to be highly correlated with any other stations. The reasoning for this seems to be that construction work was being done at the entrance to the station that impacted the usage of the station as compared to other stations. Additionally, we note that the end of line stations are still slightly correlated with one another (indicated by the bottom left and top right corners).

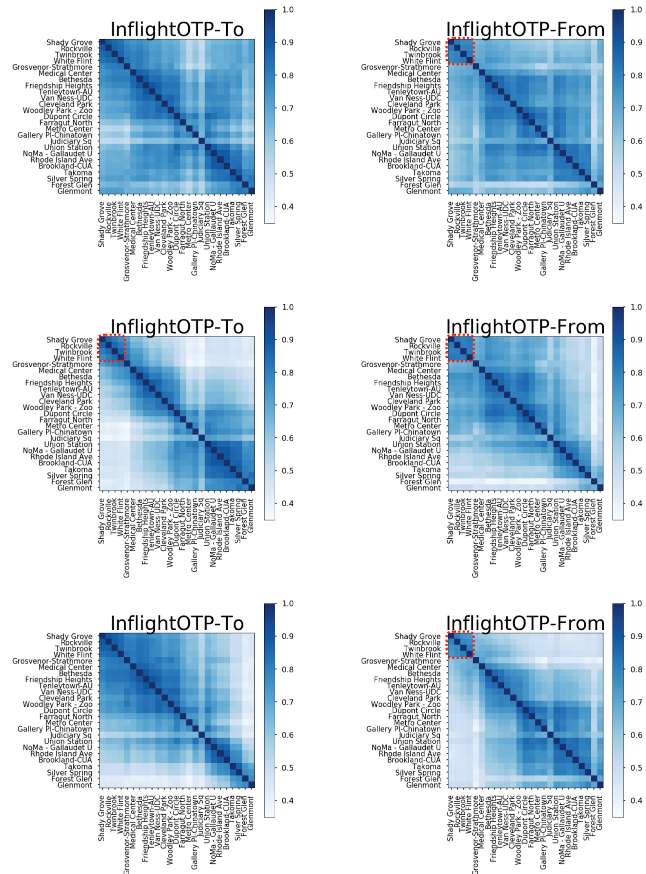

Figure 11: Pearson product-moment correlation coefficients between Red line stations' in-flight OTP at varying time shifts (Top: No Shift, Middle: Left Shift, Bottom: Right Shift). Stations are ordered on the correlation plot as they are ordered on the rail line. The left plots represent correlations between OTP calculated only from passengers which were currently in-flight and would exit the network at that station, the right plots represent correlations between OTP calculated only from passengers which were currently in-flight and had entered the network at that station. The Shady Grove, Rockville, Twinbrook, and White Flint community is highlighted in red.

We then apply time shifts to the data to reflect our belief that a delay at a station will not affect another station until at least the time it would take a train to traverse between the two stations (TT). For example, if Station 2 is 4 minutes away from Station 1 then delays at Station 1 during minute X are compared to delays at Station 2 during minute $X + 4$ when a *Left Shift* is applied. X would

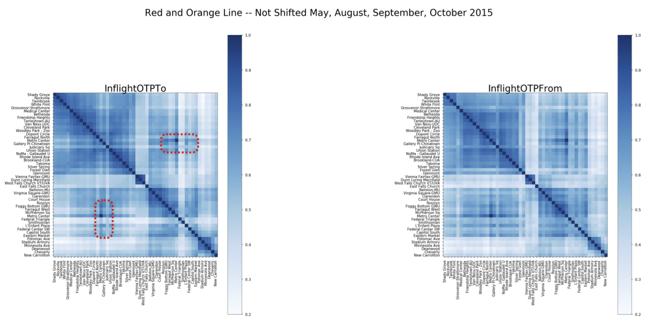

Figure 12: Pearson product-moment correlation coefficients between Red and Orange line stations' in-flight OTP with no time shift. Stations are ordered as they are ordered on the rail line. The left plots represent correlations between OTP calculated only from passengers which were currently in-flight and would exit the network at that station, the right plots represent correlations between OTP calculated only from passengers which were currently in-flight and had entered the network at that station. Metro Center's community is highlighted in red. Note the difference in color scale between this and Fig. 11.

be compared to *X - 4* for a *Right Shift*. Fig. 11 shows a depiction of the a left shift (middle) and right shift (bottom). After applying the time shifts, the communities become even more apparent. We see that the first four stations (Shady Grove, Rockville, Twinbrook, and White Flint: highlighted in Fig. 11) are all highly correlated in three of the four shifted plots (and only one of the plots without a shift) and also appear to be less correlated with the other stations on the rest of the line. The correlations between the end of line stations are no longer there, indicating that these stations may not have direct affects on each other via trains or the rail line but are correlated due to similarities in volume of customers or cyclic properties of late trip percentages.

