Forecasting and Mitigating Disruptions in Public Bus Transit Services

Chaeeun Han
Pennsylvania State University
University Park, PA, USA
cfh5554@psu.edu

Jose Paolo Talusan
Vanderbilt University
Nashville, TN, USA
jose.paolo.talusan@vanderbilt.edu

Dan Freudberg
WeGo Public Transit
Nashville, TN, USA
Dan.Freudberg@nashville.gov

Ayan Mukhopadhyay
Vanderbilt University
Nashville, TN, USA
ayan.mukhopadhyay@vanderbilt.edu

Abhishek Dubey
Vanderbilt University
Nashville, TN, USA
abhishek.dubey@vanderbilt.edu

Aron Laszka
Pennsylvania State University
University Park, PA, USA
alaszka@psu.edu

ABSTRACT

Public transportation systems often suffer from unexpected fluctuations in demand and disruptions, such as mechanical failures and medical emergencies. These fluctuations and disruptions lead to delays and overcrowding, which are detrimental to the passengers' experience and to the overall performance of the transit service. To proactively mitigate such events, many transit agencies station substitute (reserve) vehicles throughout their service areas, which they can dispatch to augment or replace vehicles on routes that suffer overcrowding or disruption. However, determining the optimal locations where substitute vehicles should be stationed is a challenging problem due to the inherent randomness of disruptions and due to the combinatorial nature of selecting locations across a city. In collaboration with the transit agency of Nashville, TN, we address this problem by introducing data-driven statistical and machine-learning models for forecasting disruptions and an effective randomized local-search algorithm for selecting locations where substitute vehicles are to be stationed. Our research demonstrates promising results in proactive disruption management, offering a practical and easily implementable solution for transit agencies to enhance the reliability of their services. Our results resonate beyond mere operational efficiency—by advancing proactive strategies, our approach fosters more resilient and accessible public transportation, contributing to equitable urban mobility and ultimately benefiting the communities that rely on public transportation the most.

KEYWORDS

Public transportation, Data-driven optimization, Disruption forecasting, Simulation, Metaheuristic optimization

1 INTRODUCTION

Modern urban ecosystems are dependent on public transit, which gives millions of people access to mobility, influences settlements, and encourages sustainable growth. Despite their necessity, public transportation systems frequently struggle to strike a balance between available infrastructure and constantly shifting demand [1]. This presents a particular challenge for mid-size U.S. cities, which typically have sparse transit networks, and where there is significant inequity in how communities rely on transit networks. Even though it is inevitable, crowding during rush hour results in longer commute times, which is detrimental to the passengers' experience. In contrast, services tend to remain underused at non-peak times, which results in operational inefficiencies, especially in less populated suburban areas [18, Chapter 8.4]. Unexpected transit disruptions caused by a variety of reasons, such as vehicle breakdowns, operational issues, or medical emergencies, compound these difficulties (Figure 1). These disruptions cause cascade effects across the entire transit network by extending travel times and disturbing established traffic patterns [20]. Both riders and operators are heavily impacted by these disruptions, highlighting the urgent need for proactive mitigation methods.

The problem setting that we consider has two distinct challenges: first, we seek to forecast disruptions in space and time, which we refer to as the *forecasting problem*, and second, we seek to optimize the stationing of substitute buses to ensure that they can promptly respond to disruptions, which we refer to as the *stationing problem*.

Figure 1: Disruptions, such as traffic accidents, hamper the reliability of the service provided by our partner transit agency. This image shows an actual disruption from 2020.

However, forecasting disruptions is challenging due to the scarcity of data. The relative rarity of disruptions means there is a limited amount of data to draw conclusions from using conventional machine learning models [29]. While prior work has explored the impact of disruptions on public transportation [9, 24, 27], there is significantly less work on attempting to predict disruptions [28]. While Yap and Cats [28] recently explored disruptions in public transit, our setting is significantly more challenging from the perspective of data availability—Yap and Cats [28] focus on disruptions in metro (subway) networks, which have more than 100 types of disruptions, as opposed to only a handful in the case of bus networks (our partner agency has 20 types of disruptions), and the frequency of disruptions in metro networks is almost thrice as high as in our problem setting.

As for the challenge of bus stationing, based on conversations with our partner transit agency, we learned that there are no formal policies for where buses are to be stationed at the start of each day. Agencies rely on a combination of historical ridership data and domain expertise to identify potential pain points across the city. Currently, our partner agency stations its buses at locations mainly near the central terminal and at high-frequency lines which experience consistent overcrowding. The stationing problem is challenging due to the combinatorial nature of the search space (i.e., substitute vehicles can be stationed at arbitrary stops) and the inherent uncertainty of the disruptions. The problem therefore must maximize expected utility[1] while considering a multitude of factors such as passenger demand, geographic locations, traffic patterns, and operational constraints [17]. As the combinatorial search space grows exponentially with the growth of urban areas, transit agencies need scalable approaches to dynamically station vehicles. Most prior work has looked at this problem from a purely demand-responsive setting [31] (where passengers reserve rides in buses, similar to taxis), which is fundamentally different from stationing buses for mitigating the effects of unexpected disruptions.

In this paper, we tackle the twin challenges of forecasting disruptions and stationing substitute vehicles in collaboration with the transit agency of Nashville, TN, a mid-size U.S. city. Specifically, we seek to estimate the likelihood of disruptions in fixed-line bus transit, and given the likelihood, optimize the stationing of substitute buses to mitigate the effects of disruptions. By using our approach, the transit agency will identify routes that are prone to disruptions and ensure that the optimal allocation of substitute buses will minimize the effects on passengers and increase operational efficiency. Specifically, we make the following contributions: **1)** We create data-driven statistical models that can estimate the probability that a given future transit trip will encounter a disruption. **2)** We formulate the stationing problem and introduce efficient heuristic algorithms to find stationing locations that minimize the impact of disruptions on passenger experience.

