Navigating Widespread Urban Transit Dynamics with Standardized Data and Scalable Models

Zackary Aemmer

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Don MacKenzie, Chair
Andisheh Ranjbari
David Shean

Program Authorized to Offer Degree:
Civil and Environmental Engineering

University of Washington

Abstract

Navigating Widespread Urban Transit Dynamics with Standardized Data and Scalable Models

Zackary Aemmer

Chair of the Supervisory Committee:
Don MacKenzie
Department of Civil and Environmental Engineering

Standardized and open source bus data including static schedules and realtime positions have become widely available in public web application programming interfaces. Though these data primarily underly popular mobile trip planning applications, they also enable new analyses in understanding, forecasting and improving bus operations across cities. Due to their lower resolution and simpler features, open data are more challenging to work with than those of the underlying sensors. However, their wide scale and standardization make them a valuable resource for researchers and planners. This work develops a set of tools for analyzing bus operations with open data. Central to this endeavor is the ongoing collection of a multi-year dataset from the King County Metro transit network in Seattle, Washington, approaching one billion tracked bus locations. First, basic roadway segment aggregation is used to visualize the spatiotemporal dynamics of different delays in the transit system. This is used to identify priority locations for transit priority treatments. Then a set of deep learning models are developed to forecast bus travel times under different data availability scenarios. Their generalizability is tested across different cities and transit networks. Finally, these models are used to estimate energy demands of a battery electric bus fleet for any city. Implications of the open data standards for energy modeling are examined, and a cross-sectional analysis reveals barriers to fleet electrification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I'd like to acknowledge and thank my committee for everything they've done to enable the learning of myself and other students. Each of them has provided invaluable inspiration and guidance to every part of this dissertation as both professors and mentors.

I'd also like to thank my advisor Don MacKenzie for the ongoing freedom to make mistakes and his considerable patience in helping me correct them. Every part of this work has benefitted immensely from his support. The Sustainable Transportation Lab has been a place of growth and connection for me, and much of this dissertation has been improved through the ideas and feedback of other members of the lab past and present. Thanks especially to Borna Arabkhedri for his collaboration on the original Transit Corridor Study, and to Eric Barber for his mentorship when I was an undergraduate researcher in the lab.

I'd also like to acknowledge and thank the Valle program and those who administer it for supporting Chapter 3 of this dissertation. I'd like to especially thank Massimiliano Ruocco, Sondre Sørbø and Alfredo Clemente whose input and guidance were absolutely essential in developing the generalizable models that Chapters 3 and 4 are built on. Thank you to the unforgettable friends and family in Trondheim who welcomed me into their lives.

Last I'd like to say thank you to my friends, without their distractions (and support) there is no chance I would have ever finished this dissertation. And above all else thank you to my family for their unending support and love. I am incredibly fortunate to have had them from the beginning.

Chapter 1
INTRODUCTION

1.1 Motivation

Transit systems decrease overall transportation sector emissions and improve population accessibility. In 2021, 63 billion passenger-miles were taken by bus or BRT (bus rapid transit) out of 176 billion total onboard transit, making buses the largest (36%) transit share for Americans [137]. Compared to light rail, subways and streetcars (5%), buses have similar per-mile operating costs (2.84$/passenger-mile compared to 2.51$) but significantly lower capital investment (0.63$/passenger-mile compared to 4.78$) [4, 3]. In the broader picture, this makes BRT or regular bus lines attractive in cities which have less capital funding available for transit, and an important feeder component of networks which either have existing rail, or are in the process of building out a rail network.

However, the shared infrastructure that makes bus transit an effective investment also exposes it to delays from congestion, shared curb-use, and other roadway users. In transit satisfaction surveys, service delivery metrics tend to take customer priority over measures related to comfort [114, 73]. This means that functional improvements such as short headways and higher travel speeds are more valuable to transit riders than auxiliary desires like clean vehicles or customer service. It is only once basic service needs have been met that riders begin to place satisfaction weight on secondary aspects of the service [10]. Thus, one of the best methods of drawing and maintaining ridership is to improve the consistency and efficiency of transit network operations. To accomplish this, planners and researchers strive for new methods of monitoring, evaluating and executing changes to bus infrastructure. This ultimately helps achieve the sustainability benefits of bus transit.

One rapidly proliferating field of development in transit operations planning and research is "big data" analyses collected from ITS (intelligent transportation system)s. These methods have exciting potential (and at times hype beyond their current capabilities) for improving transit delay and demand forecasting, automating customer service and managing operations. However, the emerging nature of these data means that nearly all tools are home-grown or constructed on a need-by-need basis, and often use closed data sources collected from proprietary software systems. This leads to the critical challenge of collaboration in this emerging field through shared knowledge, resources and findings. Fortunately, many agencies worldwide have rallied around standards for sharing transit data. Standards such as GTFS (General Transit Feed Specification) and NeTEx (Network Timetable Exchange) share "static" transit schedule and stop information, while "realtime" standards such as GTFS-RT (General Transit Feed Specification - Realtime) and SIRI (Service Interface for Realtime Information) share ongoing trip information. These standards are the foundation of popular transit trip planning tools such as Google Maps and OpenTripPlanner. They also provide opportunities for researchers and analysts to build tools and analyses that are generalizable between transit networks. This dissertation explores the potential of these open source data standards for building practical, collaborative tools for analyzing bus transit operations.

1.2 Standardized Bus Transit Data Formats

There are many sources of bus transit data and means of collecting them. Several existing works have proposed thorough taxonomies of these data [74, 120]. For the sake of completeness a summary of these sources is provided here before delving into the specifics of standardized bus data formats.

There are three primary, widely discussed transit data sources collected in most automated bus transit ITS systems. First, AFC (automatic fare collection) data is collected when a customer taps on or off a transit system. This provides trip start and end locations, which can be used to estimate OD (origin-destination) trip demand of the service. Some systems have fixed fares and do not require a tap-off, providing only trip origin information. One challenge with AFC data is understanding when a trip is a transfer, as opposed to a short stop at an intermediate destination [86]. Furthermore, cards may not use consistent identifiers out of concern for passenger privacy. This means that spatial boarding and alighting can be aggregated by location but individual traveler OD flows must be estimated separately [86].

Second, APC (automatic passenger count) data is collected onboard transit vehicles when passengers board or disembark, with doorway loop detectors or light sensors. This can provide a count of the number of passengers onboard a transit vehicle at any given time. This allows estimation of passenger-delay time due to vehicle slowdowns, as well as dwell time at individual stops among other metrics. Challenges include discerning between boarding and alighting movements, and again estimating OD information without unique passenger identifiers [74].

Last, AVL (automatic vehicle location) data is continuously collected by odometer or GPS sensors onboard a transit vehicle. This data can be used to estimate bus speeds, detect bunching, and provide travelers with arrival information among other uses. There are common challenges for GPS data, to which transit data is no exception. For example GPS-multipath errors in urban canyons throw off calculations and must be filtered or cleaned carefully, and data collected at constant frequency will have sampling bias towards locations with slower speeds [91]. Ultimately, AVL data and its derivatives can only directly inform performance in places and at times which vehicles currently or previously have been run. This is a crucial challenge in operational planning for new routes, schedule changes or otherwise predicting and forecasting bus movements in unobserved locations. Together, these three automated data collection sources can provide a fairly robust picture of transit operations including vehicle and passenger movements.

However, the systems which collect and monitor bus data are often based on proprietary hardware and software. This means that the sensor types, frequencies, data cleaning methods, performance dashboards and other aspects differ from system to system or even vehicle to vehicle [44]. Agencies may place separate contracts for hardware and monitoring software or separate AVL/AFC/APC systems. They may also incorporate in-house analyses and performance metrics, or rely on those developed by dashboard providers [144]. In other words, these systems can lack interoperability [133]. This effect has carried into bus operations research, where analyses are frequently based on single cities, routes or times from AVL/APC/AFC data as provided by a specific transit agency [147, 33, 158].

Standardized open formats for transit data can help solve this problem. Broadly, data standardization can address analysis obstacles related to metadata uncertainties (e.g., vehicle classes, network coverage), data transformation (e.g., trip/stop schedule structure, different sensor frequencies) and critical missing data (e.g., lack of coordinates, timestamps) [72]. Data standardization lowers the cost of switching providers for consumers (in this case transit agencies) increasing competition among providers and lowering costs, but on the other hand it can also stifle innovation and improvement of the standard itself, create high barrier-to-entry for all parties to re-tool for the standard, or provide market control to the dominant driver of the standard [72, 162, 67]. The former concerns are relatively low, given that transit data standards are essentially web API (application programming interface)s, which have low re-tooling costs and are adaptable to different existing hardware standards with appropriate metadata. However, the latter concerns may be valid given that the widely adopted GTFS standard is currently maintained by Google (in collaboration with industry partners), which has a vested interest in protecting its market for trip planning tools.

There are several public data sharing standards available for public transit data. They are typically broken into two components: 1) A static feed which includes all information about scheduled trips and 2) A realtime updated feed which includes all information about actual vehicle locations during ongoing trips. By far the most commonly adopted standard is the GTFS/GTFS-RT standard [82, 81]. A GTFS static feed is a zip archive containing comma separated value files for system trips, stops and other details. A GTFS-RT feed is a web API providing live updates for active trips. The primary competing static standard is NeTEx, which uses extensible markup language, and the primary competing realtime standard is SIRI [65, 166].

Advantages of NeTEx/SIRI are a neutral controlling agency, a format more complementary to web programming, and greater scope of information. Advantages of GTFS/GTFS-RT are significantly wider adoption, and a simpler and more understandable format for static data. GTFS/GTFS-RT is much more widely adopted in the US, while NeTEx/SIRI are generally EU standards. Many EU transit agencies provide both formats. Regardless of the chosen standard; their general purpose is to unify transit data such that it is interoperable with transit trip planning tools (e.g., OpenTripPlanner, Google Maps). The less-explored but emerging benefit is that planning-analysis and visualization tools can be created that work for any transit network [12, 128]. The primary challenge with these data formats is lack of personnel resources to manage and curate the data [181], and inconsistency across agencies following the standards [14]. The adoption and success of these standardized public transit data formats has also led to the emergence of similar standards for MaaS (mobility as a service) and other rideshare providers [129].

1.3 Emerging Tools and Analyses Built on Open Bus Transit Data

As standardized bus transit data sources have emerged, a variety of tools, webapps, converters and more have been developed that take advantage of or support these standards. Though some have associated research papers, many are developed independently by members of the community- perhaps more evidence towards the benefits of open and standardized data. Here, some of the different tool archetypes are explored through defining examples.

First, there are a wealth of map-based visualization tools for showing scheduled and realtime bus locations. The most comprehensive tools can show estimated positions from static data or actual positions where realtime information is available, and do so simultaneously across many different transit networks [17]. Most work however is demonstrated on a case study for one city [187, 80]. Others take their visualization a step further into analysis. For example estimating bus positions ([106]) or mining performance metrics from static schedule feeds alone [177]. Spatial performance information can also be explicitly extracted and visualized from vehicle travel times and flows in static scheduled stop arrival times [148]. Another approach is to use embedding plus clustering techniques with schedule data to compare latent variables between cities; creating visualizations of scheduled transit access and availability [84]. From a planning standpoint, it is useful to have tools which interface with APC/AFC data, which is not usually shared in standardized data feeds [76, 111, 107]. One study in this dissertation (Chapter 2) developed TransitVis, a transit performance visualization tool. It aggregates bus performance metrics and visualizes these across the network as a whole, rather than for individual ongoing bus trips.

Arrival time tools give immediate information to passengers on predicted waiting times at specific bus stops. OBA (OneBusAway) was one of the earliest, and explored the value of such tools in improving traveler experiences and perceptions of transit [173]. Without delving into individual modeling approaches, a sub-field has arisen around using open data to predict transit travel movements, though most do not contain a visualization or tool component [40, 53, 87]. For example some interface with the user through text messages [49]. TheTransitClock is an arrival time prediction model that interfaces with GTFS and GTFS-RT feeds to update times in bus schedules with realtime information, then provides that information to the public [45]. It has been deployed by transit agencies in several cities [45]. Regardless of the approach, information regarding the arrival time and certainty around that estimate can give even more control to the user and greatly increase transit satisfaction [93, 62].

Trip planning tools allow travelers to map out their trips before committing to them. This might include choosing ideal access and egress points for the transit system, finding transfer locations and times, or ultimately making decisions of whether to use transit and if so what type. OpenTripPlanner is an open source planner which uses a combination of static and realtime open data, along with travel time estimates for non-transit modes [140]. "R5R" is a routing library built on OSM (OpenStreetMap) and static transit data [150]. Transitr utilizes only realtime data in it's trip routing and time estimates [97], while Mapnificent uses only static open data to map travel time boundaries via public transit for any GTFS system [176]. Explore took the inverse approach to trip planning by allowing a user to search attractions, and receive many different transit routing options [175]. Most transit agencies provide their own trip planner (sometimes based on OpenTripPlanner or Google Maps) or are aggregated among other regional agencies in a single planner. GTFS-OSM-Sync interfaced OpenTripPlanner with OSM and open transit data feeds, creating a tool which handles trip planning and synchronizes transit network info with OSM [92]. Though not always open source, agency-maintained planners have the benefit of incorporating the most up-to-date data from the source bus network [175].

Last, tools have been developed to help data providers meet and maintain open data standards. One ongoing challenge is inconsistencies in application of the standard, whether due to error or vagueness in the definition. To address this a GTFS-RT validation tool was developed and tested across 78 feeds [14]. Findings from this study led to fundamental improvements in the GTFS-RT standard. The aforementioned GTFS-OSM-Sync also interfaces GTFS data with OSM, making agency stops/routes easier and more accurate to query [92]. Open source tools also exist to help create static and realtime feeds from underlying data [26]. Inversely, one can use open source tools to extract the transit network as a connected graph from a static feed [102]. Outside of academic research, countless other community tools exist on GitHub, with varying levels of upkeep. One compendium of such projects is awesome-transit [66].

1.4 Closed Source Bus Transit Design and Monitoring Tools

The service planning design process for bus transit service begins with network and stop layout, proceeds with timetabling, then handles vehicle and crew assignment [50]. Once in service, routes are monitored for measures of performance deemed important by the operator, and for providing realtime information to travelers. If certain routes, locations or times are performing poorly by chosen metrics, then service planning steps can be revisited and changes carried down the design chain [50]. Most software aids with one or more of these tasks:

- **Design Software:** Perform demand forecasting, route layout and scheduling.

- **Monitoring/Evaluation Software:** Maintain performance dashboards, GIS analyses, disruption detection.

- **Realtime Arrival Software:** Provide public-facing service alerts and trip updates.

- **Trip Planning Software:** Provide public-facing routing and timeschedule information.

Most of the aforementioned open source tools offer realtime arrival or trip planning, but few tackle design or monitoring/evaluation. This is where open data standards have potential to bring more consistent, collaborative analyses. Though the quantity of emerging open source transit analysis tools is high, existing tools may not meet the needs of transit planners and researchers for design and evalution tasks. Some tools require advanced programming knowledge to implement, others lack documentation, and most are prototypes. In order for open source tools to supplement or replace existing tools, they must offer clean and simple workflows for typical transit monitoring and evaluation tasks.

Several companies currently offer paid, all-in-one solutions for bus system design and evaluation. Remix is one such tool [170]. Though originally open source design-focused software, it has moved to closed source and added monitoring and evaluation functionality [64]. Tripspark offers similar functionality as well as a customer-facing arrival time and trip planning app [168]. PTV Visum focuses on network design and simulation, but also has a monitoring component (PTV Optima) [85]. Transloc offers monitoring software and a customer-facing arrival time apps [165]. Some notable design-only tools are TransCAD, Optibus, and tools from Esri built for ArcGIS [142, 54, 28]. Conveyal and Sugar both provide accessibility analysis tools that interface with OSM and GTFS data [43, 20].

Transit agencies and researchers have different resources and goals. For agencies, any combination of these or other paid tools can integrate with underlying ITS systems. They can provide comprehensive design, monitoring, realtime arrival and trip planning analyses specific to the system of interest. They ultimately offer simplicity as a service which can reduce personnel and technological ownership burdens. Regardless, some agencies have begun to adopt open source realtime arrival and trip planning software [45, 141]. Researchers and community members generally work with fewer resources and often rely on open source tools. They are also often interested in drawing comparisons across transit systems, for which standardized data and tools immensely reduce the analytical burden. For bus transit system design and evaluation these tools are in a nascent state. This could be in part due to limitations in data shared through the open standards. These data are generally lower resolution and have fewer features than those available in the underlying ITS collection systems [74, 14]. Understanding these limitations and improving the design/monitoring/evaluation tools constructed on these open standards stands to decrease the gap between research and practice, lower the barrier to transit analysis, foster collaboration across agencies, and improve the accessibility and sustainability benefits of public transit.

1.5 Outline of This Dissertation

This dissertation consists of three separate studies (in Chapters 2, 3 and 4). At the time of writing, Chapters 2 and 3 have been published as journal articles [5, 6]. These studies are synthesized under the theme of bus transit operations analysis supported by standardized, open data sources. Though they tackle different research questions, each study builds upon work in the previous chapter. They culminate in conclusions on practicality and limitations of open data standards for wide-scale bus transit analyses.

For the selective reader, the remainder of Chapter 1 contains a summary of each paper and its contribution to the greater whole of this dissertation. More detailed contribution statements can be found in Sections 2.3.4, 3.3.4 and 4.3.3. The abstracts of the completed studies are also included in Sections 2.1, 3.1 and 4.1.

Further literature reviews specific to the research questions of each individual study are found in Sections 2.3, 3.3 and 4.3. Following this chapter, Chapters 2, 3 and 4 detail the specific motivations, methods, results and findings for each study. Last, Chapter 5 summarizes the contributions of each study to the literature and draws unified conclusions on open data standards for wide-scale bus transit analysis.

- **Measurement and Classification of Transit Delays Using GTFS-RT Data** This study lays the groundwork for subsequent analyses using open source bus trip point data collected from public GTFS and GTFS-RT bus network feeds. It proposes a new perspective for aggregating individual bus observations to roadway segments, such that the system as a whole can be examined for inefficiencies, and places for targeted infrastructure improvements can be located. It also discusses the development of a prototype web tool which visualizes daily summaries of transit roadway segment performance. This study is the first to aggregate GTFS-RT data to roadway segments and classify delays in this way. It also identifies some key concerns related to the reported features in realtime data.

- **Generalization Strategies for Improving Bus Travel Time Prediction Across Networks** This study takes the step beyond single-network analysis to develop multi-network analyses and models; one of the key benefits of standardized open data. It proposes a set of machine learning based models for bus travel time prediction which are generalizable between transit networks. Modeling results reveal forecastable spatial delay patterns. It applies these models to the Seattle KCM (King County Metro) and Trondheim AtB (A to B) bus transit networks and benchmarks them against heuristics and a state of the art travel time model. Generalization strategies such as fine-tuning and training on mixed data are employed and benchmarked. It also proposes a new method for incorporating realtime trip information from other bus trips into arrival predictions with realtime data. This and the other data sources are compared for their ability to improve model performance in an ablation study. This study attempts to address the challenge of information sparsity in open source transit data by using a combination of standardized static, realtime vehicle positions and other urban data sources.

- **Empowering Electric Bus Deployment with Standardized Transit Data** This study culminates in an application and comparison of the previous methods built on open source bus data with those built on closed source data. It proposes a method of predicting trip-by-trip drive cycles for an entire bus network, that is extendable to any bus transit network which reports GTFS and GTFS-RT data. It examines challenges with the resolution of open bus data for this network-wide energy estimation. It also applies the methodology to a wide cross-section of international bus transit agencies. This study demonstrates variation in agency readiness to electrify, their energy and power needs, and how key BEB (battery electric bus) technologies influence the proportion of per-agency blocks that can be electrified.

Chapter 2

MEASUREMENT AND CLASSIFICATION OF TRANSIT DELAYS USING GTFS-RT DATA

2.1 Abstract

This paper presents a method for extracting transit performance metrics from a general transit feed specification-realtime component and aggregating them to roadway segments. A framework is then used to analyze this data in terms of consistent, predictable delays (systematic delays) and random variation on a segment-by-segment basis (stochastic delays). All methods and datasets used are generalizable to transit systems which report vehicle locations in terms of general transit feed specification-realtime parameters. This provides a network-wide screening tool that can be used to determine locations where reactive treatments (e.g., schedule padding) or proactive infrastructural changes (e.g., bus-only lanes, transit signal priority) may be effective at improving efficiency and reliability. To demonstrate this framework, a case study is performed regarding one year of data retrieved from the King County Metro bus network in Seattle, Washington. Stochastic and systematic delays were calculated and assigned to segments in the network, providing insight to spatial trends in reliability and efficiency. Findings for the study network suggest that high-pace segments create an opportunity for large, stochastic speedups, while the network as a whole may carry excessive schedule padding. In addition to the static analysis discussed in this paper, an online interactive visualization tool was developed to display ongoing performance measures in the case study region. All code is open source to encourage additional generalizable work on the general transit feed specification standards.

2.2 Introduction

2.2.1 Motivation

The flexibility afforded by sharing existing roadway infrastructure has allowed bus transit systems to flourish across a spectrum of transportation settings and cultures. However, one inherent disadvantage to sharing existing infrastructure is that buses are susceptible to delays caused by congestion, roadway controls, and shared curb use in addition to regular passenger boarding and alighting; all of which can be highly unpredictable in nature. If these delays are left unchecked, a bus system may become unreliable, and the utility it provides to its riders will diminish rapidly. This might cause ridership to drop, bringing down fare recovery, and eventually leading to service cuts compounding these issues.