We sought to expand the dataset to include two lines, orange and red. It is less clear how a time shift would be applied to these signals, therefore, we only studied the no shift case, which can be seen in Fig. 12. The ends of the lines are almost completely uncorrelated. It is much less apparent but we do see some correlation near the center of the two lines where they share a transfer station at Metro Center (highlighted in red in Fig. 12).

In order to better visualize the communities that are revealed by the correlation plot between stations, we employ k-spectral clustering [16]. Interestingly, spectral clustering generally groups the stations closest to the transfer station (on both lines) together when using the InflightOTPTo correlation matrix as a precomputed affinity matrix, while the end of lines form separate clusters for each line. One result, when using 3 clusters, is shown in Fig. 13. Note that the clustering results using the InflightOTPFrom correlation matrix does not show a similar result.

WMATA sees the benefit in identifying clusters and understanding propagation and correlation across stations because it allows them to communicate more specific and effective delay notifications. Instead of issuing network-wide or only rail-line specific

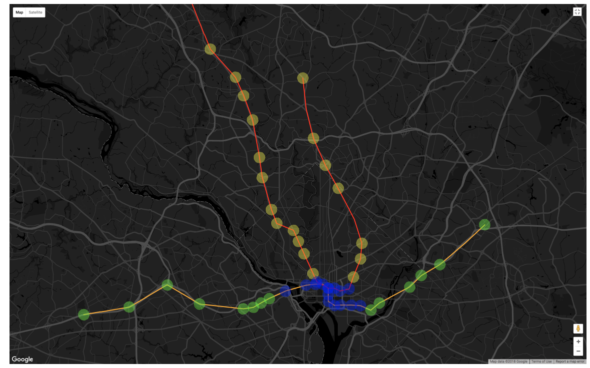

Figure 13: Google Maps depiction of the results of 3-spectral clustering using the *InflightOTPTo* correlation matrix as a precomputed affinity matrix.

alerts they could provide alerts specific to a cluster of stations. Riders can then potentially choose to make decisions that avoid these stations instead of opting to avoid the entire rail-line or metro as a whole.

7 RIDER EXPERIENCE PREDICTION USING REAL TIME PASSENGER DATA

Initially, to predict the delay of passengers currently on the network given only information that would be available in real time, the passenger state vector was used, along with minute, hour, and day-of-week. In order to conform to the real-time data availability requirement the *In-flightTo-MSTN-ddd* features were dropped. The *Late Trip Percentage* attribute was used as the values to learn with the remaining attributes as features (normalized). We employed a fully connected neural network (with dense layers of size 512, 128, and 32) and a ReLU activation function with a dropout rate of 50%. The output of these layers was fed into a single output with a sigmoid activation function for use in regression prediction of late trip percentage. The model was built and trained using the Keras [6] framework on top of TensorFlow [13] using the Adam optimizer [9] with *mean squared error* as the loss function.

The vectors were split into training and validation sets based on real-world availability. Therefore, instead of randomly splitting the rows, the data from May, August, and September of 2015 were used to train the model and the data from October 2015 was used to validate the model. The results from the model are shown in Fig. 14.

8 RIDER EXPERIENCE PREDICTION USING PRIOR RAIL NETWORK STATE

In an effort to build a predictive model of customer experience based on train movement, we construct a neural network architecture based on a series combination of deep learning primitives. This deep learning architecture allows us to predict future network state based on the highest quality subcomponent of measured data available, and then embed the state in a lower dimensional space. Researchers have demonstrated the effectiveness of conditioning a decoding recurrent neural network (RNN) with the final state of an encoded sequence when predicting a time series whose entirety depends on another sequence [7][4][23]. Here, we seek to use an