The remainder of this paper is organized as follows. Section 2 formulates the problem of forecasting disruptions and optimizing vehicle stationing in public transit. Section 3 introduces our proposed computational approaches for forecasting disruptions and optimizing stationing. Section 4 describes our real-world data and

experimental setup. Section 5 presents numerical results, demonstrating the accuracy of our forecasts and the improvement in transit-service operations due to optimizing stationing. Section 6 gives a brief overview of related work. Finally, Section 7 provides concluding remarks.

2 PROBLEM FORMULATION

In this section, we describe the characteristics of fixed-line transit, introduce disruption forecasting, and formulate the vehicle stationing problem. Fixed-line public transit relies on a set of vehicles running on a consistent and predictable schedule of trips to service passengers. Any events or incidents that delay or disrupt buses can impact their reliability leading to increased wait times, interfere with transfers, and increase uncertainty. As a contingency, transit agencies have a limited number of substitute buses, which are deployed in response to disruption and overcrowding events. Having the ability to forecast possible disruptions allows transit agencies to station substitute buses in anticipation of such events.

Trip: Fixed-line transit vehicles follow a scheduled assignment of trips. A trip pertains to individual occurrences of a bus navigating its assigned route. For a given route $r \in R$, there is a set $T(r)$ containing all trips on the route. A trip t represents a specific instance of a bus traveling a route in a particular direction at a given time. For example, the trip t of the first bus operating on Sundays for route r symbolizes a specific occurrence, distinct from other trips on that route.

Route + Direction: A bus route, combined with its direction, is a group of trips that represents a predetermined pathway recurrently followed. The set R contains all unique combinations of routes and directions in the system, where each combination $r \in R$ uniquely signifies a specific pathway and direction. For example, outbound and inbound buses might share identical stops, but since they have different points of origin and termination, they are distinct in R.

Stop: Bus stops are strategically positioned along the route, and they enable the bus service to cater to a wide spatial region. For a given trip $t \in T(r)$ on route $r \in R$, there is a set $S(t)$ of stops for that trip. For a mid-size U.S. city, it is common for some routes to share one or more stops. This sharing often aligns with many transit agencies' practice of following a spoke-hub model which is often cheaper to develop [7], where most routes have at least their origin or destination in a central, downtown terminal.

2.1 Disruption Forecasting

To assess the likelihood of a disruption occurring, we formulate disruption forecasting as a binary classification problem. However, rather than merely categorizing trips as having a disruption (1) or not (0), we focus on estimating the probability that a given trip will be classified as having a disruption. This approach enables a more granular understanding of the risks associated with different trips.

We assume that we have a dataset of past trips represented as $T = \{t_1, t_2, ..., t_n\}$. Each trip $t_i \in T$ is characterized by a vector of features X_{t_i}, which includes categorical and numerical variables. Categorical variables include route identifier combined with its direction(R_{t_i}) which is a specific route number followed by its direction name, ridership (P_{t_i}), passenger occupancy on the trip,

---

[1]We define utility later, while formalizing the problem.

service-time windows (S_{t_i}), and calendar-related variables representing the date of the trip that contains year (Y_{t_i}), month (M_{t_i}), and day of week (W_{t_i}). Numerical variables capture weather information such as precipitation intensity (I_{t_i}) and temperature. (T_{t_i}) Thus, we can represent X_{t_i} as follows:

$$X_{t_i} = (R_{t_i}, P_{t_i}, S_{t_i}, Y_{t_i}, M_{t_i}, W_{t_i}, I_{t_i}, T_{t_i}) \tag{1}$$

Also, each t_i has a corresponding label y_{t_i} indicating whether a disruption was reported during the trip.

The goal of the prediction model is to learn $\Pr[y_{t_i} = 1 \mid X_{t_i}]$ from a function f mapping the feature vectors to the labels, $f : X_{t_i} \rightarrow y_{t_i}$, using the dataset as the training data. However, the inherent randomness of disruptions, coupled with the fact that the occurrences of disruptions are very rare compared to the number of bus operations, inevitably makes forecasting disruptions a challenging task.

2.2 Vehicle Stationing

On a typical transit day, the transit agency has access to two different types of buses, regular and substitute buses. **Regular buses** are buses that have been scheduled to serve trips across the city for a given day. Once a regular bus has finished its shift for the day, they are not available for dispatch. Regular buses travel across their designated route, picking up and dropping off passengers at stops along their trip. Collectively, buses will pick up and drop passengers across J stops in the city. Passengers can be affected due to two types of events: first, at times, buses cannot accommodate more passengers due to crowding, resulting in *overage events* that are temporary, or second, buses can encounter disruptions such as mechanical failures or accidents that render them unusable for the rest of the day, which we refer to as *disruption events*. **Substitute buses** are buses that are held in reserve for the purpose of being dispatched to cover either of these events. These start their day at the main depot, travel to a predetermined stationing stop to wait for dispatching, and then end the day at the main depot. Substitute buses traveling from the main depot to a stationing stop, or from a stationing stop to a dispatch location, are not considered to be in service. The distance traveled and time spent by buses when not in service are called deadhead miles and deadhead time respectively.

Our conversations with the transit agency reveal that existing stationing strategies are largely *ad-hoc*, i.e., there are no formal and established policies for stationing. Buses are traditionally stationed by leveraging summary statistics from historical data (e.g., frequency of disruptions on a route) and domain expertise of transit operators. Similarly, there are no principled policies for dispatching substitute buses; buses are dispatched simply based on availability, with higher priority given to disruption events over overage events. Due to this, experts often consider contexts such as headways, traffic congestion, and forecasted ridership, before they either **dispatch a substitute vehicle** or **do nothing** in response to an event.