To avoid this, transit agencies and planners account for delays when scheduling buses, or simply fix their root causes. When determining where and how to target such treatments, agencies and planners must often make broad strokes to account for cumulative delays across entire routes. One of the primary tools used is schedule padding, which adds buffer time to transit segments that are consistent sources of delay. Schedule padding can improve travel time reliability by accounting for delays but can also hinder it in locations where travel time is highly variable, and it does not fundamentally improve the efficiency of the transit system. Furthermore, the iterative process of assigning buffer time to transit segments is informed using AVL systems, for which analytical tools must be built on a system-by-system basis.

However, the GTFS has provided a fully generalizable framework by which agencies can collect and share transit scheduling data. More importantly, it has led to the development and initial proliferation of the GTFS-RT standard by which actual bus locations and stop times can be shared in a standardized format, once collected by various onboard systems [11]. To date, analytical tools and support systems are relatively scarce for GTFS-RT. This may be because there is little incentive to create them, as agencies collecting this data likely have AVL-based performance measurement tools in place. Although powerful, each AVL-based tool must be individually maintained by its respective agency, and the lack of standardization prevents clear comparison between transit systems.

2.2.2 *Outline of This Work*

This paper examines how the GTFS-RT standard may be used to quantify transit performance in a way fully generalizable to any bus transit network. Bus delays are grouped into two categories: stochastic and systematic, which are quantified through several metrics. These metrics are then calculated at the segment-level across all routes and vehicles in the entire transit network. This enables prioritization for specific locations where infrastructural treatments (e.g., dedicated bus lanes, transit signal priority) or operational treatments (e.g., schedule padding) will be most effective in improving transit performance. Because these measures are quantified from aggregated data across all buses utilizing a given segment, treatments can be identified where they stand to create the largest benefits across all vehicles, rather than individual routes.

Following this introduction, prior work conducted in this area is discussed. The data collection and analysis are presented in the Methods section, with a proposed framework to identify individual instances of delay and examine temporal and spatial trends in delay from the GTFS-RT data. The Case Study section introduces the data sources used for an exploratory analysis of transit delays in the GTFS-RT system for KCM in Seattle, Washington. This is followed by Results and Discussion on what locations exhibit the most delay, how delay varies spatially, and how it relates to segment pace. Last, the Conclusion section summarizes the findings, states the limitations of the study, and presents suggestions for future research.

2.3 Literature Review

2.3.1 *Transit Performance and Delays*

Transit performance is somewhat of an amorphous concept. Many works have aggregated, surveyed, and proposed classification hierarchies for a multitude of established metrics [47, 77]. One of its most commonly emphasized components is travel time reliability; quantified through specific metrics such as expected waiting time, variance of travel time, or on-time performance [47]. In other words, metrics of reliability measure how consistent the system is. One way reliability is frequently reported is through schedule adherence, which relies on measures of individual delays or the amount of time a vehicle has deviated from schedule in a given time period [161]. As delays accumulate, buses deviate further from their schedule. Reliability is one of the most important factors in the utility provided by transit systems; shown to be valued higher than comfort [134] or frequency [25, 48]. Work has explicitly tested the user cost of stop arrival time variability [25] and found it highly impactful. Other work has documented how the utility of reliability extends to all modes of transportation, not only transit [18]. Thus, if bus transit systems are to compete with other modes their reliability must be at least comparable to cars, trains, or other modes.

One of the fundamental treatments for unreliability in transit service is "schedule padding". This is an iterative process of schedule design which allocates additional scheduled travel time to roadway segments tending towards highly variable travel times [30]. While this does not eliminate the delay, it does improve the reliability of the transit service by allowing extra time for when delays occur, and the vehicle falls behind schedule. Unfortunately, when the expected delays do not occur, this can lead to buses arriving at stops ahead of time, which also reduces the overall reliability of the system [71]. In many cases, drivers are encouraged to slow down when ahead of schedule, either by a central dispatcher or onboard clock, thus maintaining equal headways between vehicles [115]. Schedule padding is a relatively simple and inexpensive treatment to apply on a system-wide scale but does not address the actual cause of delays.

Another component of transit performance is transit efficiency, or productivity [77]. This component plays a smaller role in utility provided to individuals using the transit system but has ramifications for cost of operation and net emissions network-wide. Metrics of transit efficiency are expressed as ratios, and might include farebox recovery ratio (operating cost divided by revenue) [47], or volume to capacity ratio (bus flow divided by lane capacity) [78].

Example treatments to improve efficiency might include addition of transit signal priority, queue-jump lanes, or route changes, among others [47]. Schedule padding on its own is not sufficient to improve efficiency, as it simply adjusts bus schedules, without necessarily improving their movement. In fact, if excessive schedule padding is used and a bus slows down to remain on-time, there may be a reduction in efficiency-related metrics for that trip.

The term stochastic delay has been utilized in a transportation context by a handful of previous works. First, in work using GTFS data to determine locations where there is high schedule padding in a network [177]. Here stochastic delays were used to describe all deviations from the GTFS schedule and suggest roadway treatments. In train networks, stochastic propagation of delay was used in a method for modeling how an single delaying event can progress through an entire system [21]. In other work, the quantity of schedule deviation due to individual, delay-causing incidents was estimated through a model using a combination of deterministic and stochastic delay elements [70]. In general, stochastic delays are characterized as occurring due to "non-recurring" traffic conditions, which are thus difficult to predict [159]. These occurrences rely on probabilistic methods to predict accurately [88], and realtime monitoring systems to accommodate [70, 184].

2.3.2 AVL Data in Bus Performance Measurement

To determine performance metrics, and ultimately inform system treatments, data must be collected at stops or corridors of interest. This can be expensive, time consuming and provides only a narrow window of perspective into the nature of delays. AVL technology offers an automated means to record and archive individual bus movements and can thus help illuminate of the locations of delays for each vehicle in the system. They may also communicate with a central dispatch to provide realtime updates to a controlling agency for oversight and performance monitoring purposes.

In prior work, AVL data has been used to quantify bus reliability and efficiency at the individual bus level [23, 34]. Aggregation then allows for metrics to be calculated at the stop, corridor, route, or system levels [23]. Stop-to-stop and route level analysis in particular has been supported by AVL data, with performance measures calculated from schedule adherence being used to determine reliability throughout such corridors in the system [52]. Additionally, AVL data has been used to analyze secondary interactions which are not directly reported by the system such as bus bunching [61], travel time for specific roadway segments [22], and even in aggregate as probes to measure the performance of all roadway users, not just buses [24]. Among these and other works, some classifications for delays based on their cause, or location reported in AVL data have been proposed. For example, classifying the cause of delays based on their coordinates and surrounding infrastructure [41] or whether they occurred due to a scheduled stop [147].

Although powerful in analyzing performance, AVL/APC data is not often made publicly available, lacks a standardized format, and leads to individual "home-grown" analysis tools that are not reusable by other agencies or researchers [71]. Previous academic works have developed tools that suggest automatically generated performance metrics [124], or visualize transit performance through an analysis interface [60, 46]. While effective and useful, these precipitate a burden on individual agencies to maintain supporting software and databases for each technology and its respective analysis tools. It also creates opportunities for discrepancies in the calculation methods for metrics between systems.

2.3.3 GTFS-RT Standard

Public-facing GTFS-RT feeds provide a standardized format for repackaging AVL data, from which schedule padding and other system characteristics can be reliably ascertained [177]. Many agencies currently provide this data through APIs, from which users may request it and receive unified results. Given that the GTFS standard has been adopted by more than one thousand transit agencies globally, the GTFS-RT standard may be poised to follow. Using this specification, generalizable tools for analysis and visualization of transit data can be applied to draw comparisons between networks with ease.

There is currently much less literature on calculating performance metrics from GTFS-RT data than AVL/APC data. This is likely due to the greater establishment of AVL/APC data, and its offering of additional variables and finer temporal detail. In prior work on GTFS-RT, several projects have developed simple realtime observation tools to display immediate bus locations to an end-user such as the web application "Pantograph" and one academic work [186, 59]. Some studies have gone a step further and applied realtime prediction methods to determine delays in active systems and used that information to provide better scheduling information to travelers: This was found to lead to lower overall wait times for travelers, and a more positive perception of transit reliability [174]. Sophisticated methods for archiving GTFS-RT data have also been developed, which is the first step for unified analysis [15]. Preliminary work developed and tested a framework for analyzing bus corridors using archived GTFS-RT data [29]. Their work first developed a method for combining routes into consistent stop-to-stop corridors, then calculated aggregate performance metrics for each corridor.

Overall, GTFS-RT is in a nascent state of adoption, with few open source analysis tools available for performance analysis. One of the largest challenges currently facing GTFS-RT data is quality control [160]. Many GTFS-RT feeds are not aligned with the standard, and thus fully generalizable tools for differing feeds are not possible. As more agencies make bus position data available through GTFS-RT, and additional tools are constructed on the standard, there will be more incentive to fix the integrity of these feeds to benefit from open source work. There is currently ongoing work to develop tools specifically for validating GTFS-RT feeds [14].

2.3.4 Contribution of This Work

This paper contributes to the state of the art through a case study of bus transit data collected from a web API-based GTFS-RT system and disaggregate analysis of segment-level reliability and efficiency metrics informed by that data. It builds on prior work classifying stochastic delays with the additional distinction of systematic delays, accruing from consistent, predictable reductions in transit free-flow speed due to congestion, signals, and transit stops. It is the hope that this analysis can be built upon by future researchers and applied by planners and agencies to direct resources in transit networks to where they are needed most.

2.4 Methods

This section details the process by which GTFS-RT data was collected and analyzed, following the general steps outlined in Figure 2.1.

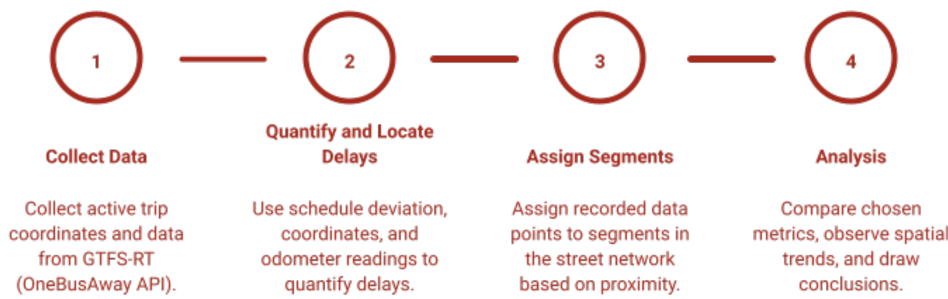

Figure 2.1: Overview of the steps used to measure and classify transit performance.

2.4.1 Collect Data

The data source for this study was a web API providing access to underlying GTFS-RT data for the study region. As a developing standard, there is ongoing debate as to which parameters should be required in GTFS-RT, and which should be optional [14]. The list below documents each parameter used in this analysis framework using the nomenclature of the API from which they were collected. At a minimum, these are the required parameters to recreate this analysis:

- activeTripId: Unique identifier for each vehicle, traversing a single route, on a single day (i.e., a single scheduled run of a route).
- vehicleId: Unique identifier for each vehicle, consistent across days and routes. It is physically painted on each vehicle.

- scheduleDeviation: Difference in seconds between a trip's known position, and where it is scheduled to be as calculated by the AVL system. Can be positive (behind schedule) or negative (ahead of schedule).

- lastKnownDistanceAlongTrip: The distance that the current trip has traveled since its inception. Resets to zero for the next trip.

- lastKnownLocation: The last updated position for a given trip.

- lastLocationUpdateTime: The timestamp corresponding to when a trip's last known location was recorded.

Queries were made at the highest resolution feasible; different buses may update the API at different frequencies, and the system itself may update at its own frequency. The higher the frequency that the data is collected, the more precisely delays can be quantified and located. This comes at heavier computational costs during the analysis and generates more calls to the API. Additionally, it becomes more computationally challenging to assign segment delays to their respective road segments as the size of the location and roadway segment datasets increase. Although the archiving framework used here has suited the needs of this project, there is room for improvement in efficiency and modularization, as detailed by recent work in the field of GTFS-RT "big-data" management [15].

2.4.2 Quantify and Locate Delays

This paper presents several metrics calculated from onboard vehicle data feeds provided in the GTFS-RT format, which fall separately under reliability and efficiency metrics. These are based on time-series data and are calculated between each vehicle update interval. Thus, the unit of analysis used is an individual instance of "delay" or step in time in which a bus performs better or worse relative to either its schedule, or its free flow speed. Metrics were chosen to capture both reliability and efficiency components of the system and are classified for treatment analysis as either stochastic or systematic delays. It is assumed that the current route schedules are appropriate for the expected quantity of delay on each given segment, and thus any schedule deviation is the result of unpredictable variation, i.e., stochastic delay.

Metrics for quantifying delay between consecutive bus locations $i-1$ and i were calculated according to Equations 2.1, 2.2, 2.3 and 2.4. Vehicle pace at time i is calculated using the lastKnownDistanceAlongTrip (TD) and lastLocationUpdateTime (LT) parameters as indicated in Equation 2.1. Total delay (combined systematic and stochastic delay) between location i and $i-1$ is then calculated as a product of the difference between the measured pace and the 95th percentile pace for the segment, and the distance traveled between those two locations (Equation 2.2).

$$\frac{LT_i - LT_{i-1}}{TD_i - TD_{i-1}} = Pace_i(sec/mi) \tag{2.1}$$

$$(Pace_{95} - Pace_i) * (TD_i - TD_{i-1}) = TotalDelay_i(sec) \tag{2.2}$$

Stochastic delay (Equation 2.3) is estimated in terms of the scheduleDeviation (SD) GTFS-RT parameter. Changes in schedule deviation indicate an instance of unpredictable delay, which is available for all buses broadcasting realtime arrival information and is defined as a cumulative measure of the delays and speedups experienced by a vehicle during a given trip. Any decrease in schedule deviation is treated as a negative delay, i.e., the bus sped up due to lack of congestion or other factors. Any increase in schedule deviation is treated as a positive delay, i.e., the bus slowed down due to delaying factors (Equation 2.2).

$$SD_i - SD_{i-1} = StochasticDelay_i(sec) \tag{2.3}$$

Systematic delay (Equation 2.4) is calculated from previously estimated stochastic and total delays. Systematic delay measures the predictable delays that accrue due to congestion and other interference.

$$TotalDelay_i - StochasticDelay_i = SystematicDelay_i \tag{2.4}$$

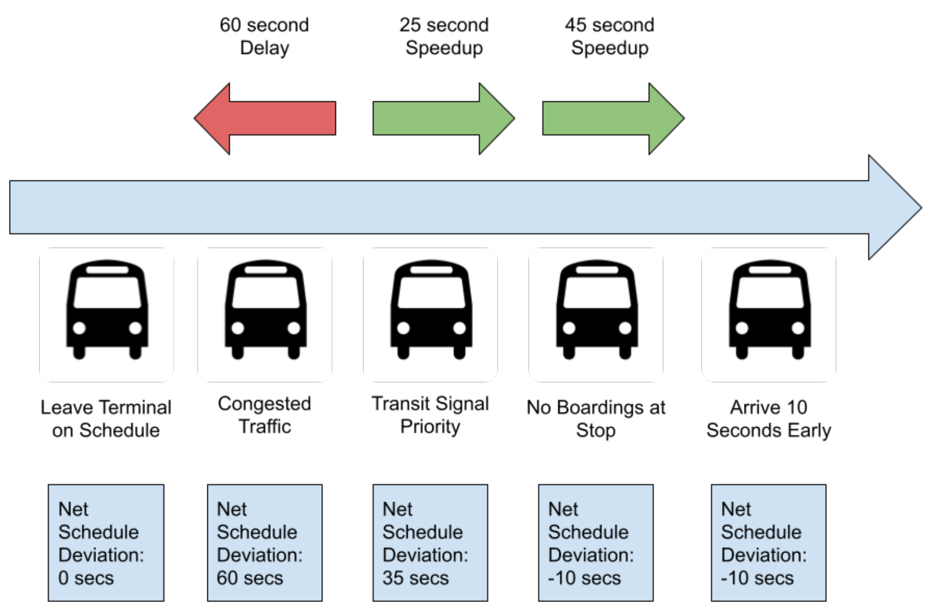

Figure 2.2: The process of determining individual instances of delay from the cumulative schedule deviation parameter.

2.4.3 Assign Segments

After the data is processed to determine all instances of delay during the study period, and each delay has been assigned a location based on the bus location of its second timestamp, each delay location is then assigned to its closest roadway segment. Contemporary practice is to use a map-matching algorithm to assign locations to the nearest geographical feature [149]. In this study, delay locations are matched to street segments based on their last-KnownLocation and spatially indexed coordinates of each roadway segment using nearest neighbors. The atomic units of aggregation used for calculating performance metrics from each observed bus coordinate in this method are street segments of approximately one block length.

2.4.4 Case Study Details

To demonstrate the collection and analysis process of reliability and efficiency performance metrics for GTFS-RT data, a case study was performed. Data was collected from the KCM bus network in Seattle, Washington, over 12 months from August 2020 to August 2021 (Figure 2.3). This section details the data sources used for this purpose.

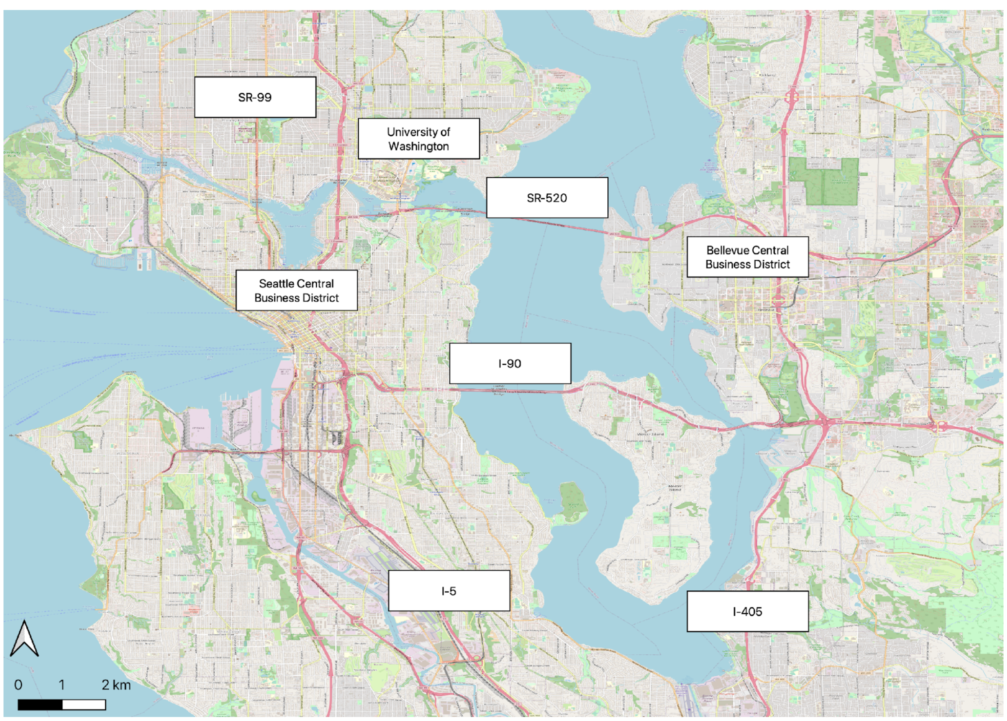

Figure 2.3: Map of case study region with major highways and CBDs labeled.

2.4.5 Data Sources

Several data sources were utilized to record and analyze system-wide bus delays in the KCM network. Bus delays, locations, and other GTFS-RT elements were drawn from the OBA API, which provides access to data for all major transit agencies in the Puget Sound region, and is managed by Sound Transit [63]. The API provides various function calls to access this data, although some underlying variables such as AVL system location accuracy and refresh rate are not documented and can be dependent on the agency or vehicle in question.

Analysis is performed at the segment-level across the KCM network, necessitating separate shapefiles for street segments and active bus routes. The street segments are drawn from the ACS (American Community Survey) TIGER shapefiles. This dataset includes all street segments regardless of whether they are utilized by the transit network and is available for all roadways in the United States. To improve computational efficiency in assigning GPS coordinates for active routes to the closest street segment, these segments are filtered to only include those traversed by buses in the KCM network. This is accomplished using a second shapefile; the bus route shapefile provided by the KCM GTFS data. These routes are buffered, and street segments contained within them are kept in the analysis. This reduces the overall number of potential segments to search when assigning bus coordinates to the nearest roadway and ensures that bus coordinates are only assigned to segments traversed by their respective route. Segments were first determined based on intersecting roadways, then long segments were broken up if longer than a specified length.

Data was collected on both weekdays and weekends. API queries were made starting at 6AM and ending at 7PM. To gather data for all active transit routes, the OBA API was queried once every ten seconds using a single API call to the "vehicles-for-agency" API endpoint, which provides a list of vehicle IDs, and information about the trip status for each vehicle in the response. To obtain information on only the active vehicles, trip statuses listed as canceled, or trips with a null identifier were removed. Additionally, any active trips without posted GPS coordinates were removed, on the assumption that these vehicles were not equipped with realtime tracking technology. These responses were then timestamped and inserted into a relational database.

2.5 Results and Discussion

2.5.1 Summary

After cleaning, the final dataset consisted of approximately 27 million tracked bus locations belonging to 6 thousand unique transit trips. Median stochastic delay was -19 seconds, median systematic delay was 95 seconds, and median total delay was 72 seconds. Figures 2.4, 2.5 and 2.6 show the distribution of a sample of each type of delay across the dataset.

