Tiako P.F. (ed) *Smart and Sustainable Applications*. Chronicle of Computing. OkIP
 https://doi.org/10.55432/978-1-6692-0005-5_1

# Predictive analytics-based evaluation of performance of public bus transportation San Antonio, Texas as a case study

Izzat Alsmadi[a*] and Mohammad Al-Ramahi[b]

*Department of Computing Science, Texas A&M University, San Antonio, USA*
*izzat.alsmadi@tamusa.edu*
*[b] Department of Accounting and Finance, Texas A&M University, San Antonio, USA*
*mrahman1@tamusa.edu*

*ABSTRACT*

*Citizens in large cities utilize public transportation as an alternative to self-driving for several reasons, such as avoiding traffic congestion and parking costs and utilizing their time for other things (e.g. reading a book or responding to emails). While large cities provide public transportation as a service to their citizens, they need to consider optimizing their budget and ensuring that public transportation is available and reliable. Using our case study, the public bus transit system in the city of San Antonio, Texas, in this paper, we used predictive analytics models to evaluate the performance of public bus transportation. We used time point stops as the target variable in order to evaluate their impact on the overall performance of the system. We also evaluated methods for the detection of potential bus-time savings and reported several examples of possible savings.*

*Keywords: Predictive analytics, GTFS, Transportation Intelligence*

## I. INTRODUCTION

Public transportation generates a large amount of data that can be continuously analyzed to better improve public transportation services. General Transit Feed Specification (GTFS) provides a standard protocol to effectively transmit real-time transit information. The data that is described in GTFS feeds is not sensitive or proprietary as it is information about public services. As GTFS data is public, this allows many users and researchers to develop tools and applications to utilize this data.

One of the most commonly emphasized criteria in public transportation is travel time reliability. This can be quantified through several metrics, such as expected waiting time, variance of travel time, or on-time performance (Danaher et al., 2020). Nevertheless, when it comes to determining the most optimal routes in public transportation, simply considering travel duration is often inadequate. Other factors like the number of transfers involved and the expenses incurred can be equally significant.

Transportation networks are usually modelled with graph structures for their intuitiveness and ability to utilize many software tools. Additionally, algorithms such as Dijkstra's algorithm can work efficiently to solve the single-source shortest-paths problem. In public transportation, many publications utilized Python libraries (e.g. Partridge, Peartree (Butts, 2021) and NetworkX) to convert GTFS feeds into directed network graphs. The graph contains two main elements (Madamori, Max-Onakpoya, Erhardt, & Baker, 2021):

- Nodes that represent bus stops. Each node has several edges that represent the departure times for all buses from that stop.
- Edges. Each edge represents a bus path from one stop to another. The edge weight is the average time it takes for a bus to get between the two stops on the edge.

Bus transportation networks can be modelled as a graph where bus stops represent the nodes and connections between those stops represent the edges. The edge costs are the costs of shortest paths between the respective nodes in the original graph. The edge cost is computed as the shortest path cost for each departure time from a source node to a destination node (Tesfaye, Augsten, Pawlik, Böhlen, & Jensen, 2022). In the time-dependent model, all nodes of the graph represent a bus stop linked together by one or more routes. A mathematical function containing a time variable defines the weight of every edge. Each query evaluates the weight according to the time of the query. In the time-expanded model, all nodes represent an event (arrival, departure or transfer), and thus, it requires more nodes and edges (Fortin, Morency, & Trépanier, 2016). In the next sub-section, we discuss time points in public transportation as designated stops used in schedule management.

### 1.1. Time Points

A time point is a public transit stop that a vehicle tries to reach at a scheduled time. A vehicle is not supposed to pass a timepoint until the scheduled time has arrived. These stops are contrasted with all other stops, besides timepoint stops, on a scheduled route for which the transit agency does not explicitly schedule an arrival/departure time. Beyond time points, drivers can have the flexibility to arrive at other stops and accommodate real-time situations related to delays in traffic or any other irregular circumstances. Figure 1 shows the variation in the number of time points in the different trips.