We present a principled optimization formulation for the stationing problem. Formally, consider a subset of stops $S_{station}$ where buses can be stationed using the agency's budget of k substitute buses. A feasible solution $\bar{x}$ is a k-subset of $S_{station}$ (i.e., $\bar{x} \subseteq S_{station}$ such that $|\bar{x}| = k$). Our goal is to find the subset of stops to station the substitute buses that will minimize their total non-service miles (D) and non-service travel duration (T) while minimizing the total

passengers left behind (L) across all stops. Formally, the problem is:

$$\operatorname{argmin}_{\bar{x} \subseteq S_{station} : |\bar{x}| = k} \mathbb{E}_P \left[D(\bar{x}; P) + T(\bar{x}; P) + \sum_{j=1}^{J} L(j, \bar{x}; P) \right] \tag{2}$$

where P is the random variable that denotes the occurrence of disruptions and overcrowding events. It is worth noting that this objective is practically impossible to calculate based on first principles. Thus, in practice, we approximate its value by simulating a number of outcomes.

3 PROPOSED APPROACH

Our approach consists of two components: **1)** We employ a multi-faceted approach to predict the likelihood of trip disruptions under given conditions, utilizing a combination of statistical and machine learning methodologies, including logistic regression [2], extreme gradient boosting (XGBoost) [4], and Poisson regression [10]. **2)** We identify the best subset of stops for stationing through a combination of greedy search and simulated annealing [22].

3.1 Disruption Forecasting

Forecasting the likelihood of disruption occurring on a certain trip is a supervised learning problem. We explore two learning-based approaches, namely logistic regression and the XGBoost model. Using the vector of features described in Equation (1), the logistic regression model can be expressed as:

$$\Pr\left[y_{t_i} = 1 | X_{t_i}\right] = \frac{1}{1 + e^{-\beta_0 - \beta_1 \cdot X_{t_i}}} \tag{3}$$

where y_{t_i} is the binary outcome (presence of disruption) for trip t_i, and β_0 and β_1 are the parameters to be estimated. Note that our problem focuses on estimating the likelihood of disruptions in a trip and does not address stop-level prediction; naturally, predicting which stop might face a disruption is practically infeasible due to the sparsity of data. The logistic regression is advantageous in that it is simple and interpretable, so it is well suited for understanding the impact of individual features on the outcome.

Also, the XGBoost model is employed with the same problem settings as logistic regression. It is employed for its robust performance and ability to handle complex nonlinear relationships. Its gradient boosting framework enables the assembly of an ensemble of weak predictive models, making it capable of capturing intricate patterns in the data [5]. Moreover, XGBoost provides a feature importance score, offering valuable insights into factors significantly influencing disruption likelihood. However, the raw output probabilities may not always represent the true likelihood of events. Calibration techniques, such as Isotonic Regression, can be employed post hoc on the model to adjust the probabilities to better reflect the true outcomes, thereby increasing their reliability in practical scenarios. It is a non-parametric approach that can fix both over-confidence and under-confidence cases well [19].

3.2 Vehicle Stationing

Selecting the best locations to station substitute buses is a combinatorial optimization problem that can not be reasonably solved in linear time due to the exponential possibilities available (over 1000 possible stops across the city). Instead, we select a subset of 25

Algorithm 1 Greedy Selection

1: $C \leftarrow$ List of candidate stops
2: $C^* \leftarrow \{\}$
3: $k \leftarrow$ Number of substitute buses
4: **while** $count(\bar{C}) < k$ **do**
5:      $C \leftarrow C \setminus \bar{C}$
6:      $\hat{c} \leftarrow \{\}$
7:      $O = \infty$
8:      **for** $c \in C$ **do**
9:          $\hat{C} \leftarrow c$
10:          $\hat{O} = cost(\hat{C})$ ▷ objective function
11:          **if** $\hat{O} < O$ **then**
12:              $\hat{c} = c$
13:              $O = \hat{O}$
14:      $C^* \leftarrow \hat{c}$
15: **return** $\bar{C}$

Algorithm 2 Neighbor Generation

1: $C \leftarrow$ Candidate stops
2: $\bar{x} \leftarrow$ Current solution
3: $i = random(0, count(S))$ ▷ uniformly at random
4: $\bar{x}[i] = random(C \setminus \bar{x})$ ▷ uniformly at random
5: **return** $\bar{x}$

Algorithm 3 Simulated Annealing Optimizer

1: $N \leftarrow$ Simulated annealing iterations
2: $K \leftarrow K_{max}$
3: $\bar{x} \leftarrow greedy(x)$ ▷ select initial solution using Algorithm 1
4: $O \leftarrow cost(\bar{x})$ ▷ objective function
5: **for** $n = 0$ through N **do**
6:      $\bar{x}_{new} = neighbor(\bar{x})$
7:      $O_{new} = cost(S^*_{new})$
8:      $\Delta O = O_{new} - O$
9:      **if** $accept(\Delta O, K)$ **then**
10:          $\bar{x} \leftarrow \bar{x}_{new}$
11:          $O \leftarrow O_{new}$
12:      $K = \frac{K}{\gamma + n}$ ▷ cool the temperature
13: **return** $\bar{x}$ ▷ best stationing plan

---

stops for stationing based on the current and potential stationing locations that the transit agency uses. We then generate an initial stationing plan using a greedy algorithm, Algorithm 1, that selects the best solutions using the same objective function in Equation (2). The resulting stationing plan is then used as the initial solution to simulated annealing. In simulated annealing, neighboring state iterations are then generated by randomly selecting a single bus, and then assigning it to a new random stop, Algorithm 2, with the constraint that no two buses should be at the same stop at the same time. Algorithm 3 details how we select the solution set that gives the minimum cost after N iterations, where mutations are dependent on the initial temperature selected and the current iteration.