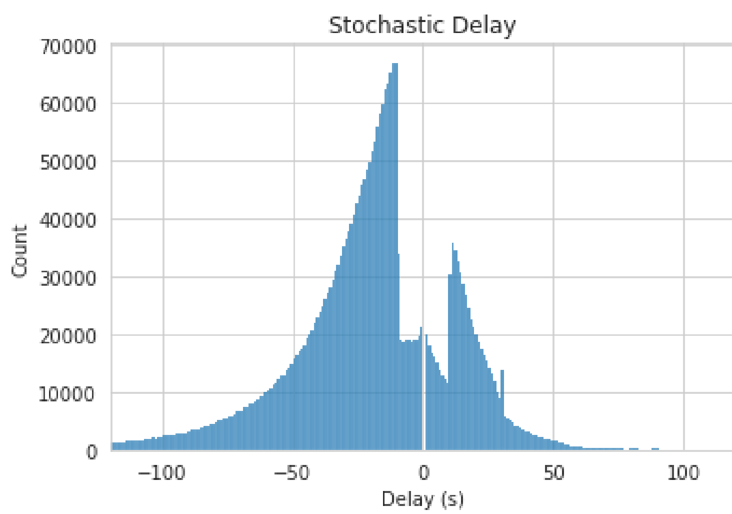

Figure 2.4: Distribution of stochastic delays in the bus locations dataset.

Stochastic delay is approximately normal and centered around approximately -19 seconds. However, as shown in Figure 2.4, instances of schedule deviation reported by the system appear to be rounded down to zero when less than 15 seconds in magnitude, leading to a discontinuity in the center of the distribution at +/- 15 seconds. Because the distribution is non-zero centered, there could be systematic delays that are unaccounted for, and potential reliability improvements gained by adjusting bus schedules. In this case, the distribution of

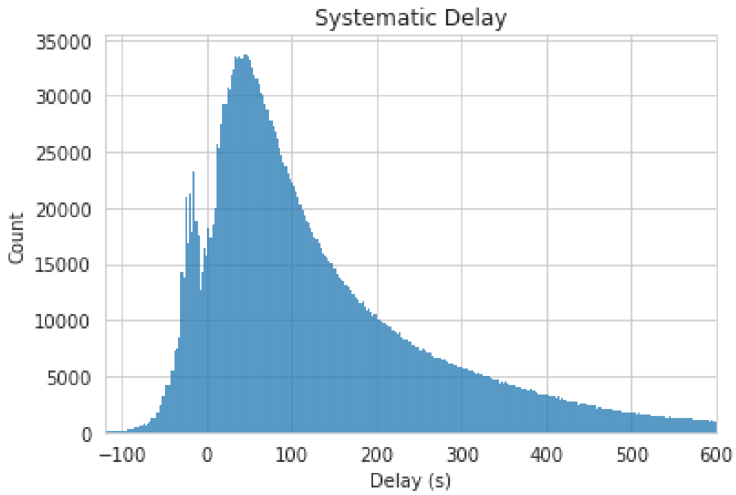

Figure 2.5: Distribution of systematic delays in the bus locations dataset.

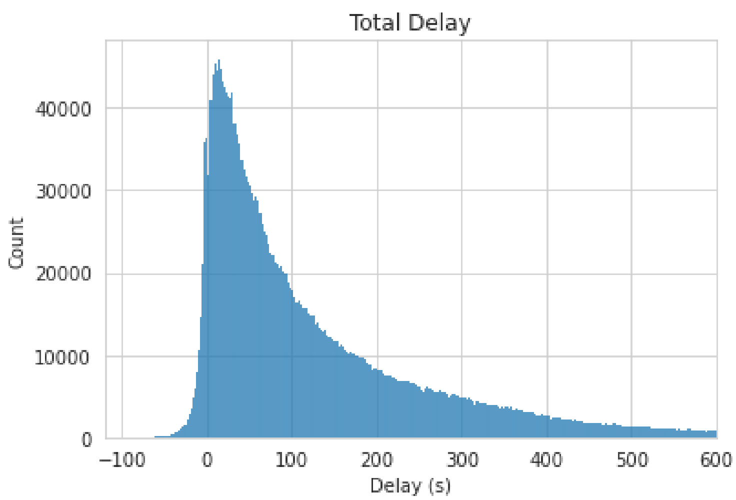

Figure 2.6: Distribution of total delays in the bus locations dataset.

stochastic delays is skewed negative; therefore, buses are traveling ahead of schedule more often than behind or on-time. One cause of this may be over-padded bus schedules, as buses are consistently traveling faster than intended. The net effect may be a less reliable system, as passengers arrive at bus stops only to find that their bus has already passed them by.

Treatments might include training for drivers to wait at hold positions when ahead of schedule on segments with high stochastic delay, or overall schedule adjustment to decrease allocated travel time. Stochastic delays greater than 120 seconds in magnitude were quite rare, with the combined set of all stochastic delays having a standard deviation of 34 seconds. The standard deviation of stochastic delay may be used to compare the relative unpredictability of segments within and across networks.

Total delay is skewed right and positive. This is because free-flow pace has been defined in this study as the 95th percentile, and total delay was calculated using the difference between actual and free-flow pace. Systematic delay is also skewed right, making it mostly positive. Due to its derivation from total and stochastic delays, there is a small discontinuity near -15 seconds. This may again be the result of schedule deviation being rounded down to zero by the system when smaller than 15 seconds in magnitude.

2.5.2 Segment Results

The analysis lends itself well to identifying hot spots in the network which carry high-magnitude delays. Figures 2.7, 2.8, 2.9, 2.10, 2.11 and 2.12 show the quantitative and spatial distributions of each classification of mean delay experienced by buses traversing a given segment in the network.

One of the surprising findings of this case study was the presence of very high stochastic speedups on some highway segments relative to local streets. This is shown most clearly on the north-south traversing I-5 segments north of Seattle which each carry average speedups of about 1-2 minutes. This may be because bus schedules accounting for time between stops group these segments with local streets. So, when a bus leaves the highway, and continues on local streets to its next stop, it accumulates delays which balance out the highway speedups. Alternatively, this may be explained by large amounts of schedule padding for highway segments (which can have highly variable speeds depending on congestion conditions) with the understanding that drivers may wait at control points if not delayed.

In comparison to the stochastic delay results, some aspects of systematic delays were

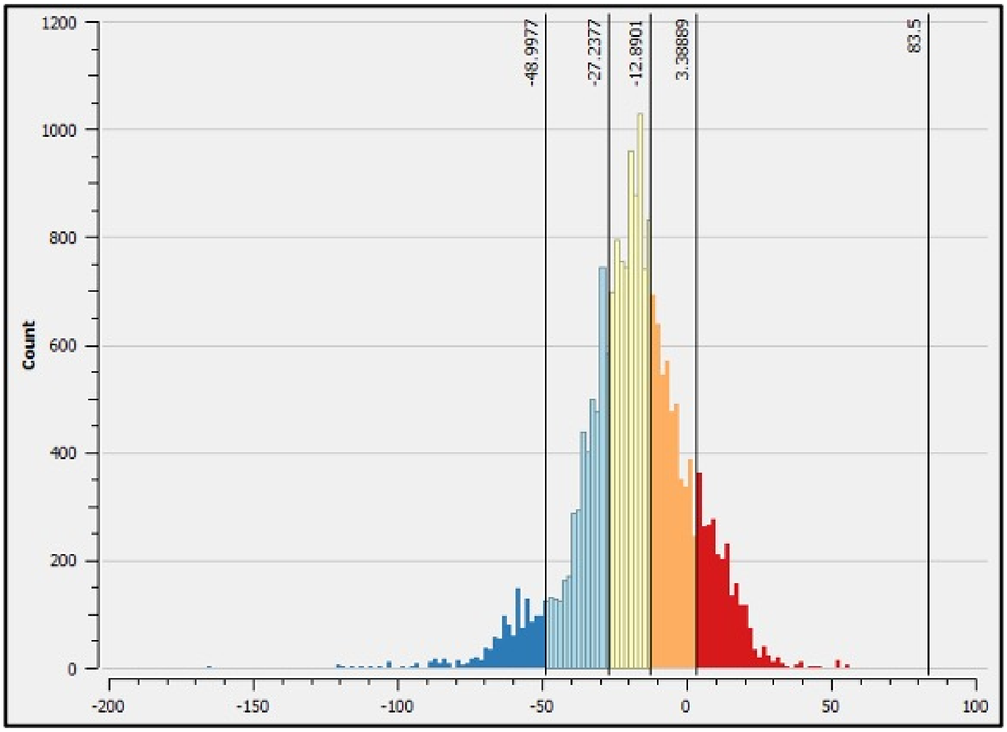

Figure 2.7: Distribution of mean stochastic delay for all segments in the study network.

similar. Again, certain highway segments such as the east-west I-90 and SR-520 segments carried relatively low average delay. However, the distribution of mean segment delay reveals a shift from speedups to slowdowns, with most segments carrying a positive average delay. With the exception of a handful of segments, the magnitude of stochastic delay is relatively small compared so systematic delay. Thus, the trends of total delay (Figures 2.11 and 2.12) tend to mirror those of systematic delay.

In addition to mean delay, the mean pace is calculated for each segment in the network across all observations in the dataset. The spatial distribution is shown in Figure 2.13. This provides perhaps the clearest picture of efficiency in the network derived from GTFS-RT, despite its lack of passenger data. The largest cluster of slower segments is found in the CBD, with additional zones to the east (CBD of neighboring city Bellevue) and north (University of Washington). As would be expected, highway segments are consistently higher pace than

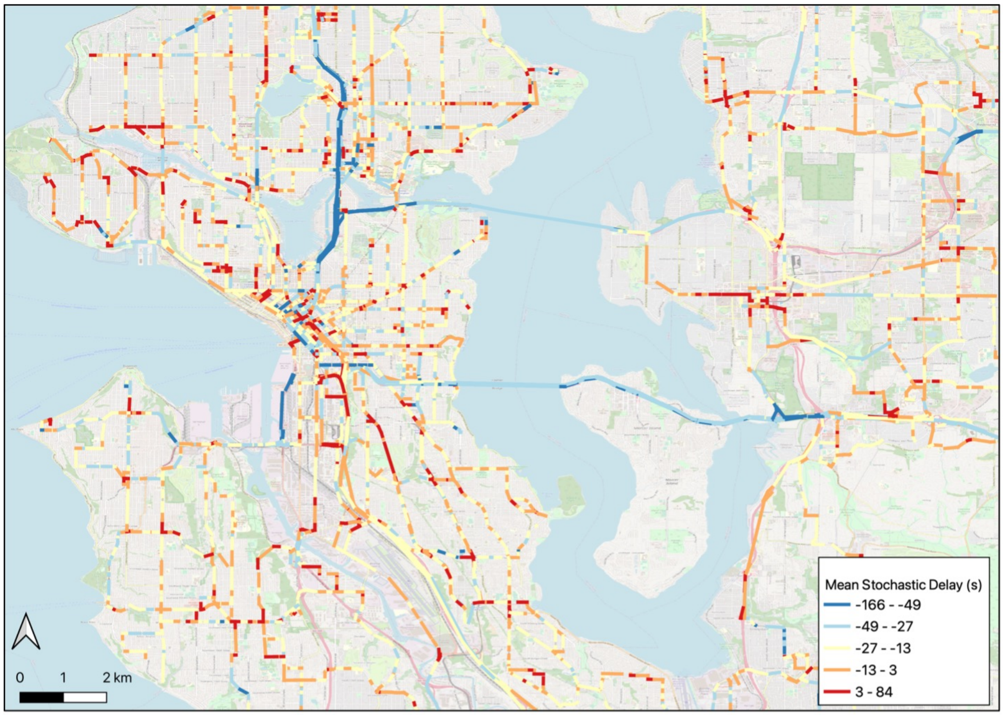

Figure 2.8: Spatial distribution of stochastic delays in the study network.

the rest of the network. Of course, this does not necessarily make them more efficient than the rest of the network, as they have fewer stops and higher speed limits.

These results present a static analysis of 12 months of transit location data; however, the primary strength of GTFS-RT is that it is real time and can provide continuous performance insights as buses traverse the network. Thus, an interactive transit network performance visualization tool "TransitVis" was developed for the data used in this paper and is available online, along with open source Python code for creating the necessary databases and performing the analysis. This tool uses the same analysis framework as discussed in this paper but provides additional performance measures and a higher temporal resolution (Figure 2.14).

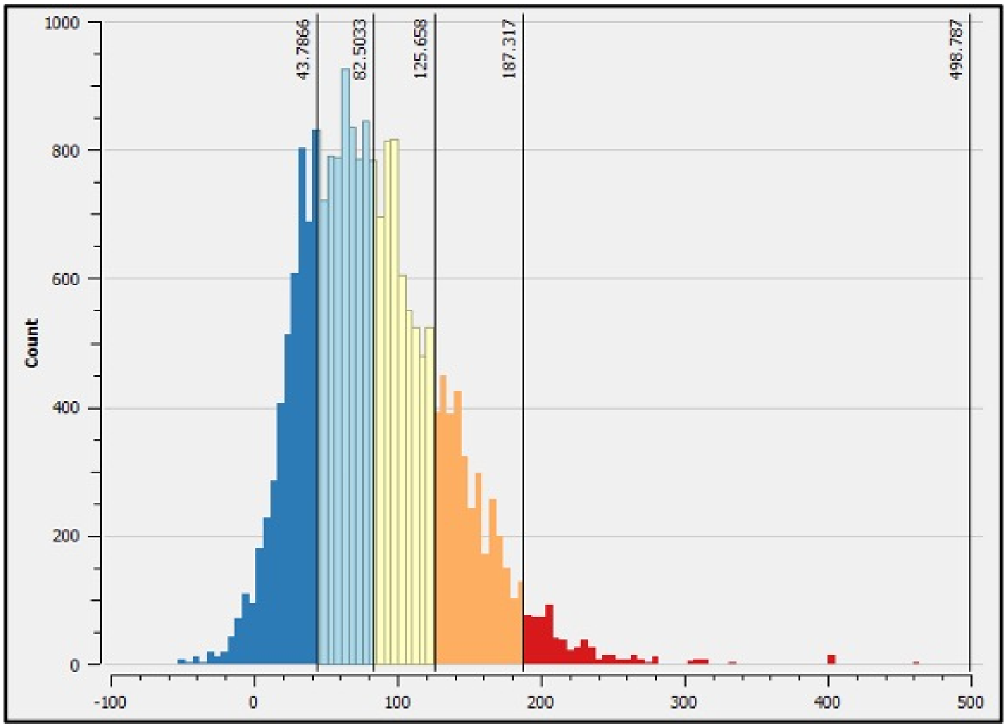

Figure 2.9: Distribution of mean systematic delay for all segments in the network.

2.6 Conclusions

This paper developed a framework for utilizing GTFS-RT data for locating and characterizing service performance metrics. It outlined a framework for collecting and analyzing transit delay data on a system-wide scale, using a case study of the KCM network in Seattle, Washington.

Two metrics were developed that GTFS-RT data was capable of informing. The first, stochastic delay, was based on the schedule deviation parameter, which is an indication of delays that are not accounted for by schedule padding or intentional driver checkpoint slowdowns. In the case study, stochastic delay was distributed equally throughout the network, with the exception of highway segments which experienced unscheduled speedups. The second metric, systematic delay, was based on odometer (distance) and timestamp (time)

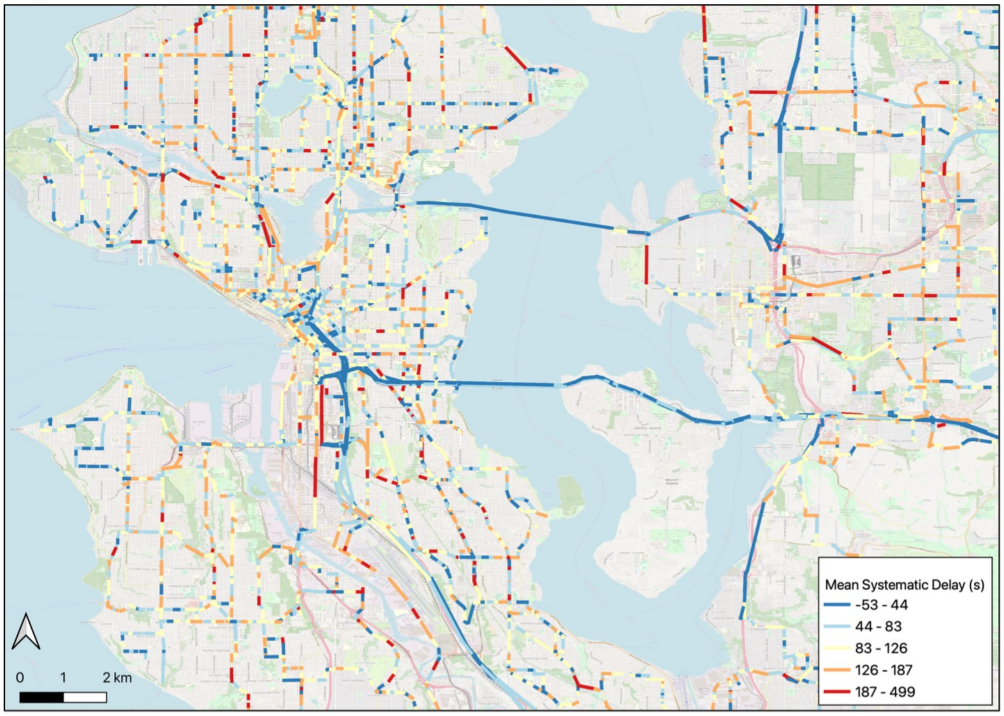

Figure 2.10: Spatial distribution of systematic delays.

sensors reported in the system. This allowed for the calculation of average speed and pace at each tracked location. 95th percentile pace was taken as the free-flow, non-delayed pace, and delay was calculated based on the GTFS-RT measurements. In theory, this should account for all systematic and stochastic delay that occurred between tracked locations. This assumes that the current schedule properly accounts for all systematic delay. However, the distribution of stochastic delays in the system was centered slightly towards speedups, indicating that perhaps on a system-wide level the current schedule is slightly over padded.

This framework allowed for segment-based analysis of transit hotspots and differentiated between types of delay which may require different types of treatment. For example, less reliable areas with negative stochastic delay may be treated by removing schedule padding. Whereas segments with high systematic delay may be inefficient and could be improved with transit signal priority or bus lanes. Any corridor identified by either of the two metrics may

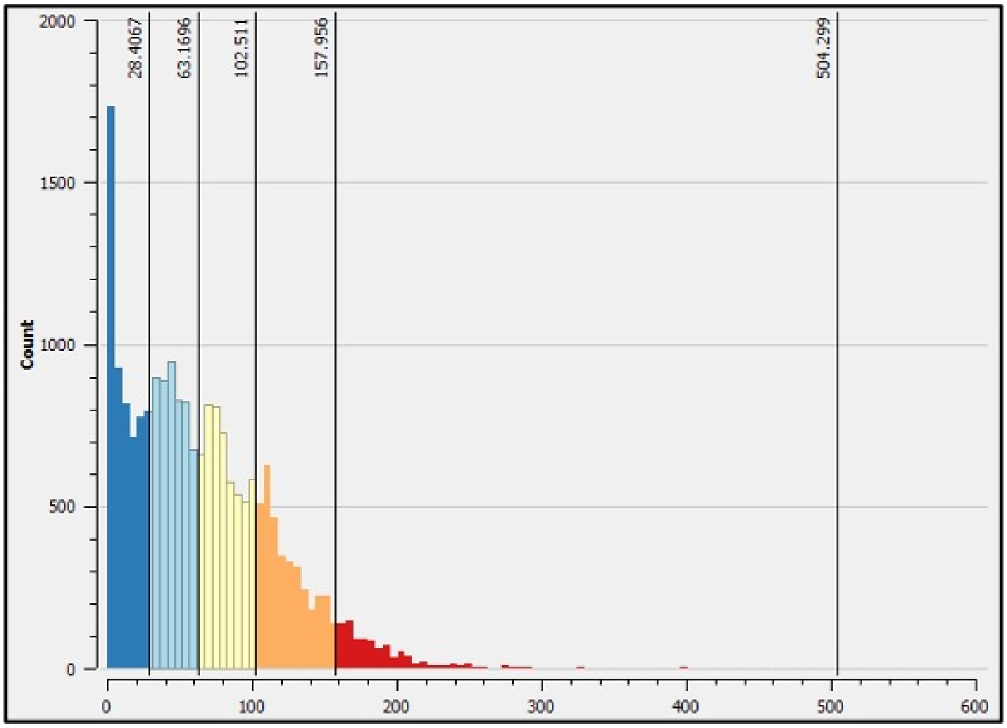

Figure 2.11: Distribution of mean total delay for all segments in the network.

also become candidate for follow-on observation to identify the exact causes of delay. In fact, because GTFS-RT has less detail than combined AVL/APC data, the main use-case for this class of analysis may be as a preliminary screening tool for locations in the network with performance anomalies, which can then be used to target additional monitoring resources more efficiently.

