On Designing Day Ahead and Same Day Ridership Level Prediction Models for City-Scale Transit Networks Using Noisy APC Data

Jose Paolo Talusan[1], Ayan Mukhopadhyay[1], Dan Freudberg[2], and Abhishek Dubey[1]

[1]Vanderbilt University
[2]Nashville Metropolitan Transit Authority

***Abstract*—The ability to accurately predict public transit ridership demand benefits passengers and transit agencies. Agencies will be able to reallocate buses to handle under or over-utilized bus routes, improving resource utilization, and passengers will be able to adjust and plan their schedules to avoid overcrowded buses and maintain a certain level of comfort. However, accurately predicting occupancy is a non-trivial task. Various reasons such as heterogeneity, evolving ridership patterns, exogenous events like weather, and other stochastic variables, make the task much more challenging. With the progress of big data, transit authorities now have access to real-time passenger occupancy information for their vehicles. The amount of data generated is staggering. While there is no shortage in data, it must still be cleaned, processed, augmented, and merged before any useful information can be generated. In this paper, we propose the use and fusion of data from multiple sources, cleaned, processed, and merged together, for use in training machine learning models to predict transit ridership. We use data that spans a 2-year period (2020-2022) incorporating transit, weather, traffic, and calendar data. The resulting data, which equates to 17 million observations, is used to train separate models for the trip and stop level prediction. We evaluate our approach on real-world transit data provided by the public transit agency of Nashville, TN. We demonstrate that the trip level model based on Xgboost and the stop level model based on LSTM outperform the baseline statistical model across the entire transit service day.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Public transportation is a vital component in any modern metropolitan city. Access to reliable forms of public transit have been known to have an impact in many aspects, such improved quality of life, reduced carbon emissions, and have an overall positive effect on social equity. However, even with the availability of public transit, it is not always guaranteed that it is always reliable and accessible. On the contrary, they are more often over-stretched or underdeveloped. As a result, most of the work being done is focused on improving the accessibility and reliability of public transit.

Traditional measures of reliable transit systems include trip frequency, punctuality, and travel time. In response, plenty of work has been done with the goal of improving travel times by identifying and reducing causes of delay [1]. However, an often overlooked element in reliability is the perceived comfort of riders [2] which can be seen as the a direct consequence of vehicle occupancy and capacity. A frequently overcrowded bus can prevent potential commuters from even considering public transit. Inversely, consistently low rider demand can be seen as an under utilization of already constrained resources. This duality of public transit is often caused by the agencies' constant struggle with providing increased transit coverage amidst highly heterogeneous ridership demand.

With the progress of big data, transit authorities now have access to and are able to provide real-time passenger occupancy information for their vehicles. Transit agencies such as the Nashville Metropolitan Transit Authority (MTA) uses Automated Passenger Counter (APC) systems that provide stop-level estimates of passenger boarding and alighting. This information have been integrated by apps such as Transit[1], which in addition to allowing potential riders to see an estimated future passenger occupancy, also use crowdsourcing to collect occupancy information from riders onboard in an effort to improve service accuracy. From the perspective of passengers, this helps them choose departure times to match their desired comfort level. For the agencies, this can be a reference for them to optimize their services by allocating resources according to predicted ridership demand. Thus, accurately predicting the maximum occupancy of each vehicle in a public transit system is pivotal in improving perceived reliability, resource optimization, and rider comfort.

Achieving accurate occupancy prediction, however, is a difficult task. There are a number of factors that can affect demand ranging from short high impact factors such as sport events and festivals to long-term factors such as school schedule and season. Additionally, stochastic traffic conditions along the route can cause variation in ridership, further increasing uncertainty. Another issue that can affect prediction is sensor data noise. As with any system that relies on a fleet of sensors and a large database, there are bound to be inconsistencies and errors [3]. This is especially true for APC systems, where passenger boarding and alighting information are recorded using infrared sensors installed on vehicle doors [4]. This can lead to erratic and misleading information. This issue brings up the need for data preparation and augmentation to ensure that the data is reliable and useful.

In this paper, we implement an end-to-end framework for

[1]https://transitapp.com/

predicting occupancy at both the stop and route levels. This ensures that our method can react to both short and long-term changes in the public transit system. We do this by analyzing and combining different spatio-temporal data such as weather, traffic, and APC data to develop a model for bus occupancy. First, we investigate how data can be augmented and merged to provide features that would expose the relationship with bus occupancy. Second, we build different models for bus-stop and transit-route levels. Finally, we demonstrate and compare our approach using actual APC data from the public transit agency of Nashville, TN. One of the key parts of our setup is a data cleaning and augmentation method that processes and cleans raw APC data. Raw APC data is often noisy with a variety of different issues regarding the accuracy and precision of passenger counts [5]. Augmenting and cleaning ensure that data used in training models is valid. We generate passenger occupancy from alighting and boarding information.

Organization: The rest of this paper is organized as follows. In Section II, we give an overview of the state-of-the-art in occupancy prediction. In Section III, we present and formulate the problem. We then discuss in-depth the APC data in Section IV and the issues accompanying the dataset. In Section VI, we validate our proposed models using real-world data from Nashville, TN. Finally, in Section VII, we give our conclusions.

II. Related Work

In this section we discuss the current state-of-the-art methods used in public transit occupancy prediction.

A. Occupancy Prediction

Given the importance of public transit and the increasing ubiquity of available vehicle data, research in the field of occupancy prediction, also known as passenger flow or transit demand prediction, has been flourishing. There is a considerable number of work done on understanding and mapping the occupancy level in public transport.