* Corresponding author Email: izzat.alsmadi@tamusa.edu



Presented at the 2023 OkIP International Conference
on Automated and Intelligent Systems (CAIS)
in Oklahoma City, OK, USA, and Online, on October 2-5, 2023

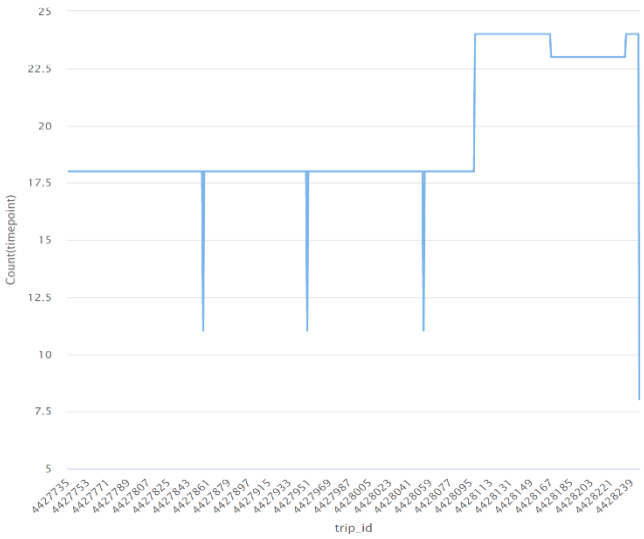


Fig. 1. Number of timepoints in San Antonio bus trips

## II. RELATED WORK

### A. The General Transit Feed Specification (GTFS)

The GTFS defines a common format for public transportation schedules and associated geographic information. GTFS is used as a format to allow public transportation authorities to exchange their data in a common format. A typical GTFS feed contains a collection of text files that describe static public transit schedules and related geodata.

Delays or uncertainty in traffic occur due to the complexity of the system and the real-time or dynamic factors related to accidents that can impact and cause issues. Lee and Miller's (2020) paper focused on evaluating or measuring accessibility to deal with travel time uncertainty. The traffic routing problem is formulated as a multi-objective optimization problem with time and reliability of the router as the two main objectives. The approach is based on a fast, non-dominated sorting algorithm (FNSA) that is adapted from Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002).

In 2008/2009, a local transit agency in Albany, NY, provided passenger data, which was used in the research paper (Zhang, 2014). The authors of the paper analyzed the data based on different times and weather conditions. In addition, they conducted extensive research on GTFS data and created a website that used Google Maps API to retrieve station distance and topography data. After testing various algorithms, the authors found that the EM algorithm and K-Means were the most effective for clustering stations. Although the machine learning strategy was able to comprehensively evaluate all stops, it was inadequate for analyzing specific routes. Therefore, the author used K-Means to create a BRT station selection tool which could cluster stops on a particular route.

### B. Timepoints and Schedule Adherence

Time points are gateways selected to manage scheduling in transportation systems. Several research papers discussed different scheduling algorithms to optimize timepoint scheduling (e.g. (Liu & Miller, 2020; Sun, Samal, White, & Dubey, 2017) (Glick, 2020; Sun, Dubey, White, & Gokhale, 2019)). In terms of scheduling and time points, drivers are expected to (Sun et al., 2017):

- Wait at the time point until the scheduled time if it arrives early.
- Departure as soon as possible if they arrive on time or late.

### C. Travel Time Savings

Arias et al. (2021) looked at creating a bus-only lane in Atlanta and how it can result in travel time savings. They went through the map of Atlanta to plan out a specific highway to have a bus-only lane. Using GTFS, they looked at the bus travel time that can be improved. They used stop times to calculate the improved time by looking at best-case scenarios versus worst-case scenarios.