This approach also does not measure other important components of transit performance, such as accessibility, which may rely on factors of the built environment to quantify, for which there is currently no standardized data source [100]. Additionally, many measures of efficiency rely on data gathered from APC systems, which record the number of passengers boarding and alighting at each transit stop. Bus occupancy is currently an experimental feature of GTFS-RT. This data is not currently available in the feeds used for this case study but would greatly improve the number of performance metrics that can be informed

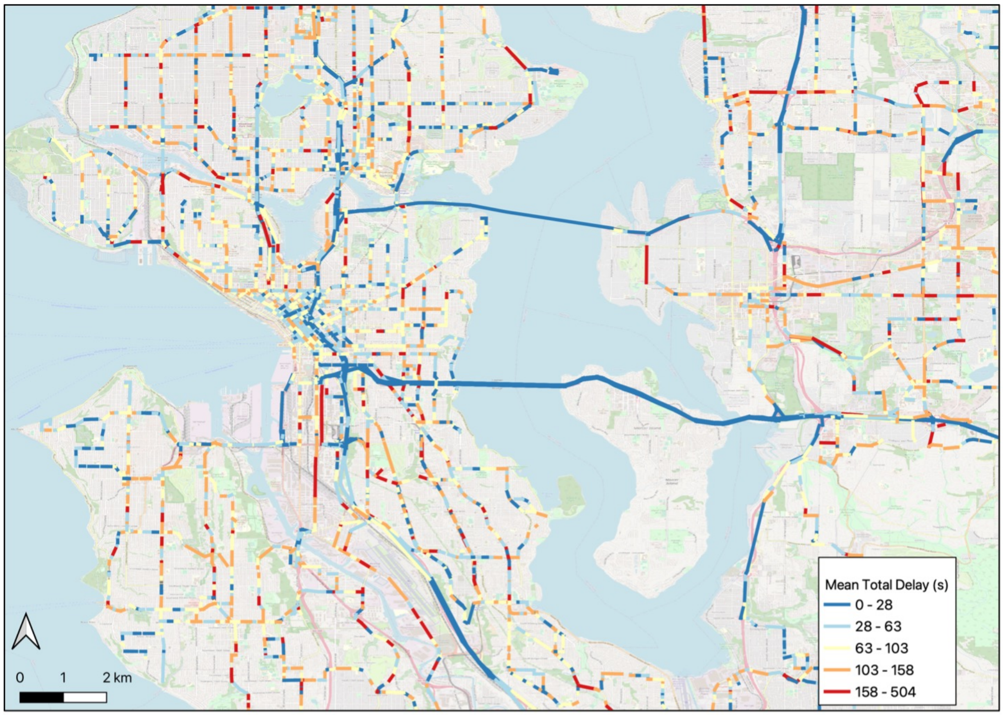

Figure 2.12: Spatial distribution of total delay.

by GTFS-RT.

Last, this work focused on a segment-based analysis of transit performance. However, the GTFS-RT standard includes additional information on the current stop of each active vehicle. This could be used to further separate analysis based on delay accrued at segments, and delay accrued at individual stops. This would provide a more nuanced analysis of why certain segments or corridors decrease performance relative to others and open the way for predictive delay models based on GTFS-RT data. Future work might build on these methods through the inclusion of AVL, field observation, and/or video data as ground truth validation for information provided in a GTFS-RT feed. This more detailed data may also allow for the development of more sophisticated modeling parameters related to bus movement such as acceleration, average queuing distance for individual intersections, or average stop dwell time. Deriving these secondary parameters from those in GTFS-RT would inform more

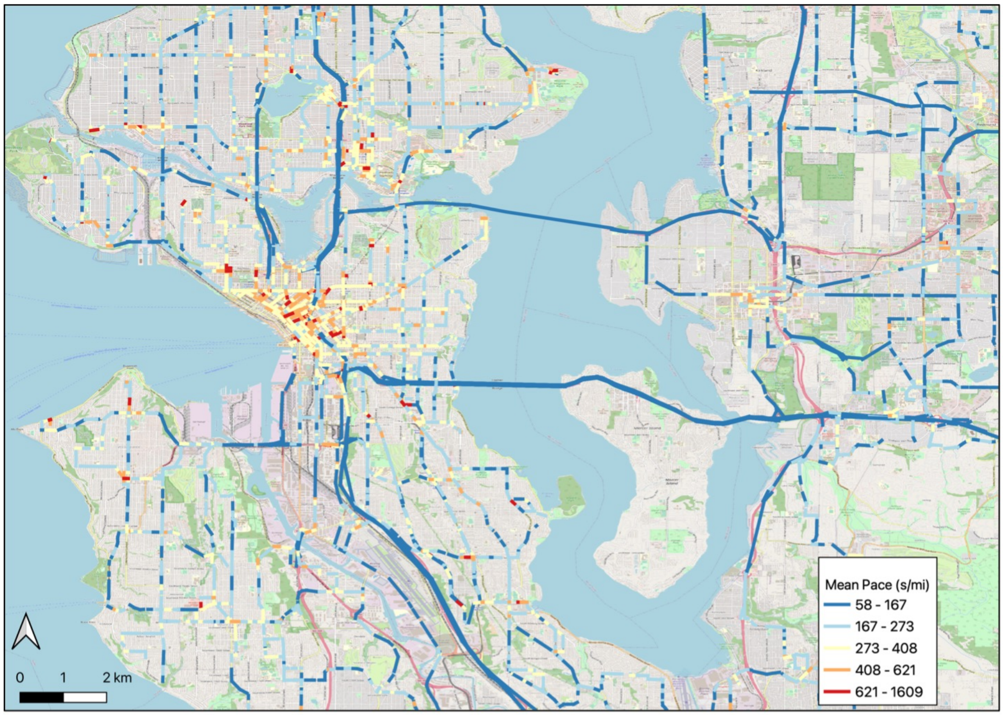

Figure 2.13: Spatial distribution of pace.

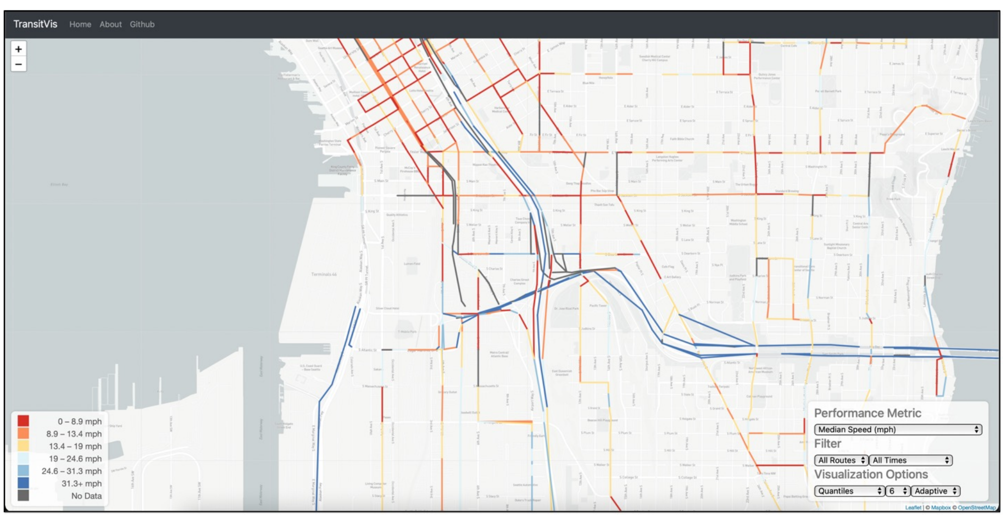

Figure 2.14: User interface for interactive visualization tool.

targeted treatments.

Chapter 3

GENERALIZATION STRATEGIES FOR IMPROVING BUS TRAVEL TIME PREDICTION ACROSS NETWORKS

3.1 Abstract

This study focuses on developing and evaluating predictive models for bus travel times adaptable to any transit network, or to new roadway segments without prior travel time data. Most prior work relies on non-standardized features such as road traffic forecasts or closed-source datasets to test predictions on a single route or network. This work leverages standardized and open-source data from general transit feed specification feeds to gather four months of realtime bus position data from Seattle and Trondheim's transit networks. It then tests and refines strategies for generalizing model predictions across both locations. To achieve this, a data pipeline is developed to process and clean the raw data, then extract features from the standardized sources. Performance is then evaluated for several deep learning and heuristic models in predicting bus travel times between source and target bus networks. Holdout data is taken from selected routes in the source city to validate the internal generalization of the models. Data from the target city is used to evaluate the external generalization of the models. An ablation study explores the impact of different open data sources on model generalization (GPS, static timetables, OpenStreetMap and other realtime trips). This is extended to the analysis to 33 international bus networks, placing the results in broader context and testing fine-tuning strategies for generalization. Results show that deep learning methods generalize well within the source network, with as little as 1% loss in mean absolute percent error on holdout routes. With minimal fine-tuning generalization is significantly improved on the target network. Model features built on static schedule data, realtime positions or OpenStreetMap embeddings improved generalization performance (up

to 7% reduction). This was more pronounced for networks with a greater initial quantity of training data. As a route-planning tool for roadways without prior data, geospatial data mining can provide reasonable bus travel time estimates. For cross-sectional bus network analysis, fine tuning on at least 100 trajectory samples for each target network is required to significantly outperform baseline heuristics. This necessitates a standardized realtime data feed in the target city.

3.2 Introduction

3.2.1 Motivation

Efficient bus transit systems play a crucial role in enhancing accessibility and reducing reliance on personal vehicles. The success of such systems hinges on factors like dependable schedules, network coverage, and swift travel times. Crucial to achieving these objectives is the ability to accurately forecast bus travel times across diverse locations within a transit network.

A bus transit agency has access to very high-resolution, network-specific, integrated data on where buses travel and what exactly slows them down. Technologies like AVL, AFC and APC, furnish comprehensive data on realtime bus locations, fare tap-on and tap-off events, and passenger boarding/alighting counts, respectively. These systems are typically proprietary and implemented on a case-by-case basis, making them specific to the transit network they are constructed around. From an agency perspective, this simplifies procurement of data collection and monitoring tools. It also ensures interoperability between the onboard bus sensors, data collection and analysis software.

However, researchers and planners often perform analyses that transcend individual transit networks. Simpler models and heuristics such as average speed suffice for broad estimates, but they overlook the variation of travel speeds within individual networks. Achieving a balance between wide applicability and network-specific nuance defines the pursuit of a 'generalizable' predictive model for bus travel times. A generalizable bus travel time model can be used by planners to predict travel speeds for temporary re-routes, headway and time of day changes, or entirely new lines. It can also be used by researchers to predict travel times across an array of transit networks. Consider the challenge of predicting adoption of a new fixed-route transit service across many cities. Specific travel times would be needed for numerous transit systems, stop configurations and road networks.

Building such a generalizable model depends on widespread availability of standardized and open-source AVL data. Fortunately, there are well-adopted standards for sharing transit data: GTFS, GTFS-RT, NeTEx and SIRI. These standards are used by many transit agencies and third-party applications to provide schedule and realtime arrival information to travelers. The GTFS and NeTEx formats are used to share static schedule data. This includes stop locations and times for all routes across different days of the week. The GTFS-RT and SIRI formats are used to share realtime bus position data. This is repackaged and standardized AVL data that includes exact positions and timestamps for individual buses.

These standardized feeds provide ample data for building and testing bus travel time models. However, they have lower resolution and feature availability than the underlying AVL data. As discussed in later sections, many prior models are built on AVL data that is requested on an individual basis from transit agencies. This gives them strong performance for specific networks and routes but ultimately limits their scope. These models are rarely tested on multiple networks, and it is unclear how well they can generalize. This limits their utility for researchers who need to predict travel times across many networks with standardized data, and for route planning purposes where prior data are unavailable.

3.2.2 Outline of This Work

This work tackles two research questions:

1. How effectively can geospatial data mining generalize travel time predictions to new routes or different bus networks?

2. To what extent can this generalization be improved through fine-tuning, or through

additional widely available features?

To address the first question, models were trained using a realtime bus position dataset collected for four months from the KCM and AtB bus networks in Seattle, United States and Trondheim, Norway. To assess generalizability, model performance was tested within each network on a set of routes that are held out from the training data. This simulates a route planning use-case where the model must predict travel times for bus trips it has not seen before. The holdout data provides ground-truth validation for this test.

Models are then tested across the two bus training networks. This simulates a use-case where a model trained extensively on a source bus network is extended to a different target network. The model which has been trained on four months of data from the target bus network provides validation for this test. Last, the models are tested on one week of data collected from 33 international bus networks. This frames the results for these two cities in a broader context.

To address the second question, fine-tuning and new features are incorporated to attempt to improve on model generalization. First, fine-tuning is tested to determine the extent to which a small sample of data from the target network can improve model performance. Then an ablation study is conducted to evaluate the impact of various feature sources to generalizability. These sources include basic GPS (global positioning system) coordinates, static schedule information, OSM tags, and ongoing trip information from the GTFS-RT feed. The study concludes by synthesizing the results of these experiments to draw conclusions regarding the viability of generalizing this class of models across different bus transit systems.

All analyses rely only on open source and standardized feature data collected from GTFS, and GTFS-RT feeds. These open data allow the models to be trained and tested on any bus network that reports position and/or schedule data in the standardized format. However, they present challenges related to lower resolution and feature availability over directly using AVL data. Incorporated embedding features from OSM data are also open source and standardized.

The subsequent section of this paper comprises a review of the state of the art in bus transit time estimation as it relates to this work. Then a description of the data and methods employed in the experiments is provided. The study concludes by engaging in a discussion of the generalization and ablation results and their implications.

3.3 Literature Review

There is little to no existing literature on fine-tuning bus travel time predictions across multiple networks with GTFS-RT or other open source standardized data. Most work focuses on comparing model architectures for a single network or route, and many rely on data sources that are specific to the network of interest. This greatly restricts their wider applicability. This section examines some of the most pertinent prior work and highlights limitations that will be addressed in this study.

3.3.1 Spatiotemporal Data Mining for Travel Time Forecasting

Spatiotemporal data mining seeks to extract patterns and relationships from large quantities of raw geospatial data. At its simplest form, coordinates of geospatial point, raster, or vector data are interpreted as features and used as inputs to model geographic quantities of interest. Map-matching and other preprocessing strategies can be used to refine these into more complex geospatial features [156]. Some common use-cases for these techniques are anomaly detection, event coupling, spatiotemporal prediction, partitioning, hotspot analysis and evaluating system changes [156, 125]. Trajectory data mining is a subset of spatiotemporal data mining which focuses on the movement of objects through space and time [156, 189, 125]. It is particularly useful for transportation systems, where GPS traces from the movement of vehicles and people can be analyzed to improve system performance and safety.

Deep learning techniques are often applied in spatiotemporal data mining for transportation. For example, they are used to model complex relationships between highly-dimensional urban features and vehicle trajectories [172]. One fundamental task in this domain is travel time prediction. Though deep learning techniques can model complex influences of the urban form on travel time, they require heavy computation, a large dataset, and lack interpretability of simpler models [172]. However, prior work has shown that deep learning models consistently outperform simpler models and heuristics for this task [16, 56, 98, 167] making them an obvious choice when feasible.

3.3.2 Applications and Limitations of Studies Using Data Mining in Bus Travel Time Forecasting

There is prior work on bus travel time prediction using spatiotemporal data mining and deep learning techniques. However, many of these studies are tested on a small subset of routes or trips. Furthermore, many rely on data sources that are specific to the network of interest. Both factors greatly restrict their wider applicability.

Among these studies, several have focused on comparing various machine learning methods for the purpose of predicting bus travel time. For example, one study evaluated a comprehensive set of classical machine learning models on data from the Tumakuru bus network [13]. They found that more complex, nonlinear models outperformed simpler ones for bus travel time prediction, but only tested models on one route, with one month of data. Similar model comparison studies in Sydney and Macae came to the same conclusion on problem complexity. They included basic deep learning approaches such as LSTM (long short-term memory) and ANN (artificial neural network) models. Their analysis was also limited to one route [143, 56]. MAGTTE took a step towards within-network generalizability by making predictions where historic data is sparse to non-existent but was also ultimately limited to three routes [121].

Others propose more complex model architectures and/or fitting techniques compared to baseline benchmarks. For example, one study tested MLP (multilayer perceptron) and LSTM deep learning approaches with and without genetic algorithm training and a Kalman filter. These were also limited to four routes [132]. Kalman filters are one of the most commonly used methods for rectifying a given state-model with predictions from a deep learning model [151]. Other studies have tested similar combined Kalman filter/LSTM approaches for forecasting bus travel times but were also limited in route and network scope [188, 146, 118, 155, 185, 190, 113].

Another challenge to the generalizability of past work is the use of closed source or difficult to obtain data. For example, Bustr is a model which underlies Google Maps and provides travel time estimates where realtime information is unavailable [16]. The spatial and temporal scope of the training data are also closed source, though GTFS-RT data is used for validation. It relies in part on a car forecast to predict bus travel times. Similarly, BusWTE is a bus-specific model which underlies Baidu Maps and was trained separately on four million records from Xiamen and Nanjing [152]. It also relies on a car forecast. Another bus-specific model is DEEPTRANS which can make predictions for the full LAMTA network. It was tested in online prediction with a web interface [164]. However, it uses a traffic forecast based on roadway loop detector data specific to the network. Each of these models takes advantage of a car forecast to predict travel times for bus routes where no prior GPS data is available, making them generalizable within their respective networks.

Last, AVL data provides specific stop-to-stop travel times and dwell times at bus stops. It is often used to train models which break the travel time prediction problem into predicting times for road segments between stops and predicting dwell times at stops. The total travel time is then added up. This is a common approach in the literature [42, 131, 182, 113]. This provides very robust estimates for specific networks. The main challenge with this approach is the fact that GTFS-RT data is not as specific as AVL data. It provides realtime bus positions and trip updates but does not provide specific travel times between stops. Therefore careful map-matching and interpolation must be applied if using the standard. There is also no readily apparent way to transfer parameters learned for a given network in this way to different networks.

Looking outside of bus travel time prediction, DeepTTE is a model developed on taxi trajectory data from Chengdu and Beijing [171]. Its code and a data sample are available open source. Likely for this reason, DeepTTE is used as a benchmark for many of the studies listed throughout this review. DeepTTE focuses on mining temporal and spatial speed information for a network from a large quantity of GPS data. Unfortunately, it is not a bus-specific model, however, some bus-specific features can be added to the existing model architecture (e.g., by simply increasing the number of features in a given input dimension). For other bus-specific studies which are benchmarked against DeepTTE, it is not clear if their additional features were added in this manner. DeepTTE is used as a flexible, multi-city tested spatiotemporal data mining benchmark in this study.

3.3.3 Using Transfer Learning to Generalize Travel Time Forecasts

There are several studies that have applied transfer learning in the travel time prediction domain. First, SSML is a model which was trained on data from Baidu Maps [57]. It incorporates few-shot samples of past driving behavioral features from an ongoing trip to improve inference for the remaining travel time of that trip. The base model on which tuning was performed was ConSTGAT [58]. The goal was to use the model to predict more accurate travel times for a trip already in progress. The model is trained on GPS data and is tested in a real-time prediction setting. The model was tested on three cities. For shorter trips (t=5 minutes) the few-shot training had little additional benefit. However, for trips longer than 10 minutes the model outperformed baselines by 5-10 minutes MAE (mean absolute error). This could be applied to bus travel time prediction for realtime arrival times, where predictions are updated in realtime based on ongoing trip information.

Second, AtHy-TNet was developed to accommodate a mixture of data from different vehicle types [116]. Though it was not necessarily bus-specific, the authors tested a model trained on taxi data then tuned on bus data. They suggest that bus data itself may be difficult to acquire in sufficient quantities for deep learning. However, this could be addressed with open bus data sources such as GTFS-RT feeds rather than individually requested AFC records from agencies, as was used in the paper.

Last, MetaST is a generic spatiotemporal data mining model that tests transfer learning for generic cross-city tasks (it is demonstrated on traffic flow and water quality prediction) [183]. In that work a pre-trained model is initialized on a source network then tuned to a target network. The model is tested on a taxi traffic volume dataset from five US cities and a bike dataset from three US cities. They benchmarked against single fine-tuning, in which the model from the source city was tuned on a small sample from the target city. They also tested multi fine-tuning, in which the model from the source city was tuned on combined samples from many different target cities.

3.3.4 Contribution of This Work

This study contributes to the literature by testing the generalizability of deep learning models for bus travel time prediction across different bus networks. The approach differs from prior work in several ways:

- Features and data are specific to bus travel time prediction. There are many spatiotemporal data mining approaches to travel time prediction, but most are not specific to buses. Those that are often impose constraints on the input features that are limited to AVL data, which are impractical to apply on a wider scale.

- Only open source and standardized data sources (GTFS, GTFS-RT and OSM) are used to train and test the models. This limits the resolution and feature availability of the data, but allows them to be trained and tested on any bus network that reports position and/or schedule data in the standardized format. No car forecasts or other closed or non-standardized features are used. Models are tested on unseen routes within the training bus network, and on entirely unseen networks.

- To improve model performance, fine-tuning is tested on single and multi-city data samples. Findings are used to examine the feasibility of transfer learning for creating generalizable bus travel time forecast models. The study does not propose new transfer- or meta-learning strategies, but rather tests the feasibility of existing strategies in the bus travel time prediction domain.

3.4 Methods

The data and experiments in this study were designed to test the generalizability and performance of various deep learning models for bus travel time prediction when trained on open source data. This section contains a description of the datasets used for training and validation from the KCM and AtB realtime bus location feeds. Then the set of experimental tasks are described which for each model are used to gauge performance across various use-cases.

3.4.1 Data Sources

Data was collected from two bus transit networks in different transportation and urban planning contexts. The training and testing data for this study are time stamped, trip-identified GPS coordinates collected from onboard bus sensors. Supplemental features are used in the models with information from the GTFS feeds for both networks and examine the effects of unavailability of either GTFS or GTFS-RT in ablation tests. The KCM network provides GTFS-RT data through the OBA API. The AtB network provides GTFS-RT data through the Entur API. Both APIs were queried at 30 second intervals, after each query all new trip updates were recorded to a database. Table 3.4.1 summarizes some key points of these datasets.

Table 3.1: Summary statistics for data collection from KCM and AtB networks.