Short-term passenger demand forecasting fall into one of two categories, parametric and non-parametric approaches. Traditionally, parametric approaches such as historical averaging [6] and autoregressive integrated moving average (ARIMA) [7] have been used to predict not only demand but traffic flow, travel times and vehicle speed. Ever since it was established, ARIMA has been known to perform well in modeling linear and stationary time series. However, ARIMA's shortcomings in taking into account seasonality and capturing non-linear relationships in data are also well known.

In contrast, non-parametric approaches build a non-linear relationship between the input and output variables without any prior knowledge. These methods gained popularity as consequence of the rapidly increasing availability of data from systems such as Advanced Public Transportation Systems (APTS) and Advanced Traveler Information Systems [4]. These techniques have been proven effective at forecasting demand based on data gathered through smart cards [8], [9]. Toque et al. [10] used Random Forest (RF) and LSTM neural networks trained on smart card data to predict travel demand. By creating multiple temporal units neural networks (MTUNN) and parallel ensemble neural networks (PENN), Tsai et al. [11] showed that it can outperform predictions based on statistical analysis of historical data. The obvious periodicity and repeatability of traffic flow data led to the development of various short-term and long-term prediction, with long-term prerdiction decreasing far slower than short-term. Wang et al. [12] uses an LSTM based Encoder-Decoder architecture to overcome the problems of gradient disappearance present in typical RNN models.

Incorporating other spatio-temporal dataset such as weather and special events have also been explored. Karnberger et al. [13] considered the effect of exogenous events on public transportation ridership. Meanwhile, Zhou et al. [14] combined smart data and weather information and found that while riders are more resilient to changes in weather, it still has an effect on the overall demand. Finally, Wood et al. generated models the passenger occupancy and demand at the next-stop/any-stop level based on APC and weather data [15] and proved that even simpler models such as RF and LSTM provide reliable estimates of future data when trained with historical information if demand patterns are fairly stable.

There has been plenty of work done in the field of public transportation with a special focus on improving reliability through understanding and forecasting passenger demand. However, our work is distinct in three ways. First, our work aims to provide occupancy prediction at both the stop and trip levels separately by forecasting short and long term demand. Second, we work on APC data which is fundamentally different from smart card data, which is the data commonly used by prior work. Smart cards are embedded with integrated circuits enabling it to process information, or in this case, allow for contactless ticketing for riding on mass transit. These cards are much more accurate and complete in their data collection [16], [17] due in part they require passengers to swipe after getting on and before getting of the vehicle. In contrast, APC data is much more noisy and introduces far more uncertainty in data collection and processing. Third, we focus on implementing this for the entire public transport system and not on a few select routes.

III. Problem Statement

Based on our conversations with the transit agency, they want to be able to identify particular trips and stops which experience overcrowding. Overcrowding increases the chances of passengers not being able to get on the bus and decreasing their overall satisfaction and willingness to take public transit again in the future. Knowing the maximum occupancy at the trip and stop level will allow them to react and prepare accordingly by increasing bus dispatch frequency thereby decreasing headway.

The primary objective of this work is to provide accurate occupancy prediction for public transit vehicles. The goal is to be able to reliable and efficiently forecast maximum ridership demand at both stop and trip levels. The problem then is,

given a fleet of heterogeneous vehicles[2], each equipped with automated passenger count systems, how are we able to model and accurately predict the maximum occupancy at any trip or stop in the future. We focus on using APC data in this paper since this is the current system being used by the transit agencies. While the use of smart cards would be better for generating models, it is not in the best interests of the transit agency to change their current system.

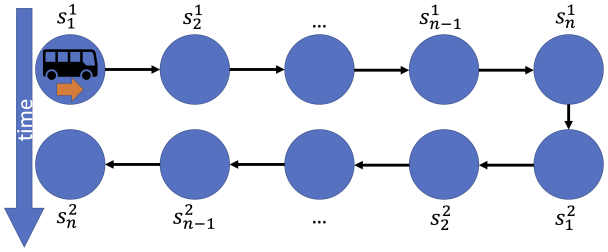

Fig. 1. A sample block assignment for a public transit vehicle

In a public transit timetable, each vehicle is assigned to serve a specific block. Each block is a collection of non-overlapping trips going back and forth a predetermined route. A single trip $t \in \mathcal{T}$ constitutes a vehicle travelling one direction across a single route, a round trip is made of two separate trips. In each trip, the bus passes by a predetermined number of stops $s \in \mathcal{S}$ where passengers can board from or alight to. At each stop a number of people get on or off the bus, this information is then recorded in the APC data as `ons` and `offs`, respectively.

We formally define a bus schedule as a collection of sequential trips $\{t_1, \ldots, t_r\}$ assigned to a bus, where each trip t_r is an ordered sequence of n stops $\{s_1^r, \ldots, s_n^r\}$. Fig. 1 shows two trips that have been assigned to a bus. The first row of stops s_1^1 to s_{1+n}^1 correspond to a trip t_1 with n stops. Once the vehicle reaches the end of this trip, it proceeds with its return trip, t_2, segment of the assigned route. Trip t_2 consists of stops from s_1^2 to s_{1+n}^2.

Our goal is to predict the passenger occupancy at the stop and trip levels. For the stop level, given a vehicle is at stop s_1^1, the goal is to predict occupancy at s_2^1. For the trip level, the goal is to predict maximum occupancy across the entire trip for any trip in the future, t_r.

IV. Data Collection and Processing

In this section we first provide an overview of the different data sources used and we describe the data augmentation and processing methods that we applied to it.

A. Data Sources

There are a variety of data from different sensors and sources that needs to be temporally and spatially joined together.

[2]In this work we use the terms vehicle and bus as public transit vehicles interchangeably.