In (Rothfeld, Fu, Balać, & Antoniou, 2021), authors presented an exploratory study in urban air mobility travel time saving for several cities, namely, Munich, Paris and San Francisco. Arias et al. (2021) used GTFS to evaluate methods for travel time-saving potentials. The study used the 2018 Metropolitan Atlanta Rapid Transit Authority (MARTA) bus network.

### D. Public Transit Routing

The issue of finding optimal routes in public transportation systems was examined by researchers. To tackle this problem, a popular method involves representing the network as a graph and applying a shortest-path algorithm to it (Rothfeld et al., 2021).

Delling, Pajor, and Werneck (2015) have presented a new approach called RAPTOR, which stands for Round-Based Public Transit Optimized Router. This method aims to determine the best possible journeys between two specified stops while minimizing both the travel time and the number of transfers needed. Unlike previous methods that rely on Dijkstra's algorithm, RAPTOR functions using a ground-based approach. This involves computing arrival times for each round by traversing each route, such as a bus line, no more than once. The algorithm is based on a dynamic program and utilizes uncomplicated data structures, resulting in efficient memory usage.

Jeon, Nam, and Jun (2018) have introduced an enhanced version of the RAPTOR algorithm, which accounts for transfer resistance and multi-path searching. They have incorporated transfer resistance during transfers and assigned distinct values based on the type of transit mode used. By examining the algorithm's output before and after modification and comparing it with the routes taken by passengers in reality, the authors of the study have demonstrated that the proposed algorithm considers the diverse route selection criteria of passengers.

## III. VIA METROPOLITAN TRANSIT

In this section, we present a summary of the VIA Metropolitan Transit bus public transit system in the city of San Antonio, Texas.

*A. Network Properties*

The following statistics reflect the current status of the San Antonio VIA bus transportation system based on public information extracted from GTFS:

- Average riders waiting time in San Antonio VIA is reported as 10 minutes (Selcraig, 2020)
- The overall number of routes is 98.
- The overall number of stops is 6127.
- The average number of trips per day is 4795.
- The number of shapes is 127258. Shapes are associated with trips and consist of a sequence of points (i.e., the geographic paths) through which the vehicle passes in order.
- The number of edges is 38590. Edges are characterized by the straight-line distance between stops.
- Stop times: 565173: Stop times represent the arrival and departure times of a trip at a stop.
- Timepoint stops: In the dataset, the total number of stops for all trips per day is 395588. Of those stops, 45448 stops are time points.

Additionally, there are several dynamic attributes of VIA that are specific to the data that we have collected and used.

- The total number of evaluated trips in our experiments is 123457.
- The number of transfers: 7314. This number is very dynamic and depends on actual trips and whether the traveller needs to switch from one bus to another.
- The number of days in the collected dataset is 140 days in 2022.
- The busiest day is 2022-10-28.
- The average edge cost, which refers to the business time between the two stops in the edge, is 53 seconds.
- Maximum edge cost is 1231 seconds.
- Average stop waiting time, outbound: 20 seconds, inbound: 20 seconds.
- Maximum stop waiting time, outbound: 71 seconds, inbound: 74 seconds. Those waiting times are for bus drivers, not riders.
- The number of gateways is 34. We used the definition and algorithm described in (Madamori et al., 2021) to create gateway stops. Gateways are special stops selected to act as hubs for other stops to collect and store network information. In a smart network, such gateways can be used to make real-time decisions to optimize network usage and resources.

Figure 2 shows a network graph for San Antonio busses network based on edges and nodes described earlier.