Table 1: Overview of Datasets

Data	Source	Scope	Features
APC	Agency	2020-2023	ridership, stops
Disruption	Agency	2020-2023	GPS location, datetime, stops
GTFS [13]	Agency	2020-2023	routes, stops
Weather	Darksky	2020-2023	location, temperature, precipitation

4 EXPERIMENTAL SETUP

4.1 Real-World Data

To conduct our analysis of disruption prediction, we used disruption data from the transit agency of Nashville, TN, transit schedule data in the form of General Transit Feed Specification (GTFS), Automated Passenger Count (APC) data, and weather data.

Automated Passenger Count data APC data records a variety of information that transit agencies use to monitor bus conditions. It uses infrared sensors along doors which trigger whenever they open for passengers. It records the number of people boarding and alighting at every bus stop along its trip. It also records the actual times it arrives and departs at every stop. This allows operators to react to bus bunching events by adding slack at certain points on the trip or by dispatching substitute buses.

Disruption data The disruption dataset contains comprehensive information about each disruption, including where and when it occurred and the underlying cause. This information includes locations specified by stop names as well as latitude and longitude coordinates, corresponding to the last stop where the bus terminates operation before the disruption. These locations facilitated the integration of disruption data with APC and weather information, ensuring consistency across datasets. This dataset covers the period of March 2020 to January 2023, during which time a total of 5,096 disruptions were recorded.

GTFS data The General Transit Feed Specification (GTFS) dataset offers insights into bus schedules, encompassing all planned transit trips. By aligning this data with disruption data, we could investigate the frequency and patterns of disruptions.

Each transit trip is characterized by unique identifiers depending on the day of the week, time of day, and bus route. This allows us to analyze disruptions with a trip of specificity. As there are no trips between midnight and 4AM, we segmented the 20 service hours into five time windows: early morning (4AM to 6AM), morning (6AM to 9AM), mid-day (9AM to 2PM), afternoon (2PM to 6PM), and evening (6PM to midnight). For the ridership, we divided occupancy rate into four categories: low (0% to 30%), moderate (30% to 60%), high (60% to 100%), and over-capacity (> 100%).

4.2 Disruption Forecasting

The disruption forecasting model uses all data shown in Table 1. For six possible categorical variables which are route identification combined with direction, service-window hours, day of week, passenger capacity, year, and month, we use one-hot encoding to train in the models. For the logistic regression model, we conduct the goodness-of-fit test and build 64 (2^6) models using all possible combinations of the variables. The goodness-of-fit test of the logistic regression model offers insight into how various independent factors affect the probability of disruptions. So, we are able to

choose the optimal combination among categorical variables. After deciding what variables are to be used for the model, we compare the performance of three disruption forecasting models; logistic regression, XGBoost, and XGBoost with Isotonic Calibration. For the experiment, we used Python 3.9.12 and Scikit-learn 1.3.0 [15] for training.

4.3 Simulator

The simulator uses data from APC and GTFS to recreate public transit activities for a single day. There are three main components to the simulator: regular buses, substitute buses, and passengers. Regular buses start and end their day at the main depot after servicing all their trips. They may pick up or drop off passengers at every stop along their trip. Substitute buses start their day at the main depot, travel to their predetermined stationing location, and then end at the main depot. When dispatched to cover for a regular bus, they must travel from their current location to the stop in question, accumulating deadhead miles for a period of time. When covering an overage event, the substitute bus will travel from its current location to some stop $\hat{s}$ where the event was reported. From then on, it will travel the same trip as the bus that reported the overage, until the final stop in the trip. Only upon finishing this would it be available for dispatch again. When dispatched to a bus that has broken down, it travels to the last visited stop of the broken bus, and from then on, covers all the trips that the broken down bus was meant to service. Only upon finishing this sequence of trips would the substitute bus be available for dispatch. Substitute buses are dispatched following a policy. The bus that is *nearest to a reported event will be considered for dispatch*, and ties are broken randomly. If the event that triggered this policy is an *overage event*, then the policy will only dispatch the bus if the number of people left behind is greater than 5% of the current bus' capacity. However, disruption events are handled immediately, if there are available substitute buses. Every time a substitute bus is dispatched, it accrues deadhead miles and deadhead time.

At each stop, a certain number of **passengers** can either board or alight the bus. Passengers can arrive at the stop as early as 10 minutes before the bus arrives and will wait for the bus for a maximum of 30 minutes. Passengers can board any bus arriving at their current location as long as the bus is traveling in the same block, route, and direction, and if bus ridership is less than bus capacity. If passengers are unable to board the bus, they remain at that stop. If a disruption event occurs, passengers currently onboard the bus are unloaded at the last passed stop and will wait for another 30 minutes for another bus to arrive. Any trip on the service day can encounter a disruption, causing the bus to fail at one of the stops along the trip, ending the service. A day can have anywhere from zero disruptions to a disruption at each of its trips.

We address stochasticity in the real world by generating chains of events for each single day. For each chain, we sample passenger counts and disruptions independently. Generative models, trained over ridership at the stop level [21], are used to sample the ridership forecast at each stop. Boarding and alighting counts are then derived from this value. Disruption events are sampled from the disruption forecast model discussed in Section 2.1. We then get the average of performance of our stationing across all the chains for each day.