- Automatic Passenger Counting (APC): Automatic Passenger Counting systems record a variety of information as the vehicle passes by bus stops. Sensors installed over vehicle doors are triggered when people exit and enter the bus, recording `offs` and `ons` respectively. Each entry in the APC is a log of the current state of the bus at a stop on a trip. This log also includes scheduled and actual stop arrival times.
- Weather: Weather data comes from multiple sources (Darksky and Weatherbit) and multiple weather stations. Data is matched based on the geographic locations of the stops and weather stations, and then joined with the APC data. Data includes precipitation, temperature and humidity.
- General Transit Feed Specification (GTFS): A dataset provided by the transit agency based on a common format for public transportation schedules and associated geographic information [18]. It includes all the schedules and time tables for all the vehicles in their fleet. It also includes the geometric routes and scheduled arrival times that can be used to compute scheduled headways and match with road traffic data. A version of the GTFS standard can be found here: https://developers.google.com/transit/gtfs/.
- Traffic: Traffic data is from INRIX [19]. It provides road segment level speed and congestion information in five minute granularities. Matching this data with APC is done by dividing the metropolitan city into one by one mile grids and identifying the grids where the bus trip's shape passes through. INRIX segments which are within these grids are then collected and the average traffic speed is obtained. This value is then joined with APC.
- Calendar: This includes information regarding city holidays and school breaks which have been shown to have an effect in the overall ridership demand [20].

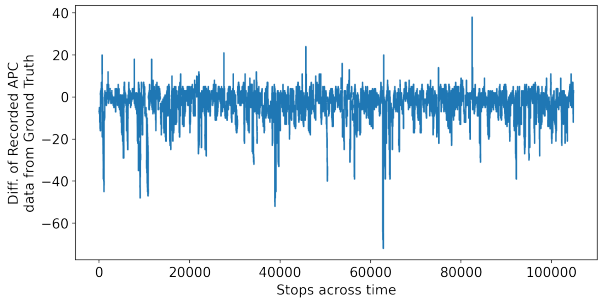

Fig. 2. Noise in occupancy sensor readings, difference between ground truth and sensor data observations over a period from 2021-03-21 to 2022-02-17.

B. Data Cleaning and Augmentation

APC data received directly from vehicles are noisy, often reporting highly erroneous data. Figure 2 shows a plot of the difference between the reported occupancy and ground truth data for a span of almost one year. The recorded data, on average, was 5 people away from the ground truth, with a

TABLE I
DATA FEATURES, SIZE AND SOURCES

Dataset	Range	Size	Rows	Features	Source	Frequency	Type	Description
Transit	01/01/2020 to 04/06/2022	831MB	17,000,000	Transit date	APC	variable	Temporal	Date of bus trip
				Route ID	APC	variable	Spatio-temporal	Unique route identifier
				Route direction name	APC	variable	Spatio-temporal	Name of route heading
				Scheduled headway	APC	variable	Spatio-temporal	Duration between buses headed in the same route and direction (per stop)
				Load	derived	variable	Spatio-temporal	Total occupancy at the stop (after alights and boards)
				Stop sequence	APC	variable	Spatio-temporal	Number of current stop within the entire trip
				Stop ID	APC	variable	Spatio-temporal	Unique stop identifier
				Past load	derived	variable	Spatio-temporal	Past loads from previous trips and stops
				Past actual headway	derived	variable	Spatio-temporal	Past actual headway from previous trips and stops
				Percent load change	derived	variable	Spatio-temporal	Percent change of occupancy from two stops or trips prior
				Percent headway change	derived	variable	Spatio-temporal	Percent change of headway from two stops or trips prior
				Zero load at trip end	APC	variable	Spatio-temporal	Boolean indicator if people should all alight at the end of the trip
Weather	01/01/2020 to 04/06/2022	300MB	226,105	Temperature	Darksky	1 hour	Spatio-temporal	Recorded temperature
				Humidity	Darksky	1 hour	Spatio-temporal	Recorded humidity
				Precipitation intensity	Darksky	1 hour	Spatio-temporal	Amount of precipitation.
Traffic	01/01/2020 to 02/28/2022	21GB	2,300,000,000	Speed	INRIX	5 minutes	Spatio-temporal	Recorded road segment traffic speed
Holidays	01/01/2020 to 04/06/2022	1MB		School breaks	calendar	1 day	Tmporal	Scheduled school breaks and holidays
				National holidays	calendar	1 day	Temporal	National holidays

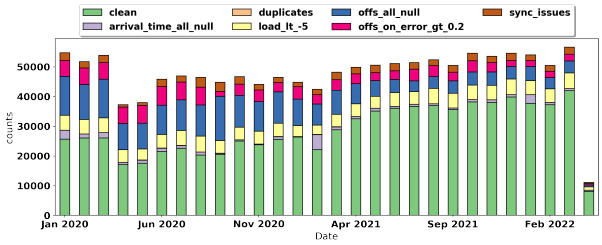

Fig. 3. Count of various issues faced when dealing with APC data

maximum error of as much as 72. Another more pressing issue with APC systems is that data is not reliably obtained. Figure 3 shows the only a fraction of the data received is in a "clean" state or a state without issues that could even be attempted to use for training. The issue is continuously improving over time as better APC maintenance practices are implemented, currently there are only less than 20% of data that is lost and deemed unclean compared to almost 50% in the past.

Thus, prior to any processing and merging, we filter out trips which have incorrect or unclean entries. Note that while APC data is received at the stop level (events are logged every time a bus reaches a stop), cleaning and filtering are done at the trip level. This is due to the fact that we remove entire trips when any of their stop entries meet certain criteria. The following are the set of rules that determine whether an entry in the APC is valid:

- Recorded occupancy is < -5.
- `offs` and `ons` error is > 0.2.
- All actual arrival times are null.
- All `offs` are null.
- The entire trip is a duplicate of a prior trip.
- Stop entries are not in the expected chronological sequence, which can sometimes happen when vehicles lose GPS signal.