Metric	Seattle (KCM)	Trondheim (AtB)
Collection Period	2023.02.13 - 2023.06.15	2023.02.13 - 2023.06.15
Observed Days	120	120
Observed Points	85,440,000	14,537,000
Approximate Shingles	3,785,000	658,000
Network boundary (sqkm)	1,972	775

3.4.2 Feature Construction

Each data source provides a set of model features (static schedule from GTFS, traffic conditions from GTFS-RT and urban environment from OSM). Similar to past work on geospatial data mining for travel time estimation, the data begins with GPS coordinate features and time embeddings [171, 152]. Information on the built environment is captured through embeddings [180]. Information from the closest stop (e.g., scheduled arrival time, distance) in the static schedule is also included [16, 131, 89]. Information from past trips in the GTFS-RT feed to estimate ongoing road traffic conditions is also used, another common feature source in past work [16, 164, 152]. In summary, the features are:

- **GPS-timestamp time embeddings.** Time of day is encoded to 1440 values with 48 embedding dimensions. Day of week is encoded to seven values with four embedding dimensions. The embeddings are trained along with each model. This provides the model with information about the time that the trajectory occurred.

- **GPS-based trajectory features.** Locally projected X and Y coordinates, distance and bearing are calculated between each point from the geometry of the trajectory data. This provides the model with information on the location and direction of each trajectory.

- **OSM tag embeddings.** Pre-trained OSM embedding models from hex2vec are gathered using the SRAI (Spatial Representation for Artificial Intelligence) library [180]. The embedding space is a hexagonal grid covering each target network. The embedding method used is a spatial skip-gram fit using negative sampling as proposed by [180]. This provides the model with information on the built environment along each trajectory.

- **Static schedule features.** Closest stop distance, scheduled travel time to that stop, and the number of stops passed are calculated between each point from the geometries of the trajectory data, and the static feed. This provides the model with information on the scheduled travel time for the inference trajectory.

- **Static schedule embeddings.** A set of GTFS embeddings based on standardized static schedules is pulled from each of the target networks (including the 33 international bus networks). The embedding space is the same hexagonal regions as the OSM tag embeddings. The embedding method used is an autoencoder fit using MSE (mean squared error) as proposed by [84]. This provides the model with information on the scheduled travel times and directions of all scheduled trips near the trajectory.

- **GTFS-RT ongoing trip features.** GPS-based features for trips in the training data that occurred prior to inference are included in the model using Equation 3.1. This provides the model with most recent information on the traffic conditions experienced by buses along the path at the time of inference.

Basic GPS and Data Cleaning

After data collection, some preprocessing steps were applied to each bus trip. A "trip" is a single scheduled run of a route on any given day. To make predictions on sub-trip-length trajectories, each observed trip is broken randomly into 2-5 shingles. "Shingle" is used to describe a continuous unique chunk of a trip [16]. Then each point of each shingle is spatially matched to its closest stop in the static timeschedule.

Next, coordinates were projected into a local CRS (coordinate reference system), and calculate distance, mean speed and bearing between consecutive points in a shingle. Then points were filtered having consecutive distances greater than five km, times greater than five minutes and speeds outside the range of 0.2-35 m/s. The main purpose of this filtering is to remove points that may have been recorded due to GPS multipathing or other sensor issues. These filtering criteria are similar to the most relevant prior work (which used GTFS-RT data as training and testing data for deep learning). Their filtering criteria were three km, five minutes, and 0.2-40 m/s [16]. DeepTTE (literature benchmark for a more complex model) does not report any data cleaning or filtering steps [172]. However, their taxi trajectories range from 11-128 points, spanning 2-50 km. [16] do not provide summary metrics for their post-processed samples.

Following this initial filtering, the distance, mean speed and bearing are re-calculated between consecutive points (note that these values have changed after the prior point filtering step). After re-calculating these features, shingles containing points which still don't meet the filtering criteria are assumed to be due to an erroneous shingle rather than temporary sensor issues. Filtering is completed by removing any shingles containing points that don't meet the prior filtering criteria. Last, 20% of the remaining data points are randomly dropped from each shingle. This is done to ensure that points are not uniformly spaced in time, as is often the case with regularly updated onboard sensors. This would allow the model to more easily predict during training/testing than at inference. Approximately 60% of the original points are retained after these preprocessing steps. Figure 3.1 summarizes the distribution of key features in the filtered shingles.

GTFS and GTFS-RT Features

All points in each shingle are spatially joined with the nearest stop locations in the static schedule corresponding to the day the data was collected. This provides the closest stop to each point of each shingle. The scheduled arrival time at the last point is used to calculate the scheduled travel time of the shingle. The distance to the closest stop is included as a feature to provide the models with information on how precise the scheduled travel time might be.

Features c are constructed which incorporate the speed, bearing, and observation age of other nearby bus trips in the prediction for an inference trip (Equation 3.1). Similar to the car forecast features used in some prior literature, this provides a realtime estimate of road conditions during inference. The observation age is the number of seconds between the nearby observation and the start of the inference trip. This is included since in some cases

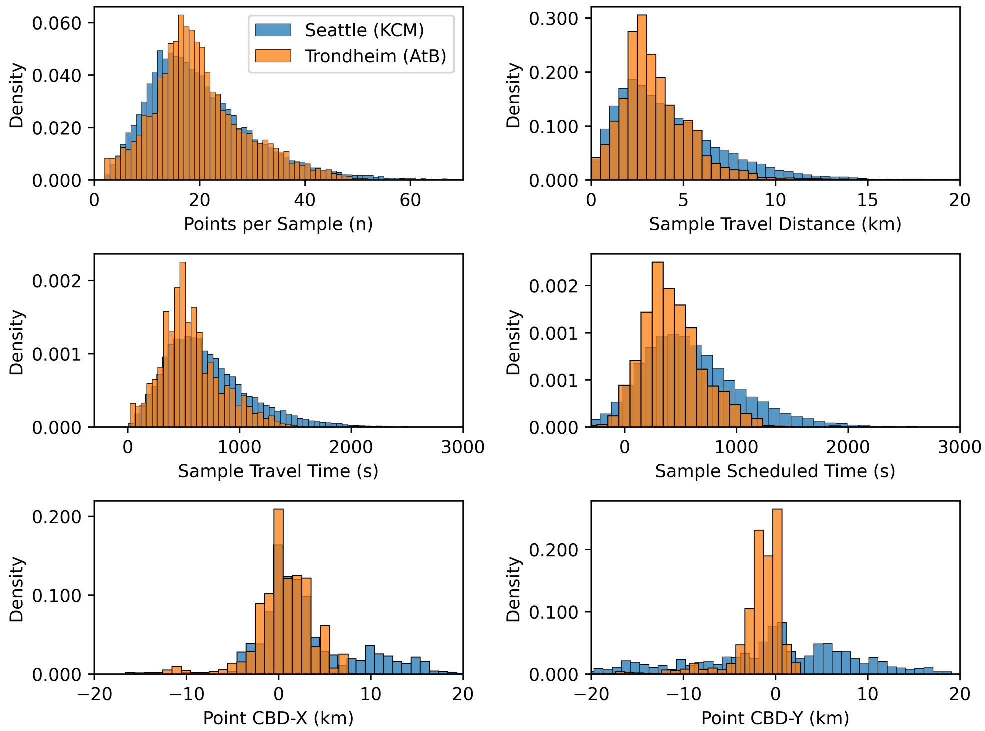

Figure 3.1: Distributions across shingle time and space. Point coordinates are shown relative to city CBD (central business district).

the most recent observation takes place long prior to the inference trip.

To build these features, each bus network was broken into a square grid with cells i, j having a resolution of 500 square meters. Then all training observations were sorted into these cells by time and location. The n most recent observations are selected from the grid cell $cell_{i,j}$ containing each point in the inference trip S of length K.

$$S_k = cell_{i,j} = \begin{Bmatrix} feat_{1,1} & feat_{1,2} & ... & feat_{1,c} \\ feat_{2,1} & ... & ... & ... \\ ... & ... & ... & ... \\ feat_{n,1} & ... & ... & feat_{n,c} \end{Bmatrix} \text{for k} \in \{1, 2, ..., K\} \tag{3.1}$$

These features are interpretable but may not capture all of the information the static schedule can provide. To attempt to improve on these hand-crafted features, embeddings were included from the GTFS feeds. To create embedding features for the GTFS static schedules the gtfs2vec method proposed by [84] was used. This approach trains an autoencoder to identify latent variables describing the spatial distribution of trip counts and stop sequences in a GTFS feed.

The autoencoder is fit using MSE. The autoencoder is trained on static feeds from the KCM and AtB networks, as well as feeds from 33 additional international bus networks using the SRAI library detailed in [83]. This library also generates hexagonal regions using the H3 library. Then the GTFS bounding box of each bus network is used to define the area to create hexagonal regions for. The points in each shingle are then matched to the region they fall within.

OSM Features

OSM data consist of geographical entities (nodes, ways, relations) covering the world. Each of these entities can have any number of attributes in the form of key-value tags. These data are submitted by volunteer users and are freely available. There are conventions but no enforced standards in the OSM tagging format.

Prior work on spatial representation of OSM data for deep learning has identified the most consistent tags and demonstrated their effectiveness in characterizing micro-regions of urban areas. They accomplished this using the "hex2vec" embedding model [180]. The hex2vec model is a modified spatial skip-gram fit with a negative sampling objective function. This means that it is trained to maximize the similarity of adjacent regions and minimize the similarity of randomly sampled non-adjacent regions. The dimensions used to measure similarity are latent variables consisting of information in their OSM tags.

In this work the pre-trained hex2vec model is used to generate embeddings for OSM tag data in each bus network. This captures information about the built environment along the trajectory of each shingle in a set of latent variables. The same regions as those used for the GTFS embedding features are used. Again points are matched to each shingle to the region they fall within.

3.4.3 Models

All training, testing, hyperparameter tuning and model experiments used k-fold cross validation with five folds to ensure that the models did not overfit the training data and that training reached stable solutions across runs. For each fold, models were trained on 80% of the training data. Models were then tested on data from a separate set of dates. This was done to ensure that no information from the testing dates was leaked when training the models.

During model selection five neural network model architectures and two baselines were tested on the first three months of data from the KCM and AtB networks:

- **Average Hourly Speeds (AVG):** Calculate the mean speed across all data during each hour of the day. Use this speed combined with total sample distance to estimate travel time.

- **Feedforward (FF):** A linear neural network [153]. Information about the shingle is passed forward through the layers. For features that occur per-observation point in a shingle (e.g., bearing) only the values from the first and last point are included as features. This prevents the need for re-sampling every shingle to fixed length.

- **Recurrent (GRU):** A GRU (gated recurrent unit)-based neural network proposed by [39]. Predictions are made per-point in a shingle and summed to get total travel time. Information from prior points can be saved and influence future points as learned by memory gates in the model. An LSTM-based recurrent model was also tested but no significant improvements were observed.

- **Convolutional (CONV):** A 1D-convolutional neural network proposed by [112]. A 1D kernel is passed over consecutive points in a shingle. Predictions are made per-point in a shingle and summed to get total travel time. Feature information from prior and future points can influence each point prediction.

- **Transformer Encoder (TRSF):** A transformer encoder neural network with self-attention and feedforward layers proposed by [169]. Predictions are made per-point in a shingle and summed to get total travel time. Each point in the shingle receives a positional encoding, then a query-key-value approach allows feature values at each point to attune differently to all other points.

- **DeepTTE (DEEPTTE):** A complex spatio-temporal neural network model, widely cited in prior literature [171]. During training predictions are made both per-point and per-shingle. During evaluation only the per-shingle estimates are used.

MAPE (mean absolute percentage error) was used to quantify model performance in all experiments (Equation3.2). MAPE is the average of the absolute percentage difference between actual y and predicted $\hat{y}$ total travel times across all samples n in the test data. This metric is commonly used in the related literature [16, 171]. It is favored for its interpretability but can fail when actual travel times are zero. It measures error relative to the length of the sample, which is valuable when comparing model errors across routes and networks.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y-\hat{y}}{y}\right| \tag{3.2}$$

DeepTTE (Deep Travel Time Estimation) was included as a benchmark. DeepTTE is a flexible, multi-city tested spatiotemporal data mining benchmark [171]. It uses a CNN (convolutional neural network) to extract spatial features from a trajectory, and feeds these convolutions with concatenated attribute (e.g., time of day) embeddings to a LSTM layer which models temporal dependence. During training, the outputs of the LSTM layer are used to predict the travel times at each point in the trajectory. During inference, the outputs of the LSTM layer are fed into a self-attention layer and concatenated again with the attribute embeddings. A multi-task learning weight is used to balance the importance of the predictions at individual points, and at the total trajectory.

These models were tested on the fourth month of data from each network (i.e., the model trained on the KCM network was tested separately on the fourth month of data from the KCM network, and from the AtB network). These models used only basic GPS trajectory features. The recurrent GRU model had the best combined performance on each source and target network without fine-tuning (Table 3.2).

Table 3.2: MAPE results for source and target cities.

Source City	KCM		AtB	
Target City	KCM	AtB	KCM	AtB
Feedforward (FF)	.23±.009	.44±.032	.70±.145	.27±.014
1D-Convolutional (CONV)	.20±.017	.24±.013	.27±.037	.15±.029
DeepTTE (DEEPTTE)	.14±.004	.28±.011	.31±.021	.11±.005
Transformer (TRSF)	.21±.002	.26±.015	.24±.007	.17±.002
Recurrent (GRU)	.18±.007	.22±.033	.26±.017	.12±.004

Models were also compared on the source city across other metrics such as RMSE (root mean squared error) and MAE (Table 3.3). Out of the standard models, the GRU had the best performance on the source network. The GRU model was selected for further hyperparameter tuning and experiments.

Hyperparameter tuning was performed on the GRU model with random parameter search and early stopping with a criterion of less than .001 MAPE improvement over three epochs. This tuning was performed using only basic GPS trajectory features. The Adam optimizer was used for all training [101]. Table 3.4 shows the range of tested hyperparameters, and those selected for the final model.

All training, testing, hyperparameter tuning and experimental tasks were carried out on a single NVIDIA 1660 Super GPU with 6GB of memory. The models were trained using

Table 3.3: Other metric results for source cities.

Source City	KCM		AtB	
Metric	RMSE	MAE	RMSE	MAE
Feedforward (FF)	17±.1	298±2	15±.1	213±2
1D-Convolutional (CONV)	10±.3	100±7	8±.7	70±11
DeepTTE (DEEPTTE)	10±.1	67±1	8±.1	47±2
Transformer (TRSF)	10±.1	102±1	9±.1	77±1
Recurrent (GRU)	10±.2	96±4	8±.1	58±1

Table 3.4: Hyperparameter testing.

Hyperparameter	Range Tested
Batch Size	128, 256, **512**, 1024
Dropout Rate	.05, .10, .20, **.40**
Hidden Size	**16**, 32, 64, 128
Hidden Layers	1, **2**, 3, 4
Adam_{α}	.0001, **.001**, .01, .10
$\text{Adam}_{\beta 1}$	.90, **.95**, .99
$\text{Adam}_{\beta 2}$	**.90**, .95, .99

version 2.2 of the PyTorch library.

3.4.4 Experimental Tasks

The first task for each model is to generalize within the bus network it was trained on (the source network). This is tested by holding out all training data from five hand-selected routes in the source network. The purpose of this test is to determine how well each model can forecast travel times under proposed scheduling or route changes. The routes are chosen at random during each fold of the k-fold cross validation.

The second task is for the model to generalize outside of the source bus network. This is tested by making predictions using the model trained on the source bus network on the target bus network. For KCM this is AtB and vice-versa. The purpose of this test is to determine how much information the model can generalize from the source network to the target network. Model performance was compared against a model trained on all the data collected from the target network.

Then an additional external generalization test was performed where each model is fine-tuned on a small subset of data from the target bus network. For fine-tuning model training was resumed for a short number of epochs with parameter weights from the existing pre-trained model. This test is used to determine how much additional data is needed to improve generalization performance to the same level as the model trained on all data from the target bus network. This is repeated on 33 additional international bus networks (Appendix .2). These networks were gathered from OpenMobilityData which indexes open transit data feeds [138]. All agencies listed in the US, Canada, Europe or Australia with both a GTFS and GTFS-RT feed were included. Feeds that required API keys were not included. For these additional networks, only a small sample of data from the GTFS-RT feed was used.

The third task for each model is an ablation test of different data sources. To perform the ablation, the model is tested with only GPS-based features and time embeddings. This test is repeated after concatenating each of the data sources to those base features. This is to test the benefit of incorporating specific open data sources on model generalizability and draw conclusions on which features are highest priority for a generalizable model.

3.5 Results and Discussion

3.5.1 Internal Network Generalization

First the performance of the selected models on the source network was compared to a set of holdout routes within the source network. Data from these holdout routes was not seen during training. Figure 3.2 compares the performance of each model on the holdout routes from the source network.

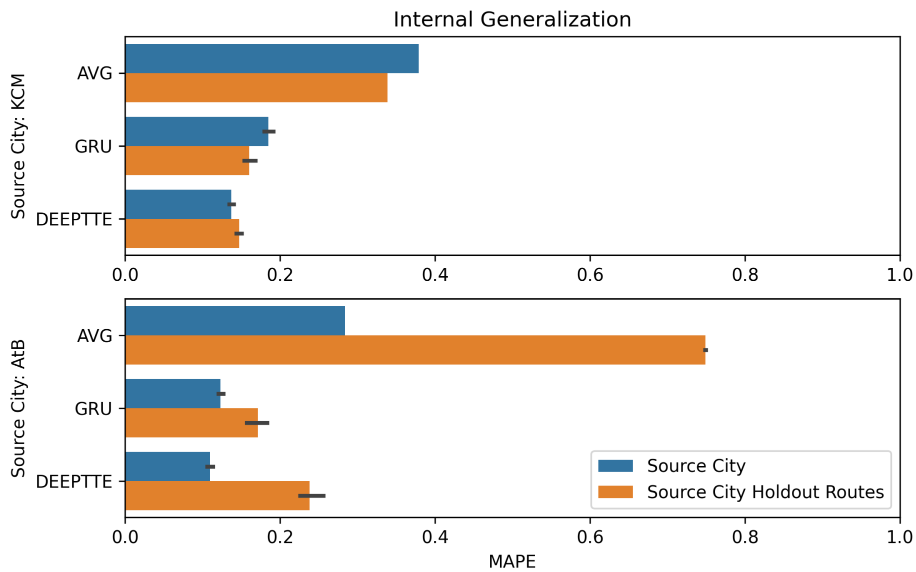

Figure 3.2: Internal generalization performance of each model on its source city.

As expected model performance on the set of holdout routes for each network is generally lower than those in the training data. However, despite having no information from these routes the deep learning models outperform the simple hourly average model in all cases. Between the deep learning approaches it was found that the simpler model performed better on the holdout routes, while DeepTTE performed better on the full source network.

This may be due to the smaller size of the training dataset compared to the original paper, in which the authors urge the usage of at least four million training samples to achieve satisfactory performance (this study includes approximately three million for KCM, and 500 thousand for AtB after filtering) [171]. It may also be due to the simpler model overfitting less on specific aspects of the non-holdout routes (DeepTTE has 10 times the GRU's parameters). This is supported by the hyperparameter tuning in which fewer parameters and higher dropout led to better generalization performance. This is also supported by the higher variance in model performance for AtB, which had fewer training samples than KCM.

Regardless, both deep learning approaches can adequately model travel times for unseen routes within the source network. A loss in performance of 1-20% MAPE can be expected depending on the original quantity of training data and complexity of the model.

3.5.2 External Network Generalization

Next the performance of the selected models on the source network is compared to all routes within the target network. This purpose of this test is to determine how much information the model can generalize to an entirely different target network. 3.3 compares the performance of each model on the target network.

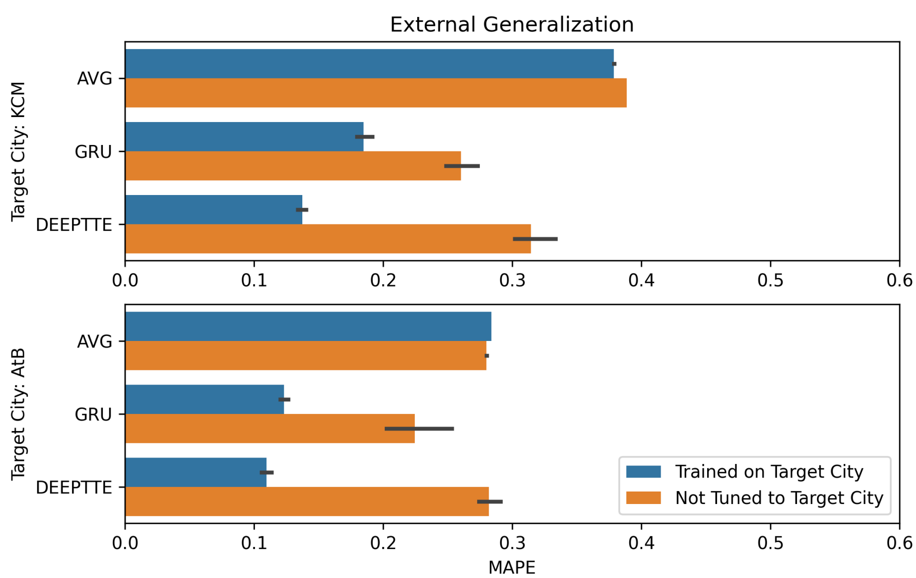

Figure 3.3: External generalization performance of each model on its source and target cities.

On average, a performance drop of 10-20% MAPE occurs when drawing predictions on a new city with no fine-tuning. The heuristic average model performs better on the AtB network, where average speeds are slightly slower (6.1m/s vs. 6.6 m/s) and have much less variance (3.6m/s vs. 4.7m/s). The more complex models did not significantly outperform the average speed model. In most cases this may not justify the use of more complex models.