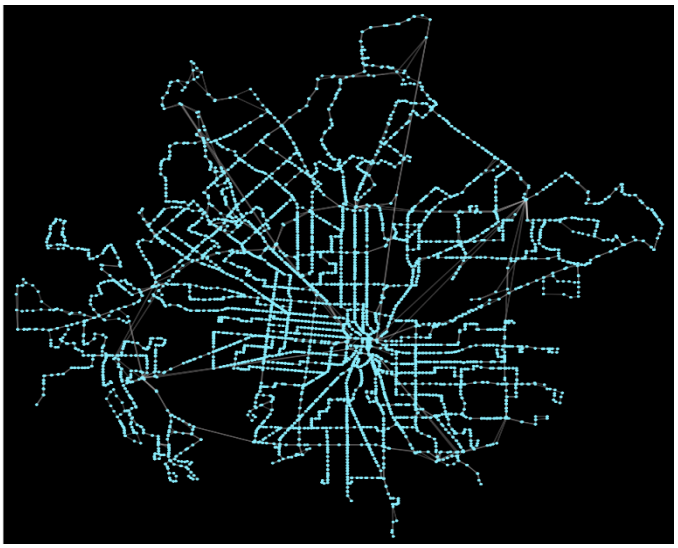

Fig. 2. San Antonio VIA Buses Network, all routes

We created Figure 2 graph object from GTFS data using NetworkX graphs. Because the graph object is formatted as an instantiated NetworkX graph, we can perform all typical network algorithms that are built into NetworkX. For example, we generated betweenness centrality.

Figure 3 shows the longitude and latitude distributions of VIA. While there is a fair distribution of stops across several latitudes, they are more condensed in a few longitudes that have higher volumes of trips. The figure shows in particular that more than 35,000 trips are reported in the longitude (-98.45 to -98.55).

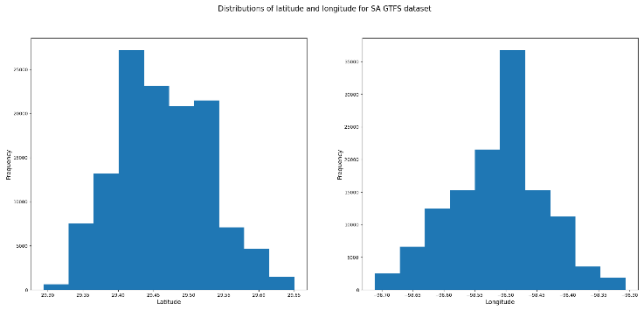


Fig. 3. Distributions of latitude and longitude for SA GTFS dataset

Betweenness centrality provides a measure for the relative importance of a node in the bus network on the basis of the fraction of shortest paths that go through this node. A high betweenness centrality can be an indication that a node is essential in connecting different parts of a network.

Betweenness centrality of a node (or a bus stop in our case) v is the sum of the fraction of all-pairs shortest paths that pass through v. For around 6000 bus stops in SA, Betweenness centrality, minimum, maximum and average values are: 0, 0.2294, and 0.01163 respectively. In order for one stop to have a high betweenness centrality, the node or the bus stop must be between many of the other nodes. The difference between minimum and maximum shows the large variation in this value from one stop to another. Some of the reported betweenness centrality in literature for other cities are higher than those of

San Antonio. For example, (Akse, 2014) (London Euston (0.403)), and (Badie Modiri, 2018) (Katz (0.34)). Other major cities, such as Mexico ,City is reported to have lower betweenness centrality (Reyna, de la Mota, & Vázquez, 2021)(0.1448) and 0.04896 in Chapel Hill Transit (Madamori et al., 2021). Table 1 shows the top 10 stops in San Antonio in terms of betweenness centrality.

TABLE I. TOP 10 STOPS IN SAN ANTONIO IN BETWEENNESS CENTRALITY

| StopID | BC Value | StopID | BC Value |
|---|---|---|---|
| 76773 | 0.2294 | 99496 | 0.16999 |
| 71839 | 0.2136 | 81726 | 0.1699 |
| 55229 | 0.2056 | 30049 | 0.1624 |
| 70996 | 0.2035 | 71926 | 0.1606 |
| 70997 | 0.1706 | 88986 | 0.1595 |