If an entry matches any one of these rules, then it and the entire trip it belongs to are considered invalid and filtered out. Once a valid APC dataset has been established, it is then merged with all the other datasets. Two sets of data are then prepared, one for the trip level and another for the stop level.

Since APC data is recorded every time a bus arrives at a stop, it needs to be aggregated into specific trips before it can be used for trip level occupancy prediction. It is first grouped per `transit date` and `trip id`, and aggregated as follows and then be used in trip level occupancy prediction.:

- weather: mean weather across all stops since it does not change within the duration of the trip.
- headway: mean headway across all stops.
- occupancy: maximum occupancy across all stops in the trip.
- others: use the first instance as the value.

For the stop level prediction, `zero load at end` an extra feature, which is not present in the trip level data is used. This feature defines whether a trip would require all passengers to alight upon reaching the final stop. The feature is useful for maintaining continuity between trips within a block. Table I lists down all the features collected and generated from the multiple dataset used in this paper.

Another challenge faced when using APC data for forecast and prediction is the need to sort before any training can begin. In the course of an entire service day multiple vehicles will be travelling across the city, many of the trips occurring simultaneously. Certain blocks are non-overlapping are traversed in sequence by a single vehicle, while others are independent. There might exist multiple routes under each block, each with its own trips that need to be arranged properly before a model such as an LSTM can be used. Otherwise, the data would be disjoint and the model would not be able to learn correctly.

All of our code is public and available here: https://github.com/smarttransit-ai/mta_occupancy_prediction

V. OCCUPANCY PREDICTION MODELS

Recall our goal is to predict passenger occupancy on public transit buses and help transit agencies plan and optimize their trips accordingly. We accomplish this by designing two different models that handle either the stop or trip level ridership demand forecasting. We train and evaluate each model

separately. Ultimately, we want to minimize the prediction error for each of the models. Error is measured by how far our model's prediction is from the ground truth.

The ground truth is the occupancy recorded by the APC data. Based on conversations with the transit agency they are interested in primarily identifying trips and stops with a high occupancy count. One of the outputs of this work will be to show potential riders how crowded the arriving bus will be. Thus, a binned output based on the absolute load is sufficient for this problem. We classified the loads based on how the agency breaks it down as well: Low: ≤ 6, Medium: $7-12$, Medium-High: $13-54$, High: $55-75$ and Very-High: ≥ 76. However, given the heterogeneity of the buses used in a public transit system, vehicle capacities are not uniform. Thus, using only absolute loads will not provide enough information regarding the crowdedness of a particular bus. One solution should be to factor in the vehicle capacity after inference and provide a crowdedness factor to the user instead of the absolute load.

A. Feature Selection

We start with an initial list of 14 features ranging from transit information such as trip date, time and direction, to weather and traffic. Features are treated as one of three categories: numerical, one-hot encoded and ordinal. Numerical values include traffic and weather. These values are scaled and normalized before they are used in training. One-hot encoded features include binary features such as is it a holiday, a school break, zero load at end, and also route id and direction and time window. Using one-hot encoding, we can transform these categorical variables into numerical ones while preventing the models from treating one category as greater than the other. On the other hand, we treat year, month, day and hour as ordinal variables where order and sequence are considered. Time windows are not considered ordinal since we want to treat each time window independent of others.

B. Trip Level Prediction

In trip level prediction, the goal is to be able identify, throughout the service day, which trips in a route experience a high number of occupancy. This allows transit agencies to react and adjust their timetables to future trips that will have a drastic change in demand. Throughout the entire service day, multiple buses will be plying the same trip along the same route and direction. The time between bus dispatch is defined as the headway. We can control the granularity of the data by selecting different time windows with larger time windows grouping together more trips. Grouping by time windows allow the model to provide a prediction for a specific trip at a given time window regardless of which vehicle is present. We divided trip level prediction into two different models which we call **day ahead** and **any day** prediction.

1) Day Ahead: In day ahead, we use data from the prior day (24 hours) to generate additional features and then predict the occupancy level for the trips in the future. If we are trying to predict the occupancy at trip t_i then:

- past actual headway percent change of trip t_{i-2} and t_{i-1}
- past load percent change of trip t_{i-2} and t_{i-1}
- past average load of trips $t_{i-P}, \cdots t_{i-1}$, where P is the number of past trips in the same route and direction.
- past average actual headway of trips $t_{i-P}, \cdots t_{i-1}$, where P is the number of past trips in the same route and direction.

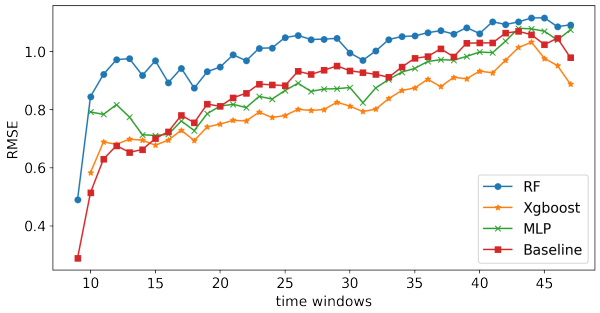

Fig. 4. Model comparison for any day prediction. Xgboost root-mean-square error (RMSE) outperforms all others across all time windows.

Training These features along with the features in Table I are then used as input features in training multiple models ranging from Random Forest, MLP, and an Xgboost model. Figure 4 shows all the RMSE for occupancy prediction across time windows for three models compared to the baseline. Xgboost outperforms all in this preliminary experiment, thus will be used from here on.

Inference Inference is done on a per trip basis, by providing the transit scheduled for the desired trip t_{i+1}, past information, weather and traffic forecast, the output would be the max occupancy of trip t_{N+1}. By doing this for all time windows, the transit agency can have an overview of the maximum occupancy at each route and direction across the entire day.