Table 2: Simulation Parameters

Parameter	Setting
simulation duration	6AM to 1PM
passenger arrival range	within 10 minutes
passenger patience	30 minutes
overage threshold	$\geq 105\%$
average number of buses per day	80.0
number of substitute buses	5
number of chains	100
number of days	50
initial temperature	100
local search iterations	500

This is an event-driven simulator that considers bus arrivals at stops, overages, and disruptions as events that require a dispatch decision. The simulator begins with the first passengers arriving at a stop and proceeds until the final bus returns to the depot for the last stop of the trip. Passengers who have not been picked up at the end of the simulation time are considered to have been left behind. For this paper, we truncate the service day to all trips that start between 6AM and 1PM. Table 2 lists the rest of the configurations used in the experiment. The initial temperature was selected after a hyperparameter search. All simulation experiments were done on Chameleon servers [11] with 187 GB of RAM and an Intel Xeon CPU with 96 cores, each with 2.4GHz.

5 NUMERICAL RESULTS

5.1 Disruption Forecasting

Model Performance. The performance of the predictive models is evaluated using cross entropy as a key metric, where lower values signify better model performance. We compare cross-entropy over accuracy due to the dataset's imbalance, revealing that just 0.2% of all incidents are disruptions ($y_{t_i} = 1$). Accuracy, skewed by the abundance of non-disruptions ($y_{t_i} = 0$), could mislead. Therefore, cross-entropy, being a more informative metric, is used to accurately gauge the model's classification performance, especially concerning the minority class.

As indicated in Table 4, three models undergo evaluation: Logistic Regression, XGBoost, and XGBoost calibrated with Isotonic Regression. XGBoost classifier calibrated with Isotonic regression shows the best performance, attributed to Isotonic Calibration's refinement of the XGBoost probability estimates, aligning predictions more closely with actual outcomes. A grid search for the best hyperparameter values for the XGBoost model found the model performs the best when the learning rate is 0.1, maximum depth is 9, minimum child weight is 1, and subsample is 0.5.

Model and Feature Selection. Five categorical features that are route identifier combined with direction, service-window hours, month, and passenger capacity, combined with numerical features which are weather related-variables like temperature and precipitation shows the optimal result. In detail, logistic regression model – one with five categorical features, excluding only the year, demonstrates the lowest cross entropy value as shown in Table 3. Subsequently, an XGBoost prediction model outperformed our logistic

Table 3: Result of logistic regression models with different feature sets variables

Features (Number of features)	Train Cross Entropy	Test Cross Entropy
Route + Direction, Service Window Hours, Day of Week, Passenger Capacity, Year, Month (6)	0.1320	0.1330
Route + Direction, Service Window Hours, Day of Week, Passenger Capacity, Month (5)	0.0903	0.0912
Route + Direction, Service Window Hours, Day of Week, Passenger Capacity (4)	0.0905	0.0913
Route + Direction, Service Window Hours, Passenger Capacity (3)	0.0946	0.0952
Route + Direction, Passenger Capacity (2)	0.0953	0.0960
Route + Direction, Service Window Hours, Day of Week, Passenger Capacity, Month, Temperature (6)	0.0903	0.0910

Table 4: Results for three disruption forecasting models

Model	Train Cross Entropy	Test Cross Entropy
Logistic Regression	0.0903	0.0910
XGBoost	0.0872	0.0881
XGBoost + Isotonic Calibration	0.0870	0.0876

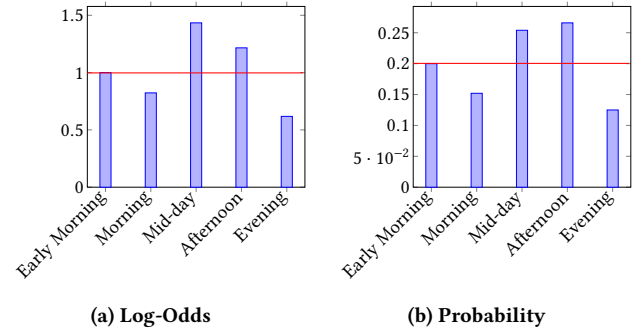**Table 5: p-value of Permutation Test for Number of Disruptions on Trips per Route**

Route	3	4	5	6	7	8	9
3	1.0	-	-	-	-	-	-
4	0.4818	1.0	-	-	-	-	-
5	0.6501	0.7945	1.0	-	-	-	-
6	0.7744	0.8496	1.0	1.0	-	-	-
7	0.6393	0.1294	0.1793	0.2252	1.0	-	-
8	1.0	0.4005	0.5305	0.5778	0.5489	1.0	-
9	0.6683	0.3148	0.2746	0.3366	0.8008	0.5264	1.0

(a) Log-Odds   (b) Probability

Figure 2: Log-Odds and Probability Values for Features.

5.2 Vehicle Stationing

We compare our results to the three different stationing assignments that the public transit agency uses. **Garage** represents the absence of a stationing plan, all substitute buses are kept at the main garage. **Hub** represents a stationing plan where all substitute buses are stationed at the central hub. Finally, **Agency**, represents the current stationing plan being used by the transit agency, based on our conversations with them.

Simulation and random local search. For every iteration of the random local search, we simulate the same day with 100 different chains of events. Figure 4 shows the progress of the objective scores for a single day. The initial allocation generated by the greedy approach results in a lower cost than the baselines, and within 50 iterations, we can find a better stationing selection. Figure 5 shows that the majority of dates, ($> 60\%$), can find the minimal objective cost within the first 100 iterations. Across multiple days, the stationing plans generated by the search algorithm result in much lower scores compared to the other stationing plans. Interestingly, having 5 buses stationed in the main hub can often outperform the searched stationing plan if there are no constraints on stationing. Finally, we compared whether optimizing once for multiple days would have better performance than running the optimizer every day. This is comparable to the agency running the algorithm a single time and then maintaining the same stationing across multiple days. Figure 7 shows that in almost all instances, optimizing for a single day will outperform a solution generated on multiple days.