### *B. Timepoint classification analysis*

While it's not clear how public transportation selects or nominate some stops to be time points and whether those time points are dynamic or flexible to change, our analysis uses them as target label. We collected, created and aggregated several features based on GTFS data. Each row represents a stop in a trip, and the target column is a binary target, the timepoint, whether this stop is a timepoint (1) or not (0). The following features are used as input features to the classification model:

• TripID
• DepartureTimeRelative: Each stop will have a departure and arrival time. In normal scenarios, for time points, those times are the same. Hence, one can be used. Additionally, in order to process departure time in machine learning models, we converted it to a relative real number between 0 and 1. For example, 08:00:00 am is converted to 0.3, 12:00:00 as 0.5.
• HourDeparture: This is a categorized column of the departure/arrival time to show only the hour part.
• StopID
• Timepoint: This is the target feature, 1 if the stop is a timepoint and zero if not.
• StopSequence: This refers to the stop order in its trip.
• PickupType: The pickup type field indicates whether passengers are picked up at a stop as part of the normal schedule or whether a pickup at the stop is not available. This field allows the transit agency to indicate that passengers must call the agency to arrange a pickup at a particular stop. Out of 565127 stop records in the dataset, only 10392 are 1, not a pick-up stop.
• DropOffType: Indicates whether passengers are dropped off at a stop as part of the normal schedule or whether a drop off at the stop is not available. This field also allows the transit agency to indicate that passengers must call to arrange a drop-off at a particular stop. The feature is very similar to PickupType and so can be eliminated.
• RouteInt: Routes are defined in the file routes.txt. They are made up of one or more trips. A trip occurs at a specific time, so the route is time-independent.

We evaluate several classical algorithms, namely: Logistic Regression, KNN, Decision Tree, Random Forest and Gaussian NB. We reported, in Table 2, several performance metrics.

TABLE II. PERFORMANCE METRICS ON DIFFERENT CLASSIFICATION MODELS

| Classifier | Accuracy | Precision | Recall & F1 | Classifier |
|---|---|---|---|---|
| LogRegression | 0.90 | 0.0 | 0.0 | 0.0 |
| KNeighbors | 0.90 | 0.0 | 0.24 | 0.01 |
| DecisionTree | 0.99 | 0.99 | 0.99 | 0.99 |
| RandomForest | 0.99 | 0.99 | 0.99 | 0.99 |
| GaussianNB | 0.90 | 0.0 | 0.0 | 0.0 |

Results in Table 2 show two classifiers, Random Forest and Decision Trees, performed well in all metrics. However, the rest of the classifiers, LR, KNN, and NB, showed very low values in precision, recall and F1. This is expected as the dataset is imbalanced; in the dataset we used, 65,000 stops that are time points, while the rest (500127) are not. In the second experiment, we sampled from the dataset equal samples for labels 0 and 1. As shown in Table 3, the under-sampling improves precision, recall and F1 for KNN and NB but significantly lowered accuracy values.

TABLE III. PERFORMANCE METRICS ON DIFFERENT CLASSIFICATION MODELS AFTER UNDERSAMPLING

| Classifier | Accuracy | Precision | Recall & F1 | Classifier |
|---|---|---|---|---|
| LogRegression | 0.55 | 0.0 | 0.0 | 0.0 |
| KNeighbors | 0.48 | 0.31 | 0.40 | 0.35 |
| DecisionTree | 0.99 | 0.99 | 0.99 | 0.99 |
| RandomForest | 0.99 | 0.99 | 0.99 | 0.99 |
| GaussianNB | 0.55 | 0.20 | 0.49 | 0.28 |

Features weights with target columns and timepoints (Figure 4) show that pickup-type, then n_stops are the most important features. Although the total number of stops that have pickup-type =1 is small (10392/565127), and also the number of stops that have timepoint value=1 (65000/565127), the correlation is shown to be high between both.