2) Any Day Trip Prediction: This model is used to predict the maximum load occupancy for any trip at any day in the future. This model is similar to the previous model. However, this model does not rely on any past information to generate a prediction. It is trained using the same type of XGBoost model as the day ahead prediction.

C. Stop Level Prediction

In contrast to the previous two models, stop level attempts to forecast the occupancy at future stops. When used with the trip level prediction, the goal is that it will allow transit agencies to have a more fine-grained view of which stops have a high passenger demand. It uses the stop level dataset generated in Section IV as input to our model. The time window is used to group vehicles that travel the same route and direction.

The data is grouped by transit date, route id, direction, stop id and time window. The occupancy data is then summed across all stops in the same group, giving us an overall idea of the occupancy at that stop for that time window. Similar to the trip level prediction loads are then assigned into the following bins: Low: ≤ 5, Medium: $6-11$, Medium-High: $12-16$, High: $17-29$ and Very-High: ≥ 30. The goal of this

model is then to predict the binned maximum occupancy for stops ahead given past p stops.

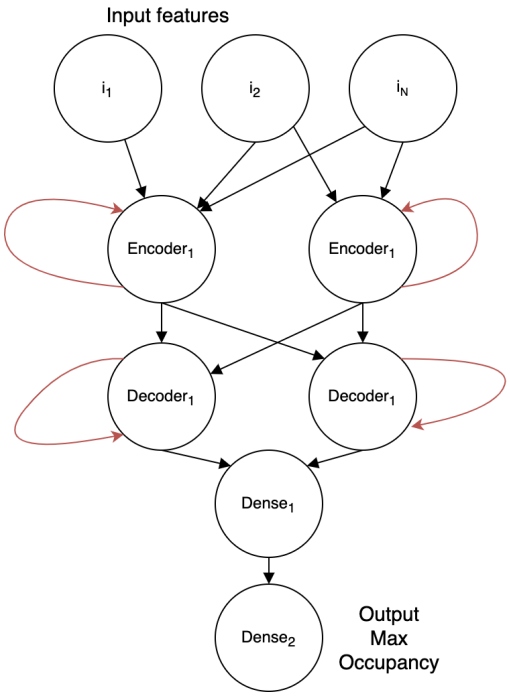

Fig. 5. LSTM Encoder-Decoder architecture

Training: Our proposed machine learning model is an LSTM encoder-decoder, see Figure 5, with two recurrent modules and two feed forward modules which output a bin corresponding to the maximum occupancy at the stop. An encoder-decoder or seq2seq model is selected to be able to leverage its ability of transforming input into some latent space and using a decoder to create a sequence from those inputs. In essence we are using the model to generate the next word in a sentence, where words are stops and the sentence are the trips. The model is trained on the past stops which are then used to predict the immediate next stop.

Inference: We use the past N stops to predict the next stop ahead in a single trip. Weather and traffic forecast are combined with scheduled transit date time for use in the prediction. The output is the predicted occupancy at the stop in a particular route and direction.

VI. Results and Discussions

In this section, we evaluate our models based on real-world public transit data from Nashville, TN. We describe our experimental setup and then present the results for the trip level and then stop level predictions.

A. Experimental Setup

We use APC data for Nashville, TN provided by Nashville Metropolitan Transit Authority (MTA). We used 28 months of data from January 2020 to April 2022. Across these two years, MTA has an average of 100 unique vehicles, serving 30 routes going in 10 different directions in a single service day (counting both weekdays and weekends and holidays). In this work, we used all possible route and direction combinations present in the dataset. All training and experiments were done on a machine with 16-core AMD CPU and 4 Nvidia Titan Xp. We measure error as the distance between the ground truth and predicted occupancy. We treat the binned classes as ordinal thus, we use:

$$y_{error} = y_{true} - \hat{y} \tag{1}$$

Predictions that are far from the truth have larger errors than predictions that are off by a single bin.

B. Trip Level Prediction

For the trip level prediction, we split the 430,404 trip data into 70% training and 30% testing. Trip level prediction model is generated using multiple algorithms such as Random Forest, MLP, LSTM, and XGBoost. Each model is scored based on a 5-fold cross-validation and compared with a baseline model. The baseline model used statistical analysis on historical data. We looked at the past trips taken along the same route and direction, then we get the maximum occupancy across all of those past trips which is then binned. We did this for all trips one, two and four weeks in the past. We found that accuracy does not improve across different baselines. The XGBoost model performs the best compared to all the other models and the baseline.

Grid search based on a 5-fold cross-validation is done to select the best hyperparameters for the model. We tested different time windows for aggregation and at every time window, the model performed better than the baseline. Figure 7 shows the counts of mean absolute errors of the predicted bin to the real bin for both the baseline and day ahead model. Our model is able to predict more trips correctly across all time windows. The model provides 40% more correct predictions and makes 29% less $y_{error} = 2$ mistakes. Figure 8 shows the RMSE of each model across time windows. This further proves that our models perform better, however, it also proves that even with the past information included in the day ahead prediction models, it only performs marginally better than the any day model. Note that the gap is due to the buses being unavailable at those hours (1:30 am to 4:00 am).

To understand which features have an effect on the final prediction models, we generated a SHAP analysis for the day ahead model. It shows that the feature with the highest impact on the model output is route and direction. This is expected since certain routes experience more demand than others due simply to the fact that these routes feature destinations that expect a lot of commuters. The next highest ones are hour and month which are due to jobs and schools having a direct effect in demand. Aside from past trip loads, all other past information had little to no impact to the overall model output. Certain features such as school breaks and national holidays also had less impact since these features essentially have the same relationship with transit demand as month.