Stationing using forecasted disruptions. We use the stationing plans generated by the search algorithm on passenger forecast and disruption data and use it on the ground truth data. Disruptions are based on the actual events on which the forecasting model was

regression model, offering a more refined prediction with an even lower cross-entropy value.

Moreover, notably, midday to afternoon service-window hours were more susceptible to disruptions compared to late or early hours as illustrated in Figure 2. Evening hours shows the lowest probability of disruptions. We analyze this impact of each feature by examining the log-odds ratio for route, service-window hours, and month. Positive odds ratio indicates a higher likelihood of disruptions.

Lastly, permutation tests are conducted to validate the significance of these findings. As shown in Table 5, the result indicated that disruptions frequencies among routes follow the same distribution, affirming that our analysis is not accounting for potential data imbalances. These outcomes underscore the robustness of our predictive model, offering valuable insights into understanding and forecasting disruptions in real world scenarios.

Features Importance. The XGBoost model's feature importance scores, presented in Figure 3, identified service-window hours as the most crucial predictor of disruption likelihood. According to the feature-importance results, service-window hours were determined to be the most important feature in this instance, followed by day of week, month, and route ID. These observations can help direct future research and inform decisions based on the output of the model.
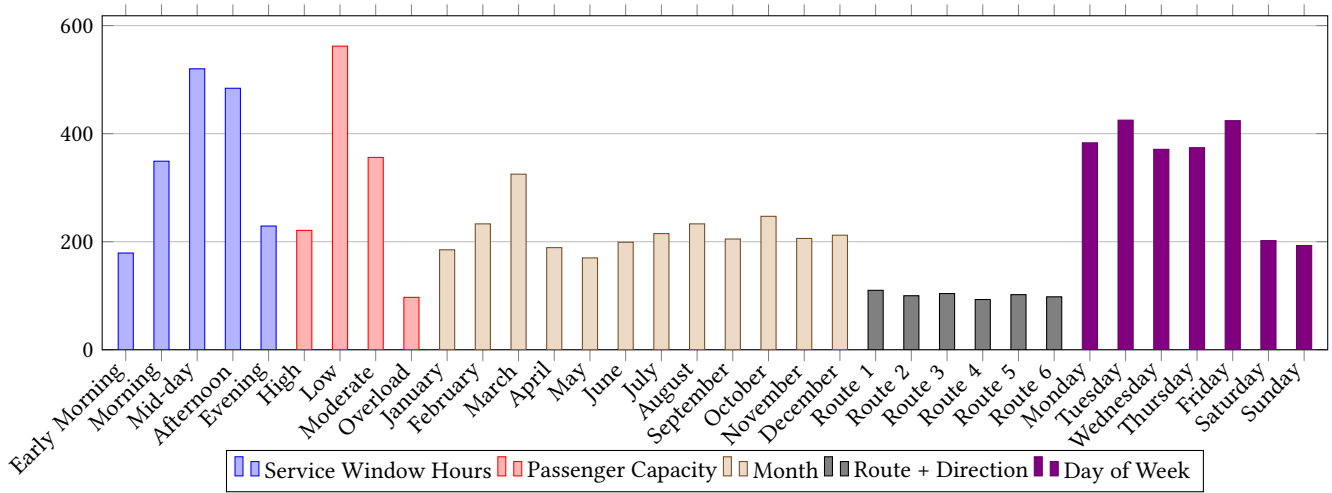
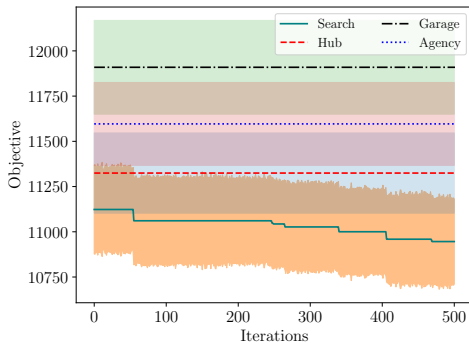Figure 3: Feature importance of one-hot encoded categorical features used for XGBoost model.

Figure 4: Comparison of cost using stationing plans based on search, garage, or agency. Lines and shaded regions represent mean and standard error across 100 chains.
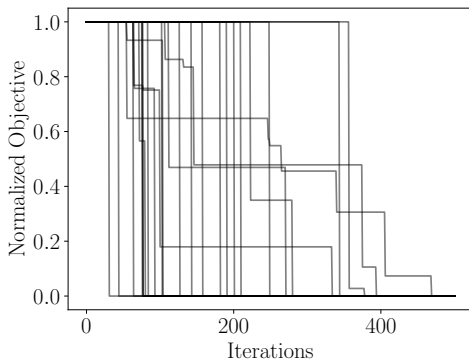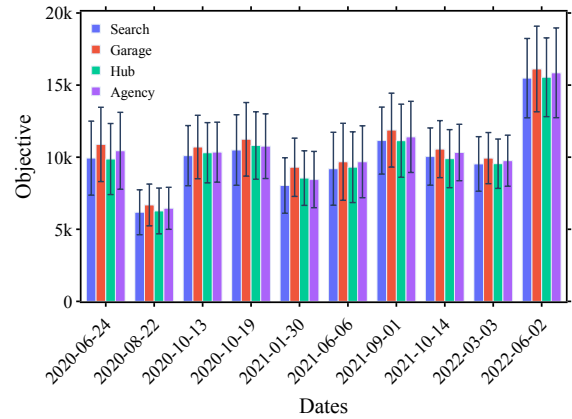

Figure 6: Comparison of different stationing plans across ten days. Lower is better.