Fine-tuning was then performed for 10 epochs with 100 samples from each target network. This led to an average net reduction of 5-10% MAPE. This means that in places where GTFS-RT data are available, a minimal collection effort can drastically improve generalization. Again, results were split based on the network. The models trained on AtB consistently improved by 2-6% MAPE. For the model trained on the larger network and greater initial samples, the tuning actually had a larger impact. This effect was limited to the GRU model. Along with the previous finding (Figure 3.3), this implies that the simpler GRU model is more capable of tuning to new networks. In both cases, it is better to begin with the GRU trained on the larger network. Without tuning the GRU has lower loss, and with tuning on 100 samples it has a greater response on the target network. On the other hand, DeepTTE performs better when tested on either source network only (no tuning, no generalization).

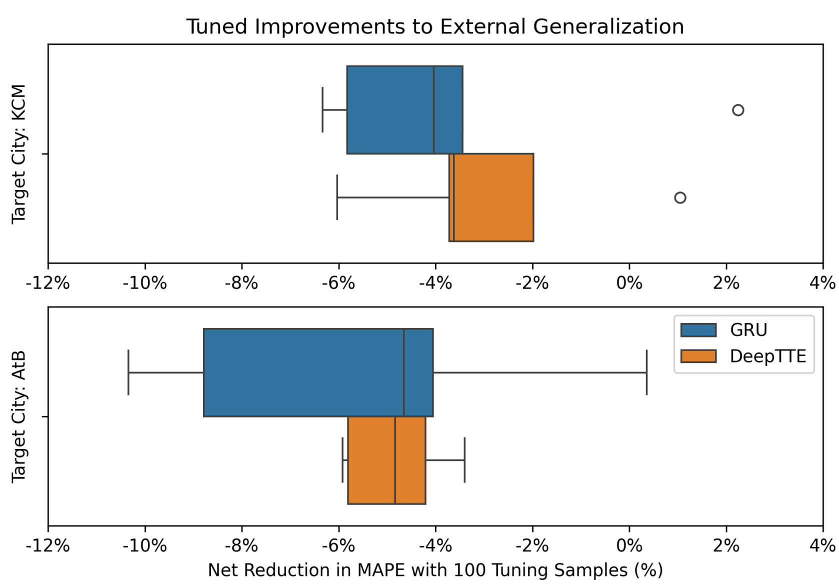

Figure 3.4: External generalization improvements for each model tuned to the target city.

To further explore the difference in model performance across cities, data from 33 additional bus networks was collected and tested with a model trained on the data from both KCM and AtB on those networks. Figure 3.5 shows these results.

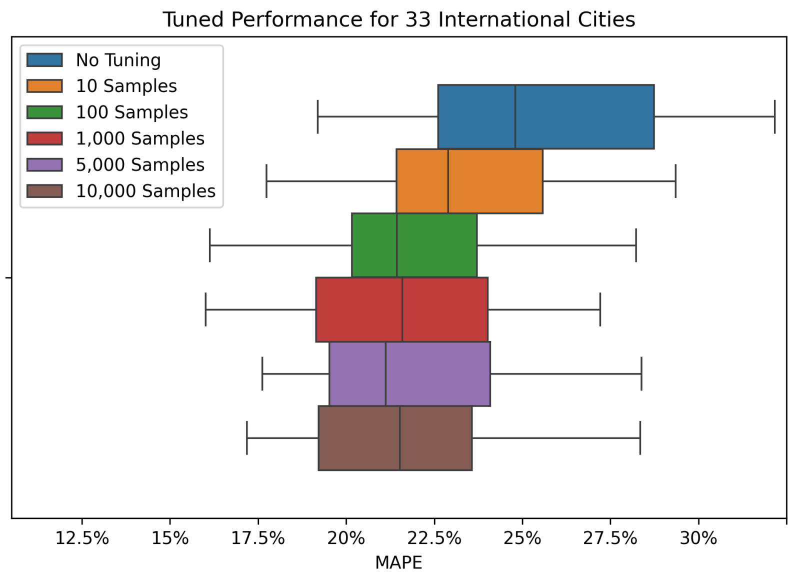

Figure 3.5: External generalization performance of each model on 33 different target cities.

There are diminishing returns from including more than 100 tuning samples. In this case the initial model was trained on a large number of samples. The shift in performance from no tuning 100 samples falls at approximately 5% MAPE. This corroborates the previous finding for models trained on the KCM network and tuned using data from AtB (Figures 3.3 and 3.4). The implication of this finding is that an adequately large sample for an initial model can be fine-tuned with as few as 100 additional samples from a target network to achieve higher performance than a heuristic average. Overall performance after tuning varies by 10% MAPE across the different networks. This means that regardless of tuning there is high variation in model performance depending on the characteristics of each individual bus network.

3.5.3 Feature Ablation

To perform the ablation, each model was first tested with only GPS-based features and time embeddings. Then each additional data source was separately tested. Figure 3.5 compares the performance of each model with additional features to those of the source network.

Table 3.5: Feature ablation results using GRU model.

Source City	KCM		AtB	
Target City	KCM	AtB	KCM	AtB
GPS	.18±.007	.22±.033	.26±.017	.12±.004
GPS+GTFS Embeddings	.18±.010	.15±.020	.25±.015	.12±.004
GPS+OSM Embeddings	.18±.007	.19±.030	.25±.013	.15±.027
GPS+Realtime Features	.21±.012	.26±.050	.24±.009	.15±.030
GPS+Static Features	.19±.006	.22±.034	.25±.020	.13±.019

First consider the models tested on each source network (outer columns of Table 3.5). These results are for baseline models with no tuning. They show that none of the additional features were able to significantly improve prediction in the source network over the baseline GPS features. However, the inner columns (models tested on the opposite network) reveal significant improvements for some models. On the smaller network, the OSM and GTFS embeddings improved the external generalization of the model by up to 7%. On the larger network, the finding was much less pronounced. MAPE improved by only 1-2%.

The implication of this finding is that it is worthwhile to train on the additional features even if they don't improve performance on the base network. When generalizing with no fine-tuning data, OSM or GTFS embeddings can improve estimates to nearly the same level as a baseline model. They may provide an additional source of information in the target network to improve generalization estimates without fine-tuning, though this same information may be provided by only the GPS data in the source network.

Note that the GTFS embeddings contained latent variables built on the count and direction of trips in a given region. In future work, the autoencoder used to find the GTFS embeddings might be trained on input features that also relate to the scheduled trip times, to improve inference related to travel time. Regardless, this was not the original goal of [84] (the embeddings as designed and used here are for a task in classifying transit accessibility).

3.6 Conclusions

In this work, bus travel time forecasting models were tested on several feature sets and generalization tasks. Many deep learning-based bus travel time forecast models rely on closed source features (e.g., road traffic forecasts) or are tested on a very limited set of routes/data from their respective transit network. To address this gap, five neural network model architectures and one baseline heuristic were tested. All data was gathered from open and standardized sources such as GTFS, GTFS-RT and OSM. Internal generalization was tested (does a model generalize to unseen routes in the same system?) as well as external generalization (does a model generalize to a different bus networks?). Model generalization was improved through fine-tuning.

Our results suggest that deep learning methods generalize well within the source network. Depending on the complexity of the model, and the size of the initial training dataset the losses for holdout routes unseen in the training data were as little as 1% MAPE. When tested on different cities, the models failed to significantly improve over heuristics. However, using as few as 100 samples from the target network, the fine-tuned model improved by 5-10% MAPE. Incorporating travel time features from a static schedule, or embeddings constructed on OSM tags for the target network significantly improved model generalization. The improvements were more pronounced when the source was a larger city with more initial training data, though the relationships identified went both ways.

There are several implications of these results for practice. First, the internal generalization tests demonstrate the viability of geospatial data mining for route- and trip-planning purposes in the source network. Though training with 3-4 million samples gives better results, a less complex model with fewer samples can still greatly outperform heuristics for this use-case. Second, the external generalization tests reveal the need for fine-tuning in order to generalize model forecasts to new networks. This means that when applying these methods across bus networks, the target city must provide a GTFS-RT feed, or AVL data must be requested on an individual basis. The quantity of data required is minimal (i.e., collectable in less than a day for nearly any network). Third, the results of the ablation study show that features of the built environment, existing schedule information, or road forecasts estimated from GTFS-RT improve generalization to the target network. If using these data or methods for cross-sectional analysis of bus networks, a GTFS feed from the target network can greatly improve estimates, if GTFS is unavailable OSM data provides nearly similar improvements. However, these additional data features only marginally improved performance on the source network.

Though this work began to develop a generalizable model, there is still opportunity for more internal and external network testing. For example, testing a more specific or varied set of holdout routes including locations that have many or relatively few training samples, or training on a larger dataset collected over several months from many different realtime feeds. Improving the static schedule embeddings by incorporating travel time features may also lead to model improvements. As the data size or number of systems increases, complex models will likely begin to perform better, necessitating more creative combinations of the architectures tested here.

Chapter 4

EMPOWERING ELECTRIC BUS DEPLOYMENT WITH STANDARDIZED TRANSIT DATA

4.1 Abstract

This study evaluates the accuracy vehicle energy consumption estimates built with GPS data having low sampling frequencies as collected from standardized bus transit feeds. It applies bus travel time forecast models to estimate trip-specific drive cycles for buses in any network reporting standardized data. Validation with high sampling frequency data reveals challenges for modeling vehicle drive cycles. A block-level energy model is then applied to a cross-sectional analysis of 40 international bus networks to examine energy and power constraints for implementing battery electric buses. Prior work on this topic has focused on aggregating standardized bus GPS data directly to route segments. This study instead uses trained models to infer speed profiles for routes and networks without pre-collected data. This allows a much wider-reaching comparison of agency electrification needs. The sensitivity of energy consumption estimates and blocking constraints is tested against key operational factors. Conclusions are then drawn on the adequacy of standardized bus data for energy analysis, and more practical findings related to electrification across agencies. Findings suggest that the lost information in low sampling frequency feeds does not significantly impact energy consumption estimates for battery electric buses. Current battery and charging technology is capable of supporting initial rollouts on low-energy blocks, but incapable of supporting full electrification. Furthermore, initial rollouts can often be entirely supported with unmanaged depot charging. However to mitigate growing peak power costs and meet block energy needs, it is essential for agencies to adopt managed charging strategies under a full electrification scenario.

4.2 Introduction

4.2.1 Motivation

Open source data and tools are useful for novel workflows that do not have industry established toolsets. The collaborative development process that they encourage helps in sharing knowledge for research-based tasks. For example, in electrification of bus networks. There are several benefits to bus electrification. Foremost, it reduces operating costs for maintenance and fuel [1]. This is a key benefit given the operations funding shortfalls currently facing transit agencies [69]. From a GHG (greenhouse gas) emissions standpoint, electrification of bus fleets is less important than personal vehicles, as they only account for 1.3% of annual emissions [8]. However, diesel engines used in many fleets produce high concentrations of unhealthy CAP (criteria air pollutants) [19, 126]. Reducing noise pollution from diesel buses also improves quality of life for residents residing near transit corridors, and improves passenger experience riding transit [130, 104].

To realize these benefits there are risks. The capital investment to transition to electric transit includes purchasing vehicles that are currently more expensive than diesel counterparts [1], though that is expected to change in the near future as battery manufacturing improves and supply chains stabilize [109]. It also requires investment in charging infrastructure and facilities to maintain electric buses. There are also personnel costs for training or hiring workers skilled in operating and maintaining electric buses. All told, there is a significant capital expenditure to reach full fleet electrification, despite cost savings over the life of each vehicle.

Thus, fleet electrification projects currently require federal funding to be cost-effective. When funding through programs such as the Low or No Emission Vehicle Program [31, 32] cover 80% of vehicle purchasing costs, BEBs have a lifecycle cost up to 23% lower than diesel or compressed natural gas counterparts [154]. This contingency on federal funding has made lessons learned from bus electrification projects isolated. This is further compounded by the number of variables regarding vehicle operation, charging strategy and interaction with utilities' power distribution infrastructure. There is simply very little real-world knowledge on designing and operating BEB fleets relative to the variety of possible implementations. As of 2023, only 2% of all bus transit was electric [79] and 44% of transit agencies could not envision a path to full fleet electrification [79]. This creates an opportunity for open source, generalizable bus fleet electrification planning tools that can be used by agencies to make informed decisions on their transition to BEBs.

4.2.2 Outline of This Work

This study addresses agency uncertainties surrounding infrastructure needs for fleet electrification. It first examines the accuracy of energy consumption estimates from low resolution open data to a set of validation data collected with a phone and GNSS (global navigation satellite system) receiver. It then applies the generative models developed in Chapter 3 to estimate standardized route cycles built on GTFS-RT data. These use standardized bus data sources such that the analysis is replicable for any agency or location, which is demonstrated by extending the analysis to 40 bus networks. The sensitivity of energy consumption estimates and design implications to key factors such as passenger load, temperature and driver behavior are tested across all networks. Conclusions are then drawn on the adequacy of tools built on open standardized transit data for predictive energy modeling and practical findings for transit agencies transitioning to BEBs.

4.3 Literature Review

4.3.1 Battery Electric Bus Technology

The primary components introduced in a BEB system are batteries and chargers. Vehicle chassis, maintenance equipment, and other components may be adapted to suit BEBs, but do not largely inform the system design. The design aspects of the battery are available capacity and maximum supported charging rate. The design aspects of the charger are location, connection type, and maximum provided charging rate [1].

BEB specific charger connection types include infrastructure-mounted cross rail, vehicle-mounted pantograph, enclosed pin/socket, and wireless inductive connections [135, 136]. They also include plug-in connections used for light duty vehicles such as SAE J1772. Charging power ranges widely from 50-1,000 kW. Each connection type supports a range of charging rates; cross rail and pantograph tend to offer the highest, with plug-in and inductive charging generally providing slower charging speeds [136]. Generally, chargers must be connected to the grid to provide the required power, though it is possible to have off-grid power storage [123, 96]. Chargers may be located at the ends of routes or blocks, at individual bus stops, or at a central depot. Different connection types offer advantages and disadvantages at each type of location (e.g., plug-in charging is impractical at individual stops). Each of these design parameters must be weighed against costs and charging strategy to determine the optimal charging system design.

Battery capacity comes with a trade off to weight. Unlike with diesel fuel, a heavy weight penalty is inflicted on energy consumption when carrying bigger batteries to support a larger range. Thus, battery capacity in excess of route needs is a hindrance. Batteries may support different charging rates, but higher rates generally lead to shorter battery lifetimes [179]. Extreme SOC (state of charge) and environmental conditions also affect lithium-ion battery lifetimes. Faster charging rates support opportunistic charging, allowing for smaller batteries to serve longer routes. Slower rates may work better with overnight depot charging. When designing charging strategy and battery sizes, an important consideration is avoiding peak-demand charging costs [1]. Battery design parameters must meet the needs of the route based on selected charging strategy.

There are several key factors that affect bus battery consumption rate on-route. Low ambient temperatures decrease energy efficiency and capacity of lithium-ion batteries, manifesting as decreased vehicle range [122]. High HVAC (heating, ventilation, and air conditioning) loads have a significant consumption impact. Stopping, idling and accelerating (e.g., at bus stops) can lead to inefficient engine operating zones, though more so for an internal combustion motor than for an AC (alternating current) motor. Acceleration also generally consumes more energy than can be recovered via regenerative braking. Slower trips with more idling consume more energy per mile due to the higher relative influence of auxiliary loads. The topography of a bus route also affects the amount of energy required to run the route (i.e., uphill routes require higher power and thus more energy to complete). The passenger load onboard the bus also has interplay with these losses. An electric bus with greater passenger loads will require more energy to travel uphill, but will also regain more energy from regenerative braking [117]. Overall speed greatly influences energy consumption, as more energy is lost to air and rolling resistances. These factors, along with heat loss and other motor inefficiencies all contribute to the rate of energy consumption for a BEB.

The total consumed energy is then a combination of energy efficiency and power demand over time, which can be measured using a vehicle drive cycle. The vehicle drive cycle or operating profile is a map of velocity over time, which is representative of typical driving conditions for a design vehicle. This provides an estimate of route energy consumption from which one can begin to devise a charging strategy and fleet electrification plan. For bus transit routes, an operating profile should be developed per-route [1] and include metrics such as average route speed, worst case route speed, distance from depot to route start, layover times, average grade, and average passenger load [1]. These metrics can be used to estimate energy consumption for a given route, and to determine the optimal battery size and charging strategy for a given route. As will be demonstrated in this work, these metrics can be estimated per-trip using open source data and tools, and forecasted for new routes.

4.3.2 Modeling Power Consumption From GPS Data

There are many approaches to developing vehicle drive cycles. Foremost, there are standardized cycles for light duty vehicles as well as buses and other vehicles [7]. These are collected using onboard vehicle speed sensors across a variety of driving styles and contexts. Main criticisms of standardized drive cycles are that they over-generalize and do not closely replicate real-world driving conditions or topography [37]. This is a particular challenge for agencies switching to BEBs, as the characteristics of individual routes and even individual drivers may significantly impact the battery and charging design for the route and/or system. Many drive cycles for buses (and general tests of vehicle performance) are collected at the Altoona testing facility [95].

GPS data has been previously used to develop bus drive cycles and estimate vehicle emissions under typical conditions for specific cities or regions [105, 108]. In these studies, onboard collection is employed to gather large quantities of drive cycle data, which is then classified according to typical times of day, route types (e.g., inter-city, within-city, peak/off hour), driving behavior, and more [163, 145]. The exact classification criteria varies by study. In some cases, it is combined with topography data from a DEM (digital elevation model) to provide a better estimate of energy consumption across specific routes [9, 119]. In the absence of real passenger loading data, probabilistic methods can also be used to estimate vehicle loads during given trip segments [9].

There are fewer studies that use open data for bus drive cycle development. The main challenge with using open data sources such as GTFS-RT for drive cycle development is that they are relatively low resolution. This makes it more challenging to distinguish acceleration and deceleration events in the drive cycle. Acceleration generally uses more energy than can be recovered through regenerative braking. Thus low resolution data will tend to underestimate energy consumption. Even AVL data is not as high of resolution as the onboard sensors can provide. In one study AVL was found to provide energy estimates 80-90% as accurate as those developed from onboard collected bus GPS data [2].

There is one prior work which used a combination of GTFS and GTFS-RT data to develop system-wide electric bus drive cycles [178]. This study resampled the coordinates of static route shapes, then map-matched collected speed observations from GTFS-RT data to said coordinates. This gave estimates of the distribution of speed for vehicles traversing each segment, similar to the analysis used in Chapter 2 of this dissertation. Their method was calibrated against phone collected GPS data, and against a synthetic drive cycle generated from pure GTFS with scheduled target speeds and probabilistic interactions with stops and intersections. Their analysis spans two routes from the Victoria Regional Transit System.

Estimating energy consumption from a given drive cycle depends on the vehicle's physics-based power consumption model. This is a function of the vehicle's powertrain, which is a combination of the engine, transmission, and other physical components. Given a speed profile, sophisticated energy models such as NREL's FASTSim (Future Automotive Systems Technology Simulator) can be used to estimate energy consumption [27, 103]. However, [178] used a slightly simpler physics-based model that was originally built as part of a different prior work [68]. This model accounts for gravity, acceleration, air resistance, rolling resistance and assumed static loads (e.g., HVAC) on the energy consumption. There is one other notable study which developed a set of open source tools for estimating bus energy consumption from GTFS data [51]. That work focused on applying standardized drive cycles to static GTFS data and exploring system costs and component degradation over time. It is possibly the best example of generalizable open software in this space, given that it can be applied to any GTFS feed and is available as a python package.

4.3.3 Contribution of This Work

This study contributes to the field of tools for bus transit design and evaluation built on standardized open data. It does so by evaluating the energy estimates developed from forecasted GTFS-RT drive cycles validated with phone and GNSS receiver data. This comparison sheds light on the reliability and effectiveness of using standardized open data for predictive energy modeling. It points to potential benefits and drawbacks of the GTFS-RT standard for energy modeling.

This study also contributes to the field of bus transit electrification by applying generalizable travel time models to construct data-informed drive cycles for new bus routes and networks. Prior work in this area has focused on standard cycles for static feeds, or on developing cycles for routes with pre-existing realtime data. By using generalizable models to conduct analysis across 40 networks, this study provides cross-sectional insights on the impact of key factors such as temperature, passenger load, and driver behavior on energy consumption and design implications for transit agencies considering BEB adoption.

These goals are stated in two research questions:

1. How accurately can bus energy consumption be predicted using a combination of 3D route geometry and models trained on standardized open bus data? What are the implications for findings on models built using open data?

2. How do blocking and charging constraints vary across agencies and with sensitivity to key operational parameters? What are the implications for BEB technology growth?

4.4 Methods

4.4.1 Drive Cycle Energy Estimation

A bus drive cycle is a map of velocity over time which is representative of typical driving conditions for a design route. A drive cycle can be combined with a vehicle energy model to estimate the typical power profile and energy required to drive said design route. Fuel consumption is driven by aerodynamic, rolling resistance, gravity and acceleration forces. These forces, auxiliary loads and powertrain component efficiencies determine the total energy required for a design vehicle to carry out a specified drive cycle. Energy consumed can be converted to fuel consumption and aggregated across routes to estimate network power needs and environmental impacts.

The NREL (National Renewable Energy Laboratory) FASTSim model is a combined vehicle and energy model. It is validated against the EPA (Environmental Protection Agency)-tested fuel consumption values of 700 vehicles, falling within 5-10% in all cases [27]. Given a design vehicle and drive cycle it can estimate power over time and total energy consumption for the cycle. FASTSim has a built-in vehicle library with 62 vehicles. These include light-, medium- and heavy-duty vehicles as well as a mix of ICEV (internal combustion engine vehicle)s, PHEV (plug-in hybrid electric vehicle)s, HEV (hybrid electric vehicle)s and BEV (battery electric vehicle)s. In this study only the FASTSim energy estimation framework is used, but it can also estimate vehicle component lifetimes and total cost of ownership.