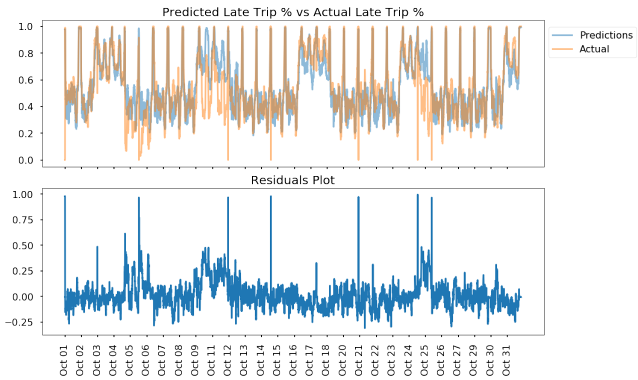

Figure 14: Forecasts for October 2015 (the model has a mean squared error of 0.0157 and an R^2 value of 0.691). Each prediction represents a minute in time.

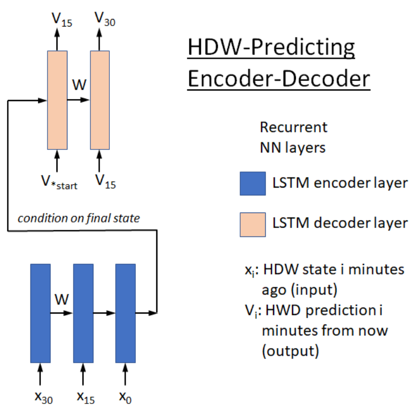

Figure 15: Seq2Seq RNN for HDW-State-Prediction

encoder-decoder sequence-to-sequence (seq2seq) RNN to predict sequences of future HDW states based on past sequences (Fig. 15).

We feed the output of the decoder RNN network into an autoencoder in order to develop clusters more closely related to the customer experience classification of interest. This predicted-embedded space becomes the basis for a classification algorithm that we show is able to predict binary and more granular class categorizations of network-wide OTP, 30 minutes in advance.

Design and training of the (seq2seq) RNN, autoencoder, and customer experience classifier could now each be performed in sequence. Restricting the scope to HDW-only states of the network will increase the number of available training sequences that can be held in memory while dovetailing with our results of customer-experience prediction from the auto-encoded state. We chose to keep the length of the sequences short and the scope of the future forecast limited to improve accuracy, predicting 15 and 30 minutes of future state based on the state 30 min and 15 min in the past and the current state. We used Chollet[5] as a guide for our model including a hidden layer of 256 LSTM cells (input/output dimension for HDW state of 513). The output layer of the decoder was set to sigmoid activation to create a binary encoding prediction. Training was via mini-batches of 64 input sequences using cross-entropy loss against the binary decoder output and stochastic gradient descent with Adam optimization. Independent validation of the HDW data prediction was via test data of length 13,004 sequences, providing 2,223,684 individual platform HDW state evaluation opportunities. For each three-bit platform output, the value was assigned to [0,0,0] if the maximum output was less than 0.33 and to the maximum argument bit otherwise. Accuracy of a 20% withheld test set during training of the seq2seq RNN prediction for the 15-min forecast was 0.962 and 0.958 for the 30-min forecast, albeit against a null-accuracy of 0.88 for each due to the relative stability of the network.

Next an autoencoder with fully connected layers of dimension 513-128-32-128-513 was trained on the 30-min forecast HDW state to create an embedding of reduced dimension of 32. The design and training of the autoencoder is typical (hidden layers with ReLU activation, output with sigmoid, cross-entropy loss, mini-batches of 128 inputs, stochastic gradient descent with Adam optimization [9], early stopping at 12 epochs). The hope in using an autoencoder to learn a nonlinear embedding for the state is that clusters of differentiated state character will also discriminate network-wide ridership experience.

The nature of the encoding as a vector space that may provide clustering associated with the desired classification suggests intuitively that k-nearest neighbor would be an appropriate classification technique to apply. To confirm our intuitive choice of kNN classification, we compared kNN to decision tree and naive Bayes classifiers and confirmed the superiority of the nearest neighbor classifier (7-13% improvement in accuracy for binary above/below mean categorization).

We compared performance of various numbers of neighbors as well as linear support vector classification (SVC) for a range of ground-truth binary threshold division values (Table 1); note that model selection in application might also be influenced by desirable operational capability (value of classification division point) as well results driven by the data. Here, we suggest 0.4 might represent a reasonable balance between predictive performance and ability to segregate normal-customer-experience mode from poor-customer-experience mode. Considering 38,588 values of late-customer percentage and corresponding HDW sequences/predictions from May, Aug, and Oct of 2015, late-customer-percentage characteristics are given in Fig. 16. Using a training dataset of 80% of values and a test data set of 7,718 values (20%) and pre-trained RNN state prediction and autoencoder networks previously described, we first considered prediction of in-flight late-customer-percentage performance based on categorization as above and below a sliding ground-truth threshold value for various numbers of kNN neighbors (Fig. 17).