Table 6: Average cost of stationing plans across several days

Stationing	Deadhead miles	Deadhead minutes	Passengers left
Agency	167	175	158
Hub	**163**	**170**	158
Garage	170	179	158
Search	167	175	**132**

Figure 5: Normalized objective costs for all test days.

trained. Figure 8 shows that the current stationing plan used by the transit agency adapts well to day-to-day trips, including overages and possible disruptions, compared to simply stationing them at the garage or without any stationing plans. Meanwhile, our proposed stationing plan outperforms all other plans. This shows that the stationing plans based on forecast data are still able to perform better than baseline stationing plans. Table 6 shows that across several days, our approach can reduce the number of people left behind by 26 while matching both deadhead miles and minutes used by the current stationing plan used by the transit agency. This shows that by combining disruption forecasts and introducing an efficient heuristic algorithm, we can improve transit reliability with almost no cost on the part of the agency.
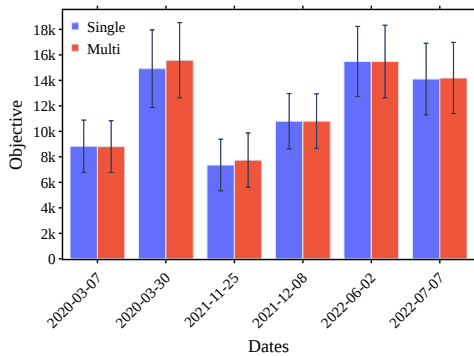
Figure 7: Comparison of daily optimization for six days and optimizing for all six days at once. Lower is better.
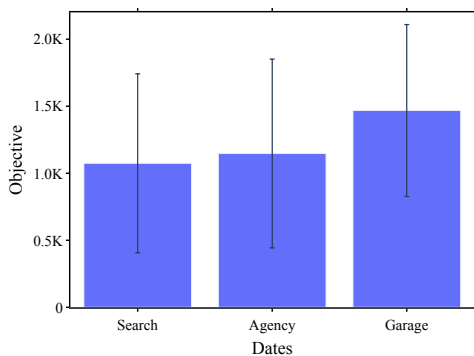

Figure 8: Using stationing plans from simulations using forecasted disruption and ridership on ground truth data. Lower is better.

6 RELATED WORK

Disruption forecasting Public transit disruption forecasting has been a relatively under-researched area. Traditionally, the focus has been on predicting traffic accidents, computing delay times, or estimating passenger flow when disruptions occur, rather than forecasting the disruptions themselves. This has proven challenging due to the unpredictability of disruptions and the scarcity of data concerning them [23]. However, recent studies have shown that non-parametric machine learning models can be utilized to analyze extensive datasets comprising numerous independent variables to forecast traffic accidents and disruptions with high accuracy [14, 26]. For instance, Yan et al. [26] employed the random forest model to forecast traffic accidents in the USA with high accuracy. Similarly, Parsa et al. [14] utilized XGBoost to predict traffic disruptions with high accuracy. The utilization of these non-parametric models provides a further advantage by elucidating the variables that play a crucial role in the prediction of disruptions [3]. Another study by Liu et al. [12] examines the potential of using automated collection data to comprehensively analyze unplanned disruption impacts. They propose a systematic approach to evaluate disruption impacts on system performance and individual responses in urban railway systems using automated fare collection (AFC) data.

Substitute Bus Services: Most of the work done in public transit using buses centers around headway control or bunching prevention. Bunching can be attributed to several factors such as traffic congestion or clustered passenger arrivals. Traditional solutions to this are schedule-based approaches and adaptive headway approaches [6]. Adaptive headway control introduces bus holding times at certain stops on a route to reduce headway gaps. Petit et al. shows how dynamic bus substitution strategies can be used to contain deviations in the schedule [16]. Zhang and Lo studied the optimal time to dispatch a substitute bus. They found that buses should only be dispatched if an event takes longer than a period of time [30].

Ridership Prediction: When considering route planning for public transit, agencies need to be aware of the ridership. Thus, this has been well studied [1]. The current state of the art uses nonparametric approaches that build upon the non-linear relationships between the input and output variables without any prior knowledge. Neural network models, particularly LSTM [25] are used to take advantage of temporal characteristics of travel demand while incorporating spatial information such as neighborhood and census tracts [8].

7 CONCLUSION

In this work, we presented a thorough strategy for solving the crucial problem of transit disruptions. We have created new ways for transportation agencies to proactively reduce delays and crowding by creating data-driven statistical models that precisely assess the possibility of disruptions. Our multifaceted methodology enabled the development of specific strategies that address both the macro and micro levels of transit service. Moreover, our approach minimized the number of passengers left behind by the strategic stationing of additional vehicles. In this way, the efficiency of transit operations can be increased, and the passenger experience can be greatly enhanced. The implications of this approach go beyond simple operational improvements. This work contributed to a more robust, responsive, and customer-focused public transportation system by setting the standard for predictive modeling for transit disruptions. Future research may look towards improving forecast performance even more, modifying it for other transit systems, and integrating it with real-world decision support settings. This attempt is a significant step towards a more dependable and effective public transit system, where disruptions are managed not just reactively but also in advance, making proactive mitigation crucial for contemporary urban mobility.

ACKNOWLEDGMENTS

REFERENCES

[1] Afiya Ayman, Juan Martinez, Philip Pugliese, Abhishek Dubey, and Aron Laszka. 2022. Neural Architecture and Feature Search for Predicting the Ridership of Public Transportation Routes. In *Proceedings of the 8th IEEE International Conference on Smart Computing (SMARTCOMP)*. 56–61.

[2] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

[3] Mu-Ming Chen and Mu-Chen Chen. 2020. Modeling road accident severity with comparisons of logistic regression, decision tree and random forest. *Information* 11, 5 (2020), 270.