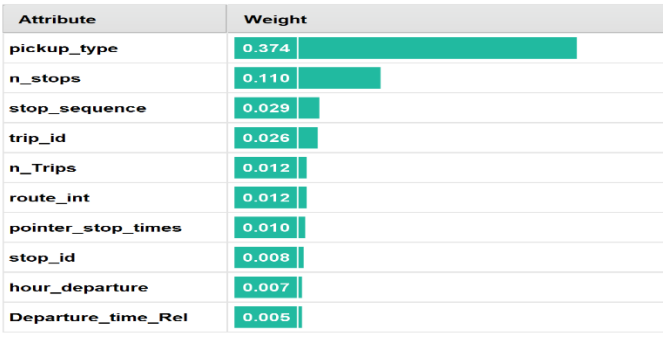


Fig. 4. Top Features and Weights

Figure 5 shows a Decision Tree (DT) based on target class timepoint. The root feature is stop-sequence. A stop sequence is a unique sequence of stops visited by a transit trip. It first reads that all stops that are the first in their trip are timepoint stops. If

there is no first stop and there are no pick-up stops, then they will not be timepoint stops. Those are the most readable edges in the DT.

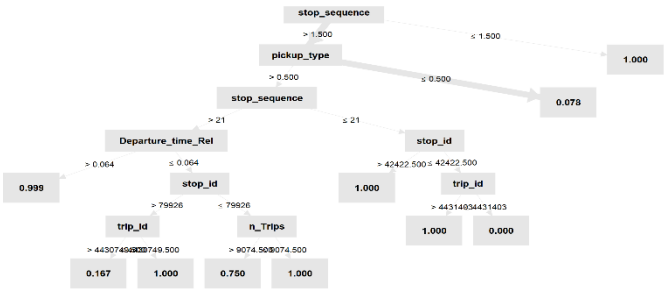


Fig. 5. Timepoint Decision Tree

## C. *Time Savings*

We used GTFS data to analyze the time savings for each route. The idea is that if the times change based on traffic, smaller wait times during less traffic and higher wait times based on heavy traffic, then if we had a dedicated lane for buses, then the heavy traffic times would be eliminated and leave the minimum wait time at each stop. In order to calculate the benefit, we take the average wait time of all stops and subtract the minimum time to give us a time savings. By adding all of the time savings of each route, we can get an idea of the time benefit of having a dedicated lane for that route. For example, to calculate the time savings for route 100, we followed the following process:

- First, we calculate the difference between the arrival times of each stop and the next one in a trip sequence of stops using the equation waitSeconds = (firstArrivalTimes - secondArrivalTimes). To do that, we used the equation firstArrivalTimes = directionTimes.arrivaltime[1:].values to retrieve all stops in a trip sequence from the index 1 (inclusive) to the end of the sequence (i.e., all stops from the second to the last), for example,(26160, 26940, 27960, 28800, 29760, 30660, 31380, 32340, 33240, 34140,35040, 35940). We then used the equation secondArrivalTime= directiontimes.arrivaltime[:-1].values to get all stops from the beginning of the trip sequence up to, but excluding, the last stop, for example, (25320, 26160, 26940, 27960, 28800, 29760, 30660, 31380, 32340, 33240, 34140, 35040)
- Second, we calculated the average of all differences using the equation average wait = np.array(waitseconds).mean()
- We then computed the "Time savings" by subtracting the minimum waiting time from the average wait for that trip using the equation Time savings = average wait – minimum

Figure 6 shows there were 12 first and second stops in a trip on route 100. Their waiting times range from 720 to 1020 seconds. The time savings calculated for this trip is 165 seconds. That means if all waiting times were 720, there would be a 165-second benefit compared to the original schedule.

```
Processing on route 100.
       Reduced trips in consideration from 135 to 31.
Route
['100' '100' '100' '100' '100' '100' '100' '100' '100' '100' '100' '100']
wait seconds
[ 840.  780. 1020.  840.  960.  900.  720.  960.  900.  900.  900.  900.]
Wait minimum
720.0
average wait
885.0
time savings = 885.0  -  720.0
165.0
------------------------
```