One additional benefit of the FASTSim model is that it can estimate the impact of road grade on energy consumption. Road grade plays an important role in the required force for acceleration and in the amount of energy recovered by regenerative braking when decelerating. The elevation data used in this study is collected from a combination of the USGS 3DEP (United States Geological Survey 3D Elevation Program) DEM (10m resolution), the EU DEM (30m resolution) and the AW3D (ALOS World 3D) DEM (30m resolution) [90, 75, 139]. The higher resolution USGS 3DEP is used as a first priority, with the others filling in where it is unavailable.

4.4.2 BEB Vehicle Design Parameters

The design vehicle for this study was based on the 2022 New Flyer XE40 BEB. The FASTSim vehicle library does not include any medium- or heavy-duty BEVs. Therefore the parameters for the design vehicle were gathered from several sources. The first and highest priority source was the Altoona bus testing report for the 2022 New Flyer XE40 [95]. For geometric parameters not found in the report, the second source was heavy-duty diesel vehicles in the FASTSim vehicle library. The last and lowest priority source were other BEVs in the FASTSim library. These were used for parameters related to the engine map and drivetrain component efficiencies. Table 4.1 shows each parameter, its source and the value used in the model.

The MC (Motor Controller) parameters describe aspects of the AC motor used for propulsion. It is analogous to an ICE (internal combustion engine). The ESS (energy storage system) is the onboard battery used to drive the AC motor. It is analogous to a fuel tank. The bus has two axles with dual rear wheels giving six total wheels. The weight parameter includes the curb weight of the vehicle with an additional 150 lbs for the driver and each seated passenger (41 total). The auxiliary power includes heating at half-capacity and no air conditioning (the same setup as used in the Altoona test cycle [95]). The motor efficiency map (required by FASTSim) was also taken from the existing FASTSim BEVs.

The Altoona testing procedure includes measured energy consumption (at the ESS) from

Table 4.1: BEB Vehicle Parameters Used in FASTSim Based on 2022 New Flyer XE40

Vehicle Parameter	**Value**	**Source**
Drag Coefficient	0.60	FASTSim Class 8 Truck
Frontal Area	10.78 m^2	Altoona Testing Report
Center of Gravity	0.53 m	FASTSim Class 8 Truck
Drive Axle Weight	64 %	Altoona Testing Report
Wheelbase	7.2 m	Altoona Testing Report
Weight	18,230 kg	Altoona Testing Report
MC Max Power	230 kW	Altoona Testing Report
MC Time to Max Power	4 s	FASTSim BEVs
ESS Max Power	1,000 kW	FASTSim BEVs
ESS Max Energy	466 kWh	Altoona Testing Report
ESS Round Trip Efficiency	0.97	FASTSim BEVs
Wheel Inertia	0.82 kg/m^2	FASTSim Class 8 Truck
Number of Wheels	6	Altoona Testing Report
Wheel Rolling Resistance Coefficient	0.006	FASTSim Class 8 Truck
Wheel Radius	0.51 m	FASTSim Class 8 Truck
Wheel Coefficient of Friction	0.70	FASTSim Class 8 Truck
Min State of Charge	10 %	FASTSim BEVs
Max State of Charge	95 %	FASTSim BEVs
Alternator Efficiency	100 %	FASTSim BEVs
Auxiliary Power	10.2 kW	Altoona Testing Report
Transmission Efficiency	98 %	FASTSim BEVs
Max Braking Regeneration	98 %	FASTSim BEVs
Time 0-60 MPH	60 s	Altoona Testing Report

standardized drive cycles. The drive cycles included for the New Flyer XE40 were the EPA HD UDDS (Heavy-Duty Urban Dynamometer Driving Schedule) cycle, the Manhattan Cycle and the Orange County Bus Cycle. The tests were performed at seated weight with heat at half-capacity and no air conditioning. These cycles were used to validate the energy consumption of the bus parameters in Table 4.1. Table 4.2 shows the energy consumption estimated by FASTSim of the BEB vehicle model on each of the standardized cycles.

Table 4.2: Design Vehicle Energy Consumption for Standardized BEB Drive Cycles

Drive Cycle	**Reported Consumption**	**Modeled Consumption**
Manhattan	2.75 kWh/mi	2.77 kWh/mi
Orange County	2.00 kWh/mi	2.18 kWh/mi
HD-UDDS	1.99 kWh/mi	1.98 kWh/mi

These results show that the BEB vehicle model is within 10% of the reported energy consumption for each of the standardized drive cycles. It is within 1% of the reported energy consumption for the Manhattan and HD-UDDS cycles.

4.4.3 Validating GTFS-RT Cycles with High Precision GNSS Receiver Data

One of the main challenges with using open standardized bus data for modeling energy consumption is the low resolution of these data. Drive cycles are typically reported at one Hz, whereas a GTFS-RT feed's update frequency can vary depending on the bus network and vehicle (typically 1/30 Hz). This study aims to address this challenge by collecting one Hz phone and GNSS receiver data from onboard active bus trips. Using the vehicle ID painted outside the vehicle, these high resolution data are matched to reported locations in the low resolution GTFS-RT data. Using FASTSim, energy estimates from drive cycles built on these data can be compared to those built on GTFS-RT data for the same trip.

Phone data from 12 trips between September 2022 and November 2022 were collected from the KCM bus network. Phone data and GNSS receiver data were also collected from three trips on March 12, 2024. The phone data was collected using an Apple iPhone 11. The GNSS receiver data was collected using an Emlid Reach RS2. Realtime kinematic corrections for the GNSS receiver data were gathered through NTRIP from the WSRN (Washington State Reference Network). The masking angle of the GNSS receiver was set to a 20° elevation angle. Both the phone and GNSS receiver were mounted to a cradle (Figure 4.1) and carried level at the center of the bus at approximately window-height.

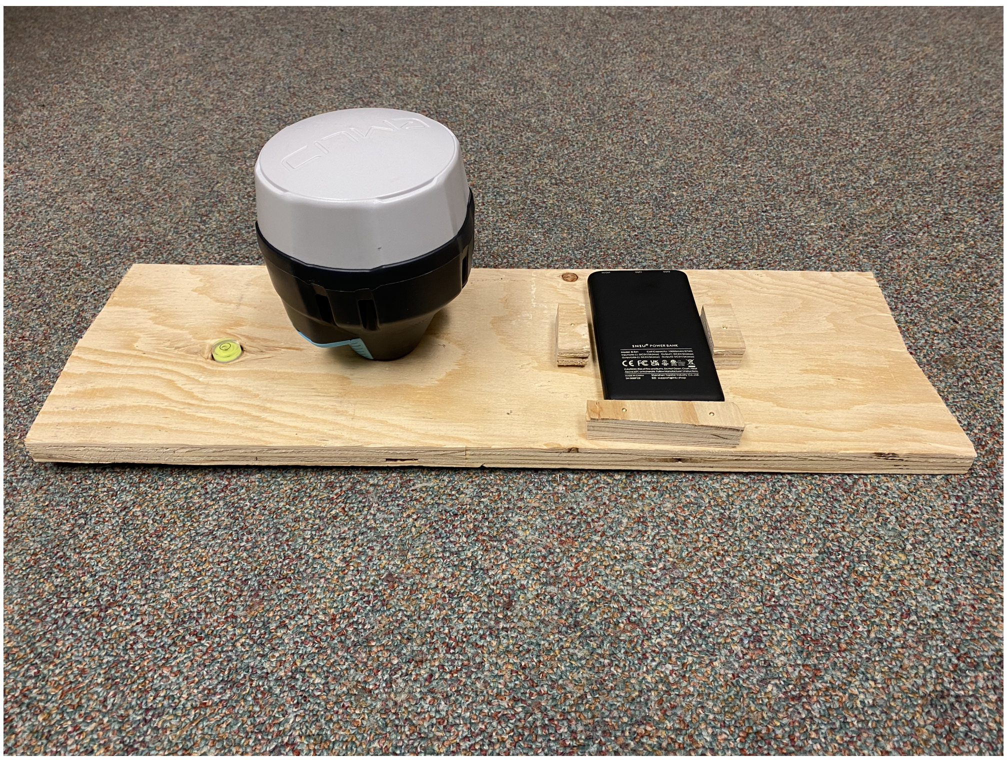

Figure 4.1: Onboard validation data collection cradle for GNSS receiver and phone data.

4.4.4 Estimating Block Energy Consumption with GTFS and GTFS-RT

A basic GTFS feed describes a set of stop times and locations for all scheduled trips in the bus network. A "trip" denotes a single run of a given route. Each trip is carried out by a single vehicle. A "block" denotes a series of trips which are scheduled to be carried out consecutively on a given service day by a single vehicle. To estimate daily bus network energy consumption, the energy consumed on all blocks is combined according to Equation 4.1.

$$DailyNetworkEnergy = \sum_{i=1}^{n} \left[\left(\sum_{j=1}^{m} \hat{C_{ij}} d_{ij} + TQp_{ij} + Pl_{ij} + C_l s_{ij} \right) + C_l(U_i + V_i) \right] \tag{4.1}$$

$$Q = (t_i - T_o)av\rho c \cdot \frac{3,600s}{1hr} \cdot \frac{1kW}{3412BTU/hr} \tag{4.2}$$

where:

- n is the number of blocks scheduled on the service date.
- m is the number of trips in block i.
- $\hat{C_{ij}}$ is the estimated drive cycle energy consumption (kWh/mi) of trip j in block i.
- d_{ij} is the total distance (mi) of trip j in block i.
- T is the assumed door open/close time (hr).
- Q is the assumed door heat energy flux (kW) from Equation 4.2 limited to the volume of air in the cabin.
- p_{ij} is the number of stops on trip j in block i.
- P is the assumed auxiliary power (kW) used during trip layovers.
- l_{ij} is the layover time (hr) between the end of trip j and the start of trip $j+1$ in block i.
- C_l is the assumed deadhead energy consumption (kWh/mi).

- s_{ij} is the layover Manhattan distance (mi) between the end of trip j and the start of trip $j+1$ in block i.

- U_i is the Manhattan distance (mi) between the depot and the start of block i.

- V_i is the Manhattan distance (mi) between the depot and the end of block i.

and:

- t_i is the temperature inside the bus cabin ($65F$).

- T_o is the assumed temperature outside the bus (F).

- a is the area of the bus door ($32ft^2$).

- v is the wind speed through the door ($0.5ft/s$).

- ρ is the density of air at sea level ($0.0765lb/ft^3$).

- c is the specific heat of air at sea level ($0.241BTU/lb$-F).

Trip and layover distances and times can easily be calculated from scheduled stop locations and times in a GTFS feed. Depot locations are approximated using k-means clustering of all block start locations. This minimizes the total block deadhead distance driven to the start of all blocks. The baseline number of depots is based on the density of depots in the KCM network. The KCM network contains seven depots covering 2,135 square miles, or .003 depots per square mile [35, 55]. A minimum of one depot is imposed for all networks. Since the driver may occupy the vehicle during layovers, the auxiliary power used between trips is assumed to be the same as during trips (Table 4.1). The baseline energy consumption used to travel between trips is assumed to be the FASTSim-estimated Manhattan drive cycle energy consumption for the design vehicle (2.77 kWh/mi).

The remaining challenge and focus of this study is to estimate $\hat{C}$ (i.e., the in-service vehicle energy consumption in kWh/mi) on a trip-by-trip basis for any bus network. First, an aggregation methodology proposed in prior work is tested on each route in the KCM network [178]. This method breaks each route into 100 m uniform-distance segments then aggregates observed speeds collected from a GTFS-RT feed to their nearest segment. The speed profile is determined by the mean speed of the observations assigned to each segment. Since the optional "shapes.txt" component is not available for all transit networks, the route shape was determined by the stop locations.

This requires pre-collected GTFS-RT or AVL data to provide speeds for the routes and times of interest. Using this same pre-collected data, travel time models can be developed and used to predict speed profiles on new routes. This allows the analysis to be extended to new bus networks without GTFS-RT feeds, or without archived data to aggregate. This study applies the generalizable travel time models built in Chapter 3 to this task. Note that The irreducible error in these models was found to range from 9-15%. An additional error of 3-10% was incurred when generalizing to bus networks not in the original training set.

The travel time models were applied to the same 100 m uniform-distance trip segments as used in the aggregation method. However, rather than aggregate GTFS-RT observations, the models were used to infer segment-by-segment travel times for each trip in the target GTFS feed. Using the static distances (100 m) and the inferred travel times, a drive cycle of speed over time was created for every trip. The time of day embedding for each trip (an input to the travel time models) was based on the initial stop time reported in the GTFS feed. This makes the estimated drive cycles sensitive to varying traffic conditions. For both drive cycle-construction methods, a Savitzky-Golay filter was applied to smooth the velocity profile [178, 51, 16]. FASTSim was then used to calculate the estimated energy consumption ($\hat{C}$) for each drive cycle.

4.4.5 Sensitivity Analysis

The energy consumption for all blocks in each of 40 international bus networks was estimated using the predicted drive cycle method. A sensitivity analysis is performed on the assumed parameters in Equations 4.1 and 4.2 (T, P, C_l, U_i, V_i, T_o) to determine the impact of various operational treatments on BEB viability across these networks (Table 4.3).

Table 4.3: Sensitivity Parameters Affecting Block Energy Consumption

Treatment	**Baseline**	**Tested Range**	**Parameters Affected**
Acc./Dec. Aggressiveness	1.00	[0.1, 5.0]	$\hat{C}_{ij}$
Passenger Load	41 $persons$	[1, 82] $persons$	$\hat{C}_{ij}$
Auxiliary Power	20 kW	[0, 40] kW	$\hat{C}_{ij}$, P
Deadhead Consumption	2.77 kWh/mi	[2, 5] kWh/mi	C_l
Depot Density	.003 $n/sqmi$	[.001, .006] $n/sqmi$	U_i, V_i
Door Open Time	30 s	[10, 120] s	T
Outside Temperature	46 F	[20, 90] F	T_o
Depot Plug Power	50 kW	[50, 350] kW	-

The sensitivity analysis is performed by varying each parameter in Table 4.3 by the range specified. The energy consumption for all blocks in each network is then re-estimated using the new parameter values. The results are then aggregated across all networks to determine the impact of each parameter on energy consumption, and the viability of BEB technology across all networks.

The acceleration/deceleration aggressiveness parameter is modeled by applying a weighted moving average filter to the drive cycle which boosts or decreases the amplitude of the drive cycle speeds. For each window, the difference of the signal from the window-mean (S_{y_i}) is boosted by a scalar B (Equations 4.3 and 4.4). Figure 4.2 shows how different values of B affect the drive cycle speed and overall magnitude of acceleration and deceleration events.

$$\hat{y}_i = S_{y_i} * B \tag{4.3}$$

$$S_{y_i} = (y_i - \frac{1}{2w}\sum_{i-w}^{i+w} y_i) \tag{4.4}$$

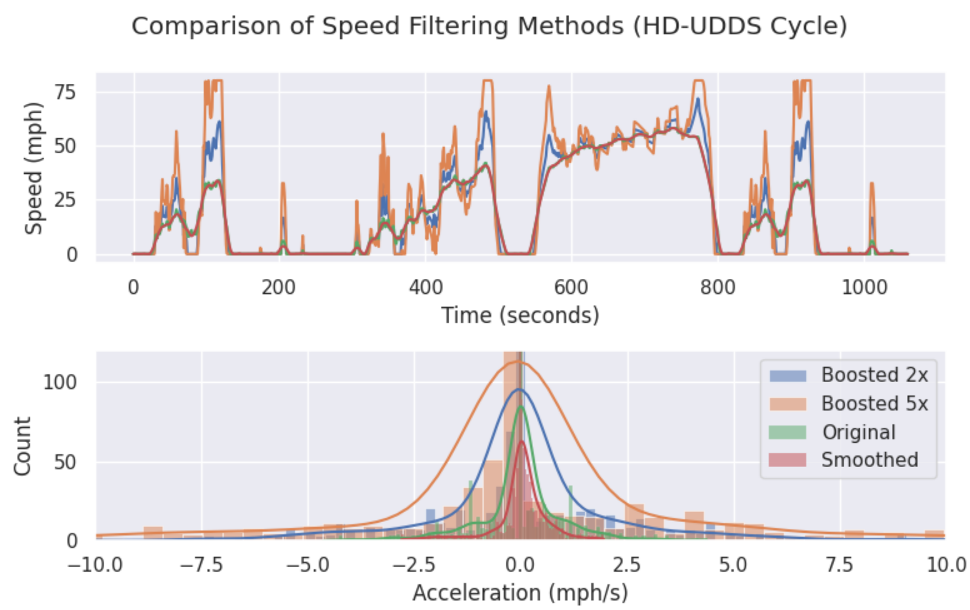

Figure 4.2: Boosting cycle deviations in speed creates higher peaks and greater-magnitude acceleration/deceleration events. 10 mph/s gives a 0-60 time of six seconds, approximately the same acceleration as a base Tesla Model 3.

Power consumption by the HVAC system due to door open time and outside temperature parameters was modeled based on simple assumptions regarding airflow in/out of the cabin and heat energy of that air at sea level [38] (Equation 4.2).

Charging needs are modeled using a simple, unmanaged depot-charging approach. At the end of each block, the bus is assumed to return to the depot and charge at full available plug power until the total consumed block energy is recharged. This approach does not account for possible opportunity charging at stops or along the blocks. It does not account for the possibility of charging at a rate less than the plug power to minimize peak loads, or during off-peak hours. It also does not account for possibly limited plug capacity at the depots, or swapping low- and high-SOC vehicles to other blocks between service days. It is intended only as a rough preliminary estimate of fleet charging needs, to inform more advanced and potentially cost-saving strategies such as on-route or managed charging.

4.5 Results and Discussion

4.5.1 Energy Modeling with Open and Standardized Bus Data

Validation of GTFS-RT Energy Estimates with Phone and GNSS Receiver Data

To understand the accuracy of energy predictions built on low resolution GTFS-RT, data were collected from a phone and GNSS receiver for three trips. These high resolution sources were used to directly construct three validation drive cycles. Figures 4.3 and 4.4 show the speed profiles of the phone and GNSS receiver location data. The speed profiles are calculated based on the derivative of position reported from each data source. The three high resolution drive cycles created for these trips were compared to drive cycles collected directly from the realtime data for the exact same vehicles.

Without filtering, the phone and GNSS trajectories have rare but large GPS point errors giving spikes of acceleration and deceleration. Applying a Savitsky-Golay filter to the phone and realtime data reduces the RMSE between the phone and the realtime speed from 10.5 mph to 9.1 mph. Despite this relatively small improvement in signal RMSE (-13%), the RMSE in energy consumption between the cycles improves significantly from 0.29 kWh/mi to 0.07 kWh/mi (-76%). Given that standard drive cycles give consumption in the realm of 2.77 kWh/mi (as shown in Table 4.2), this is a relatively small error.

The agreement in cycle consumption is driven by several factors. Figure 4.5 shows the large contribution of auxiliary power to the cycle's overall energy consumption. Auxiliary power is a fixed parameter of the design vehicle, making it the same for all cycle sources. Due to the dominant role of auxiliary power in the overall energy consumption, the impact

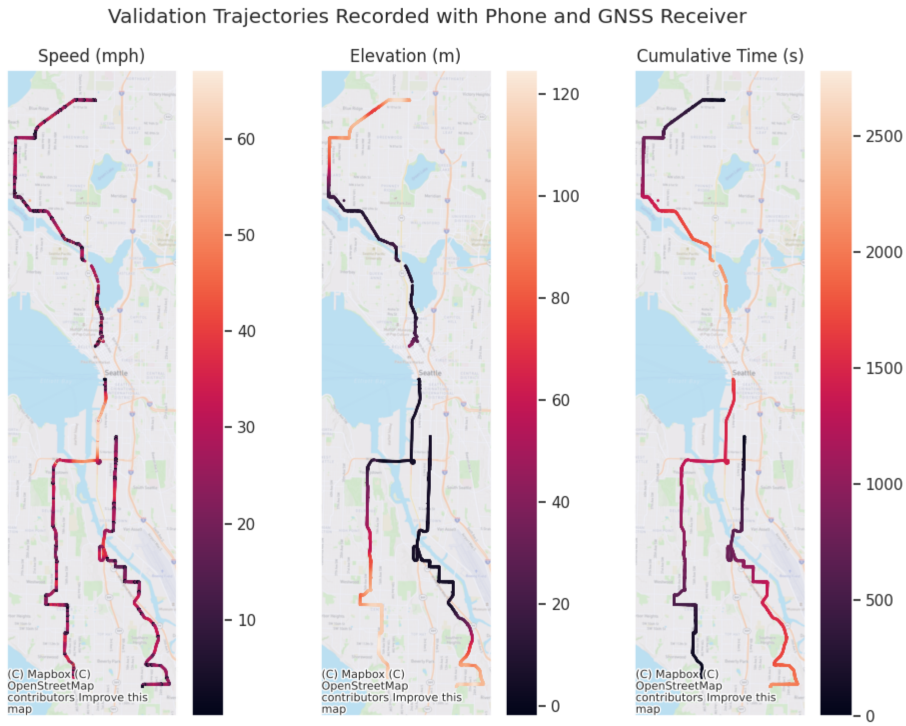

Figure 4.3: Several trips spanning the main KCM service area were captured using a phone and GNSS receiver. Drive cycles built on these high resolution sources provided validation for the realtime cycles.

of the cycle resolution differences is lessened. Note however that the aerodynamic drag and rolling resistance forces are significantly higher in the phone/GNSS cycles than the realtime cycle. This is particularly true in the last trip, which had the highest average and peak speeds of the three trips. For higher speed trips, where drag and rolling resistance begin to dominate the energy consumption, the differences in cycle resolution have a larger impact on the overall consumption. This is especially true for trips with intermittent high speeds, where the low resolution cycle completely misses high-speed events that greatly increase trip energy consumption.