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *CoRR* abs/1603.02754 (2016). arXiv:1603.02754 http://arxiv.org/abs/1603.02754

[5] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 785–794.

[6] Carlos F. Daganzo. 2009. A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. *Transportation Research Part B: Methodological* 43, 10 (2009), 913–921. https://doi.org/10.1016/j.trb.2009.04.002

[7] Carlos F. Daganzo. 2010. Structure of competitive transit networks. *Transportation Research Part B: Methodological* 44, 4 (2010), 434–446. https://doi.org/10.1016/j.trb.2009.11.001

[8] Jennifer Dill, Marc Schlossberg, Liang Ma, and Cody Meyer. 2013. Predicting Transit Ridership at Stop Level: Role of Service and Urban Form. https://api.semanticscholar.org/CorpusID:127314644

[9] Nadjla Ghaemi, Aurelius A Zilko, Fei Yan, Oded Cats, Dorota Kurowicka, and Rob MP Goverde. 2018. Impact of railway disruption predictions and rescheduling on passenger delays. *Journal of Rail Transport Planning & Management* 8, 2 (2018), 103–122.

[10] Joseph M Hilbe. 2014. *Modeling count data*. Cambridge University Press.

[11] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons Learned from the Chameleon Testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association.

[12] Luyu Liu, Harvey J Miller, and Jonathan Scheff. 2020. The impacts of COVID-19 pandemic on public transit demand in the United States. *PLOS One* 15, 11 (2020), e0242476.

[13] Bibiana McHugh. 2013. Pioneering Open Data Standards: The GTFS Story. In *Beyond Transparency: Open Data and the Future of Civic Innovation*. Code for America Press, Chapter 10, 125–135.

[14] Amir Bahador Parsa, Ali Movahedi, Homa Taghipour, Sybil Derrible, and Abolfazl (Kouros) Mohammadian. 2019. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accident Analysis & Prevention* 136 (2019), 105405. https://api.semanticscholar.org/CorpusID:209447626

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[16] Antoine Petit, Chao Lei, and Yanfeng Ouyang. 2019. Multiline Bus Bunching Control via Vehicle Substitution. *Transportation Research Part B: Methodological* 126 (2019), 68–86. https://doi.org/10.1016/j.trb.2019.05.009

[17] Kousik Rajesh, Eklavya Jain, and Prakash Kotecha. 2022. A Multi-Objective approach to the Electric Vehicle Routing Problem. *arXiv preprint arXiv:2208.12440* (2022).

[18] Jean-Paul Rodrigue. 2020. *The Geography of Transport Systems* (5th ed.). Routledge, London.

[19] Telmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. 2023. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning* 112 (2023), 3211–3260.

[20] Huijun Sun, Jianjun Wu, Lijuan Wu, Xiaoyong Yan, and Ziyou Gao. 2016. Estimating the influence of common disruptions on urban rail transit networks. *Transportation Research Part A: Policy and Practice* 94 (2016), 62–75.

[21] Jose Paolo Talusan, Ayan Mukhopadhyay, Dan Freudberg, and Abhishek Dubey. 2022. On Designing Day Ahead and Same Day Ridership Level Prediction Models for City-Scale Transit Networks Using Noisy APC Data. In *Proceedings of the 2022 IEEE International Conference on Big Data (Big Data)*. IEEE Computer Society, Los Alamitos, CA, USA, 5598–5606. https://doi.org/10.1109/BigData55660.2022.10020390

[22] Peter J. M. van Laarhoven and Emile H. L. Aarts. 1987. *Simulated annealing*. Springer Netherlands, Dordrecht, 7–15. https://doi.org/10.1007/978-94-015-7744-1_2

[23] Sayyed Mohsen Vazirizade, Ayan Mukhopadhyay, Geoffrey Pettet, Said El Said, Hiba Baroud, and Abhishek Dubey. 2021. Learning incident prediction models over large geographical areas for emergency response. In *Proceedings of the 2021 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 424–429.

[24] Xinyuan Wang, Jian Li, and Rongjie Yu. 2023. Modeling disruption durations of subway service via random survival forests: The case of Shanghai. *Journal of Transportation Safety & Security* 15, 2 (2023), 215–237.

[25] T. Yamamura, I. Arai, M. Kakiuchi, A. Endo, and K. Fujikawa. 2023. Bus Ridership Prediction with Time Section, Weather, and Ridership Trend Aware Multiple LSTM. In *Proceedings of the 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE Computer Society, Los Alamitos, CA, USA, 509–514. https://doi.org/10.1109/PerComWorkshops56833.2023.10150218

[26] Miaomiao Yan and Yindong Shen. 2022. Traffic accident severity prediction based on random forest. *Sustainability* 14, 3 (2022), 1729.

[27] Menno Yap. 2020. *Measuring, predicting and controlling disruption impacts for urban public transport*. Ph.D. Dissertation. TRAIL Research School.

[28] Menno Yap and Oded Cats. 2019. Analysis and prediction of disruptions in metro networks. In *Proceedings of the 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 1–7.

[29] Menno Yap and Oded Cats. 2021. Predicting disruptions and their passenger delay impacts for public transport stops. *Transportation* 48 (2021), 1703–1731.

[30] Shuyang Zhang and Hong K. Lo. 2018. Metro disruption management: Optimal initiation time of substitute bus services under uncertain system recovery time. *Transportation Research Part C: Emerging Technologies* 97 (Dec. 2018), 409–427. https://doi.org/10.1016/j.trc.2018.11.001

[31] Xuemei Zhou, Guohui Wei, Yunbo Zhang, Qianlin Wang, and Huanwu Guo. 2023. Optimizing Multi-Vehicle Demand-Responsive Bus Dispatching: A Real-Time Reservation-Based Approach. *Sustainability* 15, 7 (2023), 5909.