Since one of the end goals of this work is to help the transit agency plan for sudden high occupancy events, we evaluate the

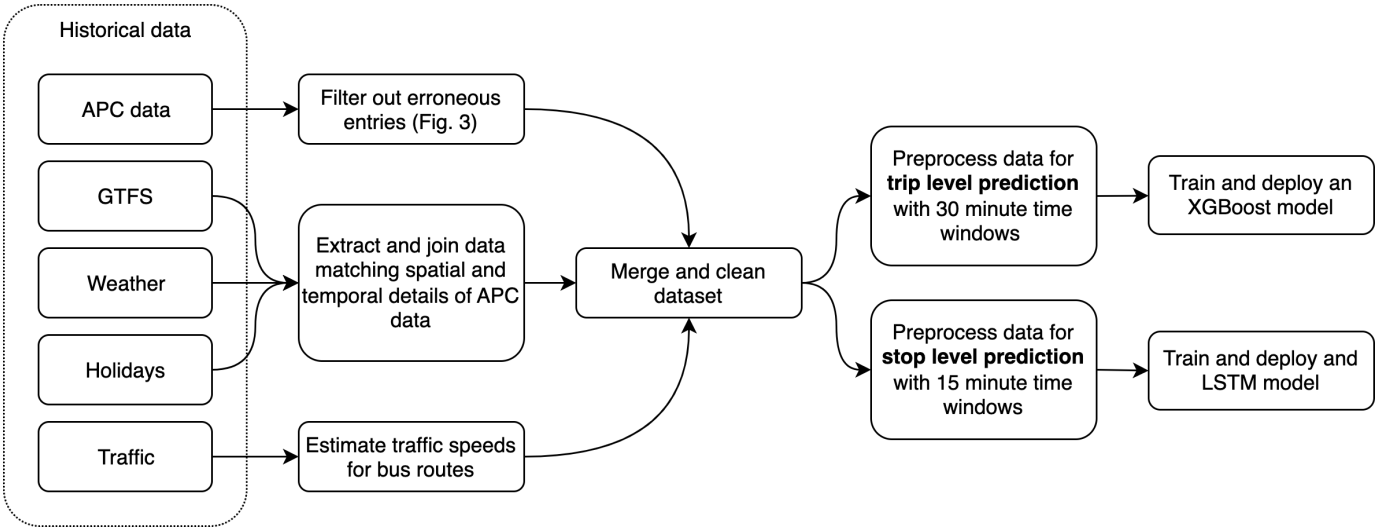

Fig. 6. End-to-end framework showing how historical data is extracted, filtered, merged, and cleaned for use in both the trip level and stop level prediction models. Once the models have been trained and deployed, they can be used to continuously predict passenger occupancy. Once new data from the current service day is available, the process is repeated, improving the performance of the system over time.

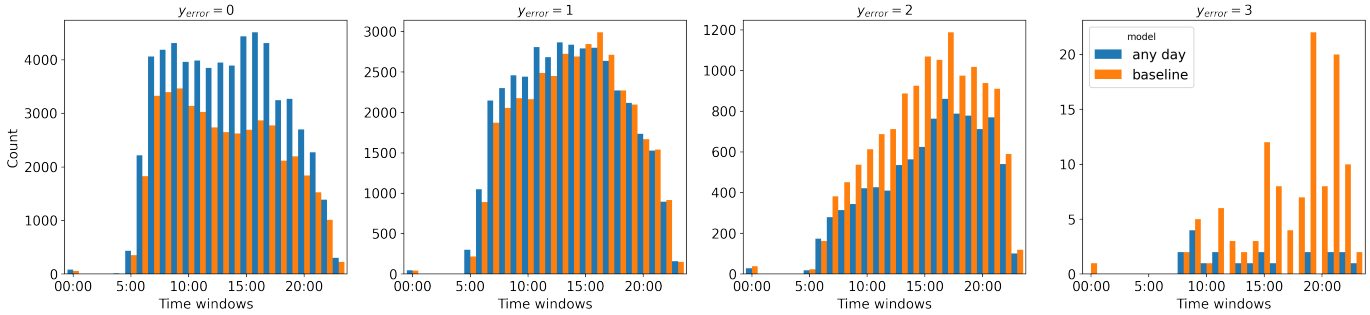

Fig. 7. Comparison between baseline and any day trip level models. From left to right, the plots show the counts of mean absolute error, $y_{error} = \{0, 1, 2, 3\}$, of the predicted to the true label. The model is able to provide 40% more correct predictions and 29% less mistakes than the baseline.

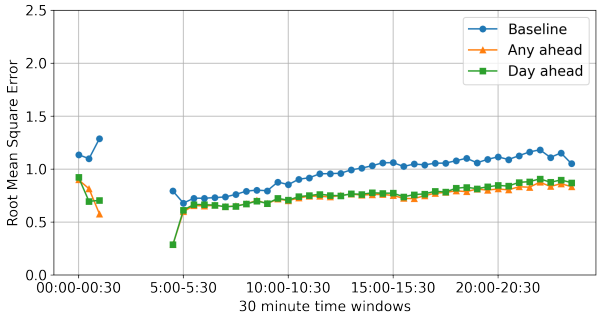

Fig. 8. RMSE of baseline, day ahead, and any day models. The gap in the data is due to buses not travelling at those hours (1:30 am to 4:00am).

ability of the model to distinguish between low (0-11) and high (12-100) number of occupants. In Table II we summarize the precision, recall and F1 scores for the any day prediction given different time windows. The model is able to distinguish high and low occupancy 61% of the time. We can see the effect of increasing the time window has on precision and recall which