Fig. 6. Time savings for a trip on Route 100

We appended each iteration of the route to get the list of all time savings for each trip. For route 100, there were 107 trips, as shown in figure 7. We computed the average time savings for all trips to get 135.72 seconds. This was done with a time constraint of 700 a.m. to 10 a.m. only.

```
Processing on route 100.
       Reduced trips in consideration from 135 to 107.
      Time Savings
Route
100      135.729007
```

Fig. 7. Number of trips in route 100

Table 4 shows the statistics of the time saving for all routes. The table provides statistics for time delay and late arrival for a set of 78 routes. The statistics are summarized as follows:

- Number of Routes: There are a total of 78 routes included in the analysis.
- Mean: The mean (average) time delay across all routes is approximately 45 minutes and 4.42 seconds.
- Maximum: The maximum time delay observed among the routes is 6 hours, 35 minutes, and 22.97 seconds.
- Minimum: The minimum time delay observed among the routes is 0 hours, 0 minutes, and 0 seconds, indicating that some routes arrive exactly on time.
- Standard Deviation: The standard deviation provides a measure of the variability or dispersion of the time delays across the routes. The standard deviation value is not provided in the given information.

TABLE IV. TIME SAVINGS FOR ALL ROUTES

| Num of routes | Mean | Max | Min | Std |
|---|---|---|---|---|
| 78 | 0:45:04.42 | 6:35:22.97 | 0:00:00 | 01:37:15 |

These statistics offer insights into the average, maximum, and minimum time delays experienced by passengers across the set of 78 routes. It indicates the range of delays observed and provides an overview of the distribution of delays.

Table 5 shows the top routes with the highest average time savings (i.e., average late arrival time to the destination stop point), along with corresponding trip names. The "Time savings" column indicates the average delay in arrival time experienced by passengers for each specific route, which could

be saved by having a dedicated lane for buses to avoid the traffic that causes the delay. The values are presented in hours, minutes, and seconds.

Analyzing this data can help identify routes with the highest average delays, allowing for potential improvements in scheduling, operational efficiency, and passenger communication.

TABLE V. TOP ROUTES WITH THE HIGHEST AVERAGE TIME SAVINGS

| Route | Trips in Route | Time savings |
|---|---|---|
| 42 | 4432390, 4432363 | 6:35:22 |
| 515 | 4433667, 4433654 | 5:25:06 |
| 25 | 4430451, 4430507 | 5:11:11 |
| 2 | 4429320, 4429319, 4429273 | 5:09:16 |
| 97 | 4440015, 4440032, 4440018 | 4:46:49 |
| 36 | 4431980, 4431995, 4431979 | 4:45:50 |
| 9 | 4439129, 4439074 | 4:39:52 |
| 88 | 4438827, 4438781,4438828,4438780 | 4:21:39 |
| 28 | 4430989, 4430940, 4430939 | 3:38:08 |
| 34 | 4431847,4431804,4431846,4431803 | 3:33:54 |
| 75 | 4437746, 4437698, 4437747 | 2:02:45 |
| 632 | 4436607, 4436571, 4436564 | 1:13:52 |

## IV. CONCLUSION AND FUTURE WORK

Machine learning approaches are proposed to improve performance and many aspects of public transportation systems. The evaluation of historical and real-time data public transportation data can help make better and more informative planning and decisions. Timepoint stops are used as a scheduling tool to track performance. Using timepoint stops as a target column, in this paper, we integrated several input features from the different GTFS tables. We reported results from several classifiers, while Random Forest and Decision Trees showed the overall best results. We also conducted results to evaluate approaches to save time in the different bus routes. We showed that significant time can be saved while maintaining system requirements or constraints. Looking ahead, several avenues for future research and development in this field can be identified. First, the integration of real-time data streams into the predictive models could offer a more dynamic understanding of system performance and allow for on-the-fly adjustments. Second, incorporating user-centric factors such as passenger preferences, peak travel times, and route popularity could further enhance the accuracy and applicability of the predictive models.

## ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 2131193. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

Akse, F. A. (2014). Aggregate waiting time reduction on public transportation networks.

Arias, D., Todd, K., Krieger, J., Maddox, S., Haley, P., Watkins, K. E., & Berrebi, S. (2021). Using gtfs to calculate travel time savings potential of bus preferential treatments. Transportation Research Record, 2675(9), 1643-1654.

Badie Modiri, A. (2018). Error and attack tolerance of public transportation networks: a temporal networks approach.

Butts, k. K. (2021). Retrieved from https://github.com/kuanb/peartree

Danaher, A., Wensley, J., Dunham, A., Orosz, T., Avery, R., Cobb, K., . . . Connor, M. (2020). Minutes Matter: A Bus Transit Service Reliability Guidebook.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.

Delling, D., Pajor, T., & Werneck, R. F. (2015). Round-based public transit routing. Transportation Science, 49(3), 591-604.

Fortin, P., Morency, C., & Trépanier, M. (2016). Innovative GTFS data application for transit network analysis using a graph-oriented method. Journal of Public Transportation, 19(4), 18-37.

Glick, T. B. (2020). Methodologies to Quantify Transit Performance Metrics at the System-Level Using High-Resolution GPS, Stop-Level, and GTFS Archived Transit Data. Portland State University,

Jeon, I., Nam, H., & Jun, C. (2018). A schedule-based public transit routing algorithm for finding K-shortest paths considering transfer penalties. The Journal of The Korea Institute of Intelligent Transport Systems, 17(3), 72-86.

Lee, J., & Miller, H. J. (2020). Robust accessibility: Measuring accessibility based on travelers' heterogeneous strategies for managing travel time uncertainty. Journal of Transport Geography, 86, 102747.

Liu, L., & Miller, H. J. (2020). Does real-time transit information reduce waiting time? An empirical analysis. Transportation Research Part A: Policy and Practice, 141, 167-179.

Madamori, O., Max-Onakpoya, E., Erhardt, G. D., & Baker, C. E. (2021). Enabling opportunistic low-cost smart cities by using tactical edge node placement. Paper presented at the 2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS).

Reyna, O. S. S., de la Mota, I. F., & Vázquez, K. R. (2021). Complex networks analysis: Mexico's city metro system during the pandemic of COVID-19. Case Studies on Transport Policy, 9(4), 1459-1466.

Rothfeld, R., Fu, M., Balać, M., & Antoniou, C. (2021). Potential urban air mobility travel time savings: An exploratory analysis of Munich, Paris, and San Francisco. Sustainability, 13(4), 2217.

Selcraig, B. (2020). Early Success of San Antonio's VIA Link Prompts Citywide Expansion Plan. Retrieved from https://www.expressnews.com/news/local/article/Early-success-of-San-Antonio-s-VIA-Link-prompts-15075197.php

Sun, F., Dubey, A., White, J., & Gokhale, A. (2019). Transit-hub: A smart public transportation decision support system with multi-timescale analytical services. Cluster Computing, 22, 2239-2254.

Sun, F., Samal, C., White, J., & Dubey, A. (2017). Unsupervised mechanisms for optimizing on-time performance of fixed schedule transit vehicles. Paper presented at the 2017 IEEE International Conference on Smart Computing (SMARTCOMP).

Tesfaye, B., Augsten, N., Pawlik, M., Böhlen, M. H., & Jensen, C. S. (2022). Speeding up reachability queries in public transport networks using graph partitioning. Information Systems Frontiers, 24(1), 11-29.

Zhang, T. (2014). Bus stop usage evaluation and BRT station selection strategy by machine learning methods: State University of New York at Albany.