Considering the character of the late-customer-percentage histogram, it seems likely that the classifier has difficulty differentiating standard modes of network operation leading to common customer experience in the lower, more bunched section of the late-customer fraction histogram, below 0.4 (Fig.16).

WMATA is interested in increasing prediction performance and granularity at lower values of customer late percentage to refine performance during standard operations to maximize efficiency. However, WMATA is more interested in the less frequent, but more impactful, longer delays in support of its "Rush Hour Promise" initiative. Therefore, sacrificing granularity at the lower range of late-ridership percentage in the interest of improving percentages at the higher range may be an appropriate tradeoff. Varying the

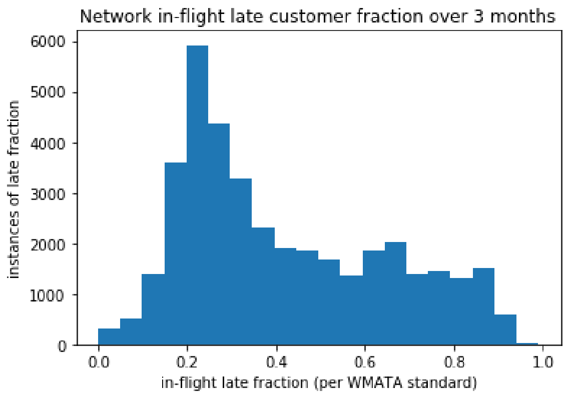

Figure 16: Histogram of In-Flight Percentage of Late Customers

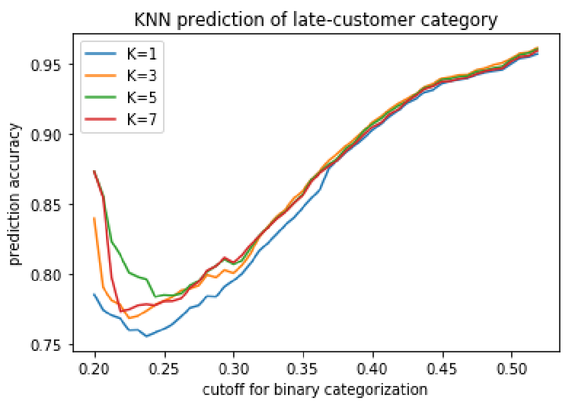

Figure 17: Binary Classifier Accuracy Versus Threshold Value

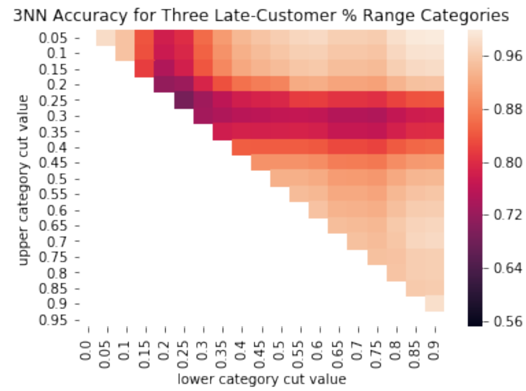

Figure 18: Accuracy Heat Map for Three Customer Experience Classes

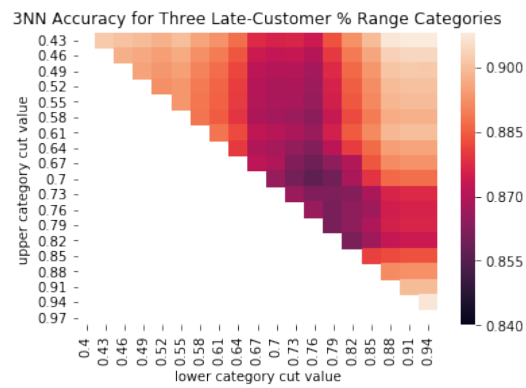

Figure 19: Accuracy Heat Map for Four Customer Experience Classes

categorization threshold for various numbers of categories in late-customer percentage confirms our suspicion that values in the lower quartile/near the mode of operation result in poor classifier performance. Fig. 17 shows variation in the accuracy for a single threshold value/binary classification. Adopting a three neighbor classifier with the lowest threshold value near the inflection point 0.4 to discriminate between normal operation modes below, and operation modes of concern above (see bold values in Table 1) additional categories may be introduced. Fig. 18 and 19 show heat maps of accuracy varying two threshold values for 3 neighbors. When the three threshold values are set to 0.4, 0.55, and 0.7 we are still able to achieve an overall classifier accuracy of 0.867.