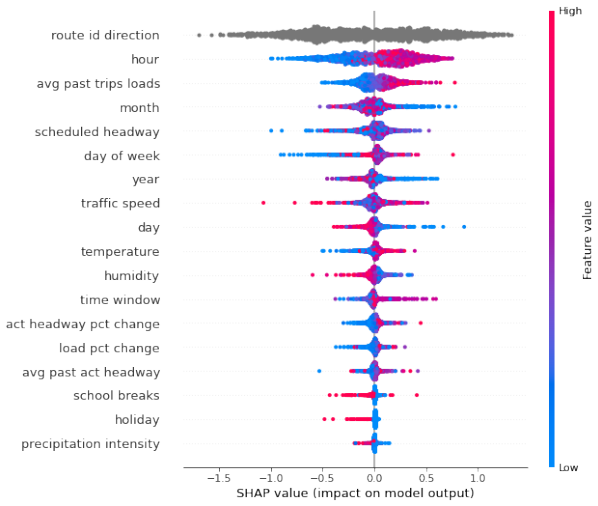

Fig. 9. SHAP Feature Analysis for the trip level model

TABLE II
EFFECT OF VARYING TIME WINDOW ON ANY DAY PREDICTION OF LOW VS. HIGH OCCUPANCY

time window (min)	precision	recall	F1 score
1	**0.5860**	0.6289	0.6066
10	0.5833	0.6357	0.6083
20	0.5750	0.6511	0.6107
30	0.5693	0.6615	0.6120
40	0.5526	0.6884	0.6131
50	0.5469	0.6989	**0.6136**
60	0.5419	0.7064	0.6133
120	0.5043	**0.7695**	0.6093

TABLE III
EFFECT OF VARYING PAST STOPS WITH CONSTANT TIME WINDOW (15 MINUTES) ON LOW VS. HIGH OCCUPANCY PREDICTION OF NEXT STOP

past stops	precision	recall	F1 score
1	0.9276	0.9435	0.9355
3	**0.9697**	0.9381	**0.9536**
5	0.9407	**0.9623**	0.9514
10	0.9428	0.9569	0.9498

is expected since having smaller time windows result in finer grained predictions which can approximate the ground truth better. While the differences in F1 scores are negligible, the most accurate time windows are those between the extremes.

C. Stop Level Prediction

For stop level prediction, we split 17M rows of data into the following:

- Training: 2020-01-01 to 2021-06-30
- Validation: 2021-06-30 to 2021-10-31
- Testing: 2021-10-31 to 2022-04-06

While COVID-19 has had a negative impact on public ridership in the last couple of years [21], we found that this division of training, validation, and testing is still preferred to maximize the amount of available data for training.

A hyperparameter search was done to identify the optimal learning rate, batch size, size of hidden layers, and number of past stops to use in predicting the next stop. The model is compared to multiple baseline models, a simple rolling baseline where only the immediate past stop occupancy is used, a statistical analysis based baseline which gets the max or mean occupancies of the stop in the past (matching route, direction, time window and day of week). Both the number of errors and root mean squared error were used to evaluate the model.

TABLE IV
EFFECT OF VARYING TIME WINDOWS STOPS WITH CONSTANT PAST STOPS (5 STOPS) ON LOW VS. HIGH OCCUPANCY PREDICTION OF NEXT STOP

time window	precision	recall	F1 score
15	0.9008	0.9077	0.9042
30	0.9273	0.8870	0.9067
45	0.9783	0.9000	0.9375
60	**0.9881**	0.9540	**0.9708**
90	0.9583	**0.9583**	0.9583

Similar to the any day model, we choose various values of past stops and time windows. We see in Tables III and IV that using different hyperparameters had an effect on the prediction ability of the model. However, the difference between the values are very small and almost negligible.

For evaluation we uniformly select 5000 random trips which have at least 10 stops in a trip. We then compare the baselines with the model trained with the hyperparameters resulting from the grid search. Figure 10 shows the ability of the model to predict the 6th stop given the preceding 5. It is able to predict more accurately than even the rolling baseline.

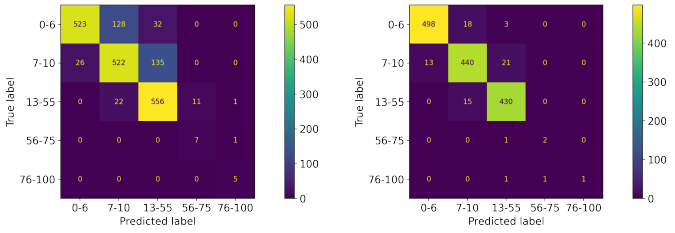

Fig. 10. Confusion matrices when predicting the 6th stop (left) using a rolling baseline, (right) using the past 5th stops as input to the model.

In contrast to the rolling baseline which can only predict the next stop, our model is able to predict any number stops given an initial seed of past stops. Figure 11 shows the error count as the number of predicted stops increase. The baseline mean is unable to generate a prediction for stops in the future. The difference in counts is due to not being able to find past data that matches the features of the future stop.

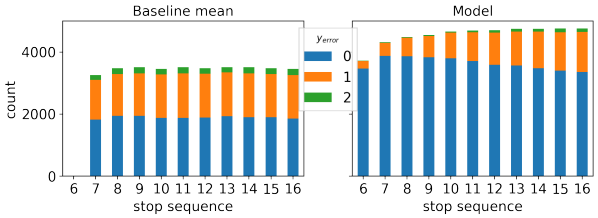

Fig. 11. Count of errors per stop in the future

Finally in Figure 12, we show that the results from the model stay consistent throughout different months.