9 CONCLUSION

Metrorail data, both passenger tap-in/tap-out and train movement, enable modeling of WMATA's OTP metric. Unfortunately, as is common in the transit industry, the data sets are large and cumbersome to work with. They show lack of diligence in fields with manual coding/entry, issues with sensor data, corruption in file export and handoff and other challenges that result in 'dirty data'. Some aspects of the data, particularly those naturally flowing from network operation and not under WMATA's direct control, exhibit long-tail power-law scale-free distributions. The network itself is geospatially distributed and the data has important temporal sequencing, but the nature of the geospatial network is not simple, consisting of separate, yet interconnected and overlapping networks.

This initial exploration created new bases for defining the state of the network, algorithms for extracting regularized information from the datasets, and demonstrated that regression, encoders, neural networks, and classifiers are capable of providing WMATA with predictive and passenger-associative knowledge that they have never had available to high accuracy.

10 FUTURE WORK AND NEXT STEPS TO DEPLOYMENT

Further research regarding delay propagation could investigate a more accurate depiction that either simultaneously applies a shift in both directions based on a passenger's direction of travel or investigates the correlations by direction. This should increase correlations because if a single train is delayed it should primarily affect only customers traveling in that direction. Certain delays like single-tracking (where trains traveling in both directions have to share a single section of track) and over-crowding can still propagate across both directions which is why we saw the results we did in our analysis.

WMATA is interested in utilizing the prediction models operationally in practice. Some remaining steps to deployment include

Table 1: Table Performance of Binary Classifiers Versus Threshold Value.

Threshold	Accuracy			Recall			Precision		
	3NN	5NN	SVC	3NN	5NN	SVC	3NN	5NN	SVC
0.200	0.8393	0.8730	0.8760	0.8915	0.9757	0.9855	0.9157	0.8853	0.8814
0.250	0.7804	0.7848	0.7932	0.7612	0.8350	0.9336	0.9037	0.8490	0.7990
0.300	0.8003	0.8066	0.8111	0.7384	0.7569	0.7618	0.9006	0.8945	0.8982
0.350	0.8593	0.8570	0.8496	0.7855	0.8017	0.7681	0.9175	0.8974	0.9132
0.400	**0.9081**	0.9068	0.8871	**0.8018**	0.8078	0.8111	**0.9831**	0.9726	0.9190
0.450	0.9396	0.9385	0.9118	0.8583	0.8627	0.8681	0.9827	0.9745	0.8993
0.500	0.9540	0.9531	0.9100	0.8807	0.8799	0.8900	0.9797	0.9776	0.8482

automating the data cleaning systems described here and integrating incident data. WMATA has records for about 10,000 incidents a year. These incidents range from personal breaks by a train conductor to entire lines being shutdown to regularly scheduled maintenance. Integrating maintenance records is crucial to ensure that the forecasting ability mentioned here dovetails with operational planning processes seamlessly.

The system as described in this paper could be applied to any rail network that collects Tap-In/Tap-Out data and/or Train Movement Data (departure time from stations, i.e., HDW). Additional datasets described in Section 3.3 and Fig. 6 are typically maintained by most transit authorities and could be created manually if not.

ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation via grants DGE-1545362, IIS-1633363, and by the Army Research Laboratory under grant W911NF-17-1-0021. The US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, Army Research Laboratory, or the U.S. Government. We also acknowledge the assistance of Nikhil Muralidhar, Sathappan Muthiah, and Mitch Wagner. WMATA employees also provided time, guidance, and suggestions for improving real-world applicability, as well as provided the requested supplemental data during the project.

REFERENCES

[1] Annabell Berger, Andreas Gebhardt, Matthias Müller-Hannemann, and Martin Ostrowski. 2011. Stochastic delay prediction in large train networks. In *OASIcs-OpenAccess Series in Informatics*, Vol. 20. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[2] M. Carey and A. Kwieciski. 1994. Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research Part B: Methodological* 28, 4 (1994), 251–267.

[3] Mei Chen, Xiaobo Liu, Jingxin Xia, and Steven I. Chien. 2004. A Dynamic Bus-Arrival Time Prediction Model Based on APC Data. *ComputerâĂŘAided Civil and Infrastructure Engineering* 19.5 (2004), 364–376.

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR* abs/1406.1078 (2014). arXiv:1406.1078 http://arxiv.org/abs/1406.1078

[5] François Chollet. 2017. A ten-minute introduction to sequence-to-sequence learning in Keras. *The Keras Blog* (online). (2017). https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html.

[6] François Chollet et al. 2015. Keras. (2015). https://github.com/fchollet/keras

[7] Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR* abs/1308.0850 (2013). arXiv:1308.0850 http://arxiv.org/abs/1308.0850

[8] Ravi Kalaputapu and Michael J Demetsky. 1995. Modeling Schedule Deviations of Buses Using Automatic Vehicle-Location Data and Artificial Neural Networks. *Transportation Research Record* 1497 (1995), 44–52.

[9] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

[10] Jaimyoung Kwon, Benjamin Coifman, and Peter Bickel. 2000. Day-to-day travel-time trends and travel-time prediction from loop-detector data. *Transportation Research Record: Journal of the Transportation Research Board* 1717 (2000), 120–129.

[11] Steven I-Jy Kwon, Yuqing Ding, and Chienhung Wei. 2002. Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering* 128.5 (2002), 429–438.

[12] Xiaolei Ma, Yao-Jan Wu, Yinhai Wang, Feng Chen, and Jianfeng Liu. 2013. Mining smart card data for transit ridersâĂŹ travel patterns. *Transportation Research Part C: Emerging Technologies* 36 (2013), 1–12.

[13] A. Martin et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). https://www.tensorflow.org/ Software available from tensorflow.org.

[14] Ludolf E Meester and Sander Muns. 2007. Stochastic delay propagation in railway networks and phase-type distributions. *Transportation Research Part B: Methodological* 41, 2 (2007), 218–230.

[15] Sanjin Milinković, Milan Marković, Slavko Vesković, Miloš Ivić, and Norbert Pavlović. 2013. A fuzzy Petri net model to estimate train delays. *Simulation Modelling Practice and Theory* 33 (2013), 144–157.

[16] Andrew Y Ng, Michael I Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*. 849–856.

[17] Luca Oneto, Emanuele Fumeo, Giorgio Clerico, Renzo Canepa, Federico Papa, Carlo Dambra, Nadia Mazzino, and Davide Anguita. 2016. Delay Prediction System for Large-Scale Railway Networks Based on Big Data Analytics. In *INNS Conference on Big Data*. Springer, 139–150.

[18] Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. 2011. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies* 19, 4 (2011), 557–568.

[19] Jan Peters, Bastian Emig, Marten Jung, and Stefan Schmidt. 2005. Prediction of delays in public transportation using neural networks. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, Vol. 2. IEEE, 92–97.

[20] Suporn Pongnumkul, Thanakij Pechprasarn, Narin Kunaseth, and Kornchawal Chaipah. 2014. Improving arrival time prediction of Thailand's passenger trains using historical travel times. In *Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference on*. IEEE, 307–312.

[21] Jame G. Strathman and Janet R. Hopper. 1993. Empirical Analysis of Bus Transit On-Time Performance. *Transportation Research Part A* 27-2 (1993), 93–100.

[22] Hongyu Sun, Henry X. Liu, Heng Xiao, Rachel R. He, and Bin Ran. 2003. Short term traffic forecasting using the local linear regression model. *82nd Annual Meeting of the Transportation Research Board, Washington, DC* (2003).

[23] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[24] WMATA. 2018. *Rush Hour Promise*. (2018). https://www.wmata.com/fares/smartrip/rush-hour-promise.cfm.

[25] Masoud Yaghini, Mohammad M Khoshraftar, and Masoud Seyedabadi. 2013. Railway passenger train delay prediction via neural network model. *Journal of advanced transportation* 47, 3 (2013), 355–368.

[26] Chen Zhong, Stefan Müller Arisona, Xianfeng Huang, Michael Batty, and Gerhard Schmitt. 2014. Detecting the dynamics of urban structure through spatial network analysis. *International Journal of Geographical Information Science* 28, 11 (2014), 2178–2199.