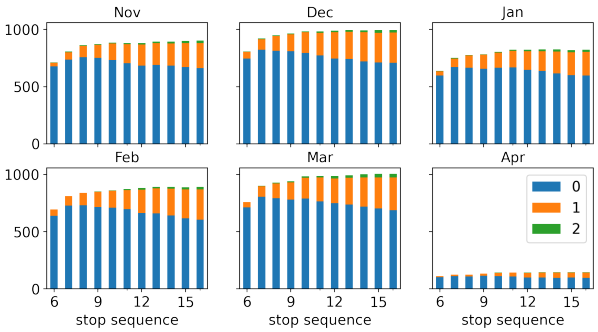

Fig. 12. Count of mean absolute errors per month (2022) per stop in the future

VII. Conclusion

The ability to predict and forecast transit occupancy accurately is a boon not only to passengers but to transit agency. Passengers will be able to adjust their schedules or plans to meet their comfort requirements. Transit planners will have the opportunity to allocate resources much more efficiently. However, predicting occupancy is a non-trivial task. Due to the difficulty of this problem, we proposed to utilize not only available data from the transit agencies, automated passenger counter (APC), but to leverage any additional datasets that can provide further insight in the ridership demands. In this paper, we presented a way to collect, process and augment data from the transit agency, and merge it with traffic, weather, general transit feed specification (GTFS) to obtain some meaningful compilation of data. Our key contribution is proposing two separate models for predicting the trip and stop level occupancy. We found that we are able to outperform baseline statistical analysis using the trained models.

Acknowledgment

This material is based upon work sponsored by the National Science Foundation under Award Number 1952011 and the Federal Transit Administration COVID-19 Research Grant under Federal Award Identification Number TN-2021-015-00.

References

[1] H. S. Levinson, *Analyzing transit travel time performance*, 1983, no. 915.

[2] E. Echaniz, R. Cordera, A. Rodriguez, S. Nogués, P. Coppola, and L. dell'Olio, "Spatial and temporal variation of user satisfaction in public transport systems," *Transport Policy*, vol. 117, pp. 88–97, Mar. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967070X22000038

[3] M. Siebert and D. Ellenberger, "Validation of automatic passenger counting: introducing the t-test-induced equivalence test," *Transportation*, vol. 47, no. 6, pp. 3031–3045, Dec. 2020. [Online]. Available: http://link.springer.com/10.1007/s11116-019-09991-9

[4] J. Patnaik, S. Chien, and A. Bladikas, "Estimation of Bus Arrival Times Using APC Data," *Journal of Public Transportation*, vol. 7, no. 1, pp. 1–20, Mar. 2004. [Online]. Available: http://scholarcommons.usf.edu/jpt/vol7/iss1/1/

[5] T. J. Kimpel, J. G. Strathman, D. Griffin, S. Callas, and R. L. Gerhart, "Automatic passenger counter evaluation: Implications for national transit database reporting," *Transportation Research Record*, vol. 1835, no. 1, pp. 93–100, 2003. [Online]. Available: https://doi.org/10.3141/1835-12

[6] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: Comparison of modeling approaches," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 261–266, 1997.

[7] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record*, vol. 1644, no. 1, pp. 132–141, 1998.

[8] Z. Gong, B. Du, Z. Liu, W. Zeng, P. Perez, and K. Wu, "SD-seq2seq : A Deep Learning Model for Bus Bunching Prediction Based on Smart Card Data," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. Honolulu, HI, USA: IEEE, Aug. 2020, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/9209686/

[9] Q. Ouyang, Y. Lv, J. Ma, and J. Li, "An LSTM-Based Method Considering History and Real-Time Data for Passenger Flow Prediction," *Applied Sciences*, vol. 10, no. 11, p. 3788, May 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/11/3788

[10] F. Toque, M. Khouadjia, E. Come, M. Trepanier, and L. Oukhellou, "Short & long term forecasting of multimodal transport passenger flows with machine learning methods," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. Yokohama: IEEE, Oct. 2017, pp. 560–566. [Online]. Available: http://ieeexplore.ieee.org/document/8317939/

[11] T.-H. Tsai, C.-K. Lee, and C.-H. Wei, "Neural network based temporal feature models for short-term railway passenger demand forecasting," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3728–3736, Mar. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417408001516

[12] Z. Wang, X. Su, and Z. Ding, "Long-term traffic prediction based on lstm encoder-decoder architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561–6571, 2021.

[13] S. Karnberger and C. Antoniou, "Network–wide prediction of public transportation ridership using spatio–temporal link–level information," *Journal of Transport Geography*, vol. 82, p. 102549, Jan. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S096669231930376X

[14] M. Zhou, D. Wang, Q. Li, Y. Yue, W. Tu, and R. Cao, "Impacts of weather on public transport ridership: Results from mining data from different sources," *Transportation Research Part C: Emerging Technologies*, vol. 75, pp. 17–29, Feb. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X16302492

[15] J. Wood, Z. Yu, and V. V. Gayah, "Development and evaluation of frameworks for real-time bus passenger occupancy prediction," *International Journal of Transportation Science and Technology*, Mar. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2046043022000296

[16] X. Ma, Y.-J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders' travel patterns," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 1–12, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X13001630

[17] J. Y. Park, D.-J. Kim, and Y. Lim, "Use of smart card data to define public transit use in seoul, south korea," *Transportation Research Record*, vol. 2063, no. 1, pp. 3–9, 2008. [Online]. Available: https://doi.org/10.3141/2063-01

[18] B. McHugh, "Pioneering open data standards: The gtfs story," in *Beyond Transparency: Open Data and the Future of Civic Innovation*. Code for America Press, 2013, ch. 10, pp. 125–135.

[19] INRIX. Leading transportation analytics solutions — inrix. [Online]. Available: https://inrix.com/

[20] S. A. Kashfi, J. M. Bunker, and T. Yigitcanlar, "Understanding the effects of complex seasonality on suburban daily transit ridership," *Journal of Transport Geography*, vol. 46, pp. 67–80, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0966692315000897

[21] M. Wilbur, A. Ayman, A. Ouyang, V. Poon, R. Kabir, A. Vadali, P. Pugliese, D. Freudberg, A. Laszka, and A. Dubey, "Impact of covid-19 on public transit accessibility and ridership," in *Preprint at Arxiv*, 2020.