In addition to missing sporadic high-speed events, there are many acceleration and decel-

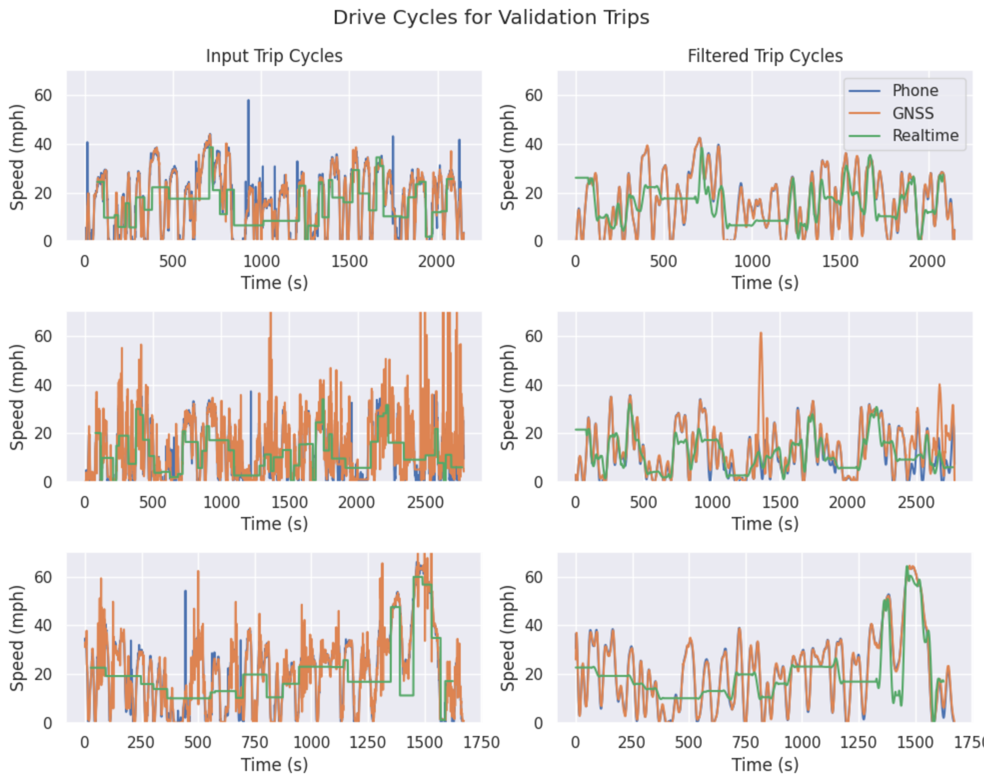

Figure 4.4: Clipping speeds to physical limits and applying a Savitzky-Golay filter to the phone and GNSS receiver data reduces the RMSE between the phone and GNSS by 2.3 mph, and between the phone and the realtime by 1.4 mph.

eration events missed in the realtime cycles due to their lower temporal resolution. However due to high maximum regenerative braking recovery for BEVs in FastSIM (98%) these are less impactful on overall consumption than they would be in a real-world scenario. More energy is typically spent accelerating than can be recovered decelerating with regenerative braking. This drives greater energy losses in the phone and GNSS cycles where there are more captured instances of acceleration and deceleration. For reference, reducing the maximum regenerative braking energy recovery to 0% increases the phone and realtime average consumptions by 1.07 kWh/mi (64%) and 0.85 kWh/mi (52%) respectively.

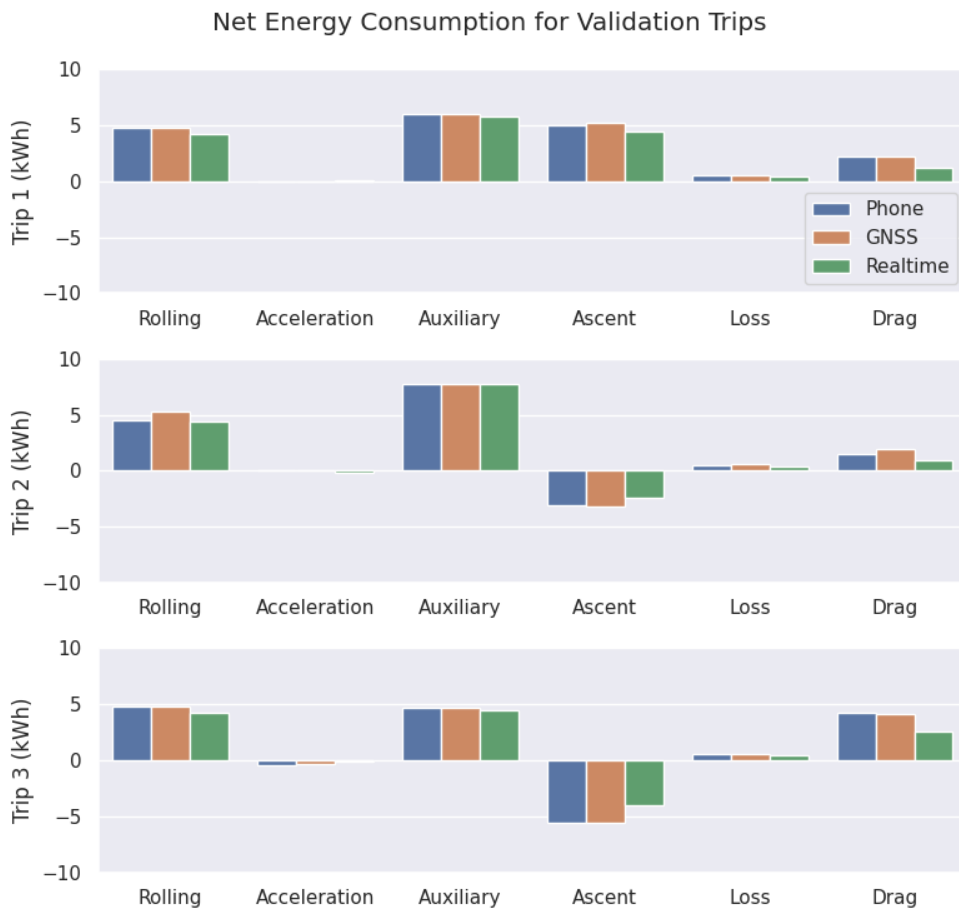

Figure 4.5: Net contribution of different BEB power consumption sources to total FASTSim-modeled BEB energy use. Trips 2-3 have a net elevation loss.

Last, the ascent energy data reveal a challenge with using realtime data for elevation analysis. Due to the lower resolution of the realtime trajectory, the interpolated path on the DEM strays off path when the vehicle traveled quickly around a corner. The result is a spike in elevation for the predicted drive cycle that was not experienced by the actual vehicle (Figure 4.6). In theory, the elevation data used as input to FASTSim should be identical for both cycles. This means that the power consumed in gaining elevation or regained from regenerative braking should be nearly the same for both sources. However, the elevation spike from the low resolution creates discrepancies in the power required to drive the cycle which contributes to error in net consumption.

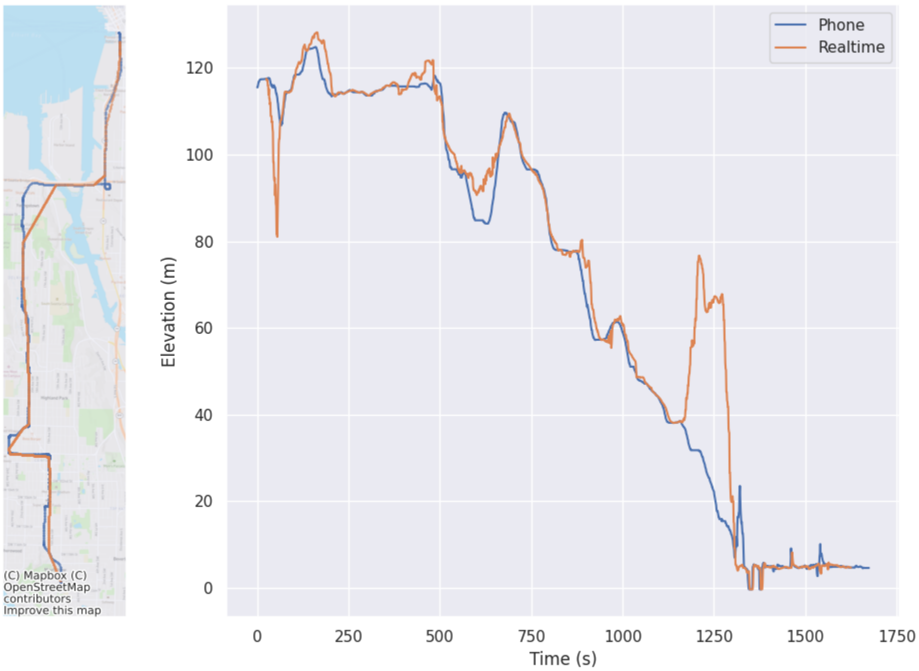

Figure 4.6: Validation Trip 3 trajectory and elevation profile for phone and realtime data sources. In the second half of the trip (as the vehicle approaches sea-level) the low resolution realtime data create an error in the elevation profile.

Comparison to Direct Aggregation of GTFS-RT

One validation method was tested for constructing drive cycles directly from GTFS-RT data. The validation method aggregated observed speed data from the GTFS-RT feed to a set of uniform-distance points according to prior work [178]. This was compared to the method using a generalizable travel time model to predict speeds between uniform-distance points. All drive cycles were smoothed using the Savitzky-Golay filter. Figure 4.7 shows the resulting drive cycles from applying these methods to three random trips from the KCM feed. The aggregation method uses data collected from the GTFS-RT feed for March 2024.

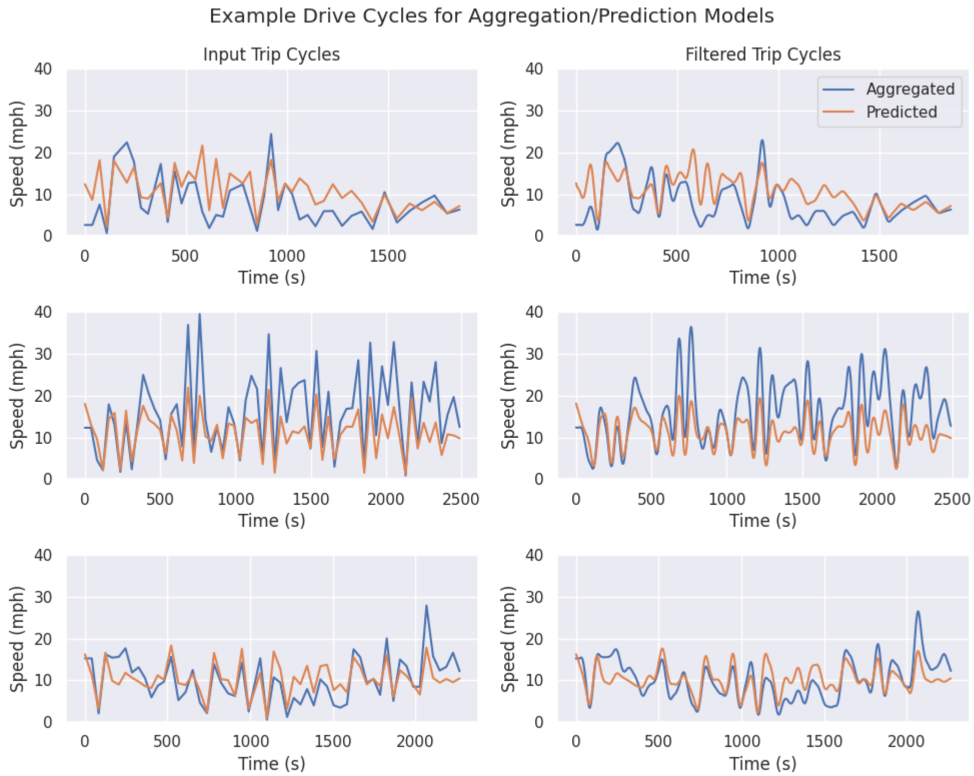

Figure 4.7: Sample of three trip drive cycles modeled using aggregation and prediction methods and smoothed using a Savitsky-Golay filter. The predicted method is less accurate on trips with sporadic high speeds. The RMSE for all trips in the feed is 0.23 kWh/mi.

The aggregation method approximates the most likely drive cycle of all buses traversing a route. The prediction method attempts to forecast a drive cycle based on all prior information from trips in the network. While the approximate speeds are similar, the prediction method tends to underestimate the amplitude of acceleration events compared to the aggregated profile (shown in cycles 3-4), especially when that acceleration occurs sporadically. This may be due to the training data missing acceleration and short term high speed events. Across all trips in the KCM network, the average RMSE of the aggregated and predicted cycles is 0.33 kWh/mi. This means that there is a small loss in precision by using predicted models rather than directly aggregating the GTFS-RT data, as would be expected from the results in Chapter 3. However, the benefit of using the travel time model is the ability to infer drive cycles for trips and networks without prior data.

Implications on Using the GTFS-RT Standard for Energy Modeling

The results demonstrate that drive cycles built on GTFS-RT tend to slightly underestimate energy consumption. This is driven by missed acceleration events that create higher aerodynamic and rolling resistance forces. The validation trajectories built on phone/GNSS receiver sources found a difference in estimated consumption of 0.29 kWh/mi compared to the realtime data. This was reduced to 0.07 kWh/mi by applying a Savitsky-Golay filter to both sources.

In this case, the relatively low error was accomplished due to the regenerative braking recovery, auxiliary power and (usually) identical elevation profiles of each source. At times, the low resolution of the realtime data led to differing elevation profiles. High contribution of auxiliary loads to trip energy consumption made drive cycle inaccuracies less impactful, especially for slow trips. If the same methodology were applied to non-BEB drive cycles, the lack of regenerative braking recovery would lead to greater power required for acceleration, and a larger discrepancy in consumption estimates. With a high resolution GTFS-RT feed, the discrepancy in consumption estimates would be reduced.

The GTFS-RT standard also uses low resolution ridership data (e.g., low, medium or high crowding) [81]. In the next section, differences in weight due to passenger load are shown to have an effect on energy consumption and block electrification viability. With more precise ridership data, the energy consumption of each block could be more accurately estimated. Higher resolution ridership data may also enable better estimates of door open/close times and auxiliary loads, which are also shown to have a significant impact on energy consumption. Last, the optional block ID parameter in the GTFS standard limits the trip-by-trip analysis for some networks.

4.5.2 Cross-Sectional Sensitivity Analysis

Modeling Baseline Blocking Constraints with KCM Network

Network-wide predicted bus drive cycles were used to model total energy needs for any given transit network. To validate these findings, data from existing BEBs in the KCM network for the month of December 2023 were used. The energy consumption of each block ($\hat{C_{ij}}$) was first estimated using the predicted drive cycles and FASTSim, then applied to Equation 4.1. The total energy and distance of each block including deadhead give a mean block consumption of 3.14 kWh/mi (Figure 4.8).

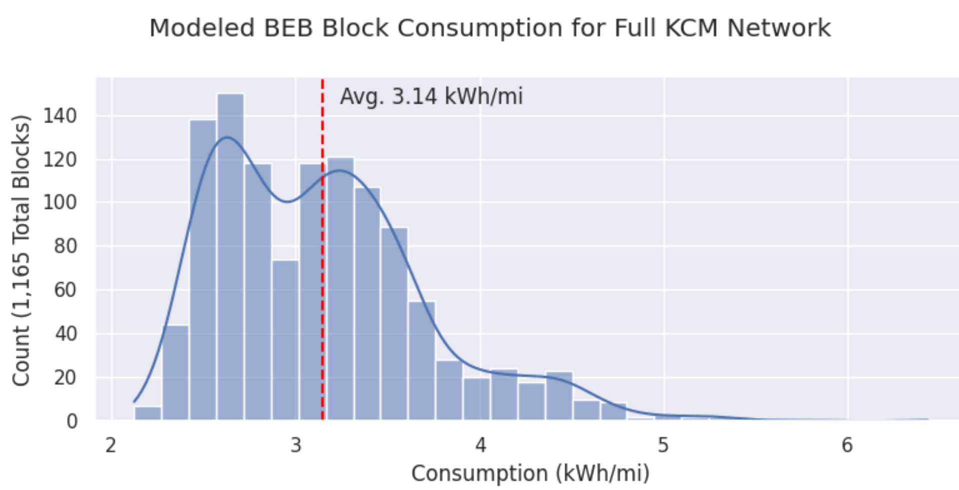

Figure 4.8: Predicted block consumption for the KCM network on a Wednesday in December 2023. Block consumption includes both in-service and out-of-service energy use as described in Equation 4.1.

Total block energy requirements impose constraints on the blocks which can be electrified without opportunity charging. Figure 4.9 shows the distribution of total block energy consumption for the KCM network. The battery capacity of the design vehicle (466 kWh) was used to determine the number of blocks that could be feasibly completed on a single charge. Under baseline assumptions, this was around 80% of blocks in the KCM network. The remaining 20% of blocks would require additional charging on-route, or larger BEB ranges. This is with new batteries; over the service life of the design vehicle (12 years) the number of blocks capable of being served would decrease. This might be addressed by rolling fleet replacements placing newer BEBs on the most energy-consuming blocks, as roughly 40% of blocks could still be covered by older vehicles with as low as half of the new battery capacity.

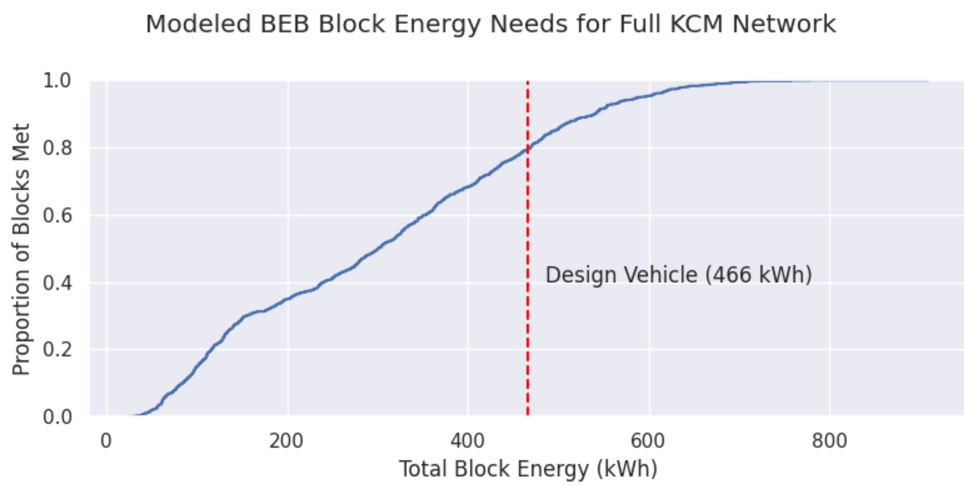

Figure 4.9: Predicted net block energy needs for the KCM network on a Wednesday in December 2023. The design vehicle battery capacity (466 kWh) was used to determine which blocks could feasibly be completed on a single charge.

The predicted block energy consumption was validated with reported block energy consumption for existing BEBs in the KCM network. Figure 4.10 shows the breakdown of energy consumption reported for the month of December 2023. The actual reported average daily energy consumption across the month for each block was matched to blocks from the modeled energy consumption. The error bars show the standard deviation of actual energy consumption for the various days reported for each block. The right plot shows the same data, but with trip-level consumption values $\hat{C}_{ij}$ and C_l scaled by 1.25. This factor is based on the ratio of average block energy for the energy model in this work (which is based on Altoona testing conditions) with a model prepared by Parametrix specifically for KCM [127].

This scalar is shown for illustrative purposes and was not used in any other results.

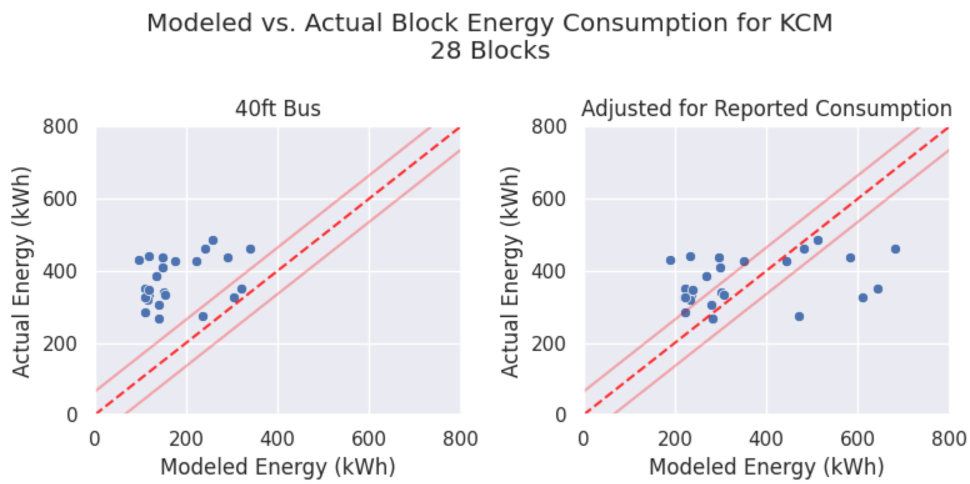

Figure 4.10: Modeled and actual BEB consumption on blocks in the KCM network (only electrified blocks having data are shown). The prediction method underestimates block energy needs. This is in part due to the design vehicle being based on Altoona testing conditions and simplifying assumptions regarding depot locations and routing. Data were collected from the KCM network during the month of December 2023.

From a modeling standpoint there may be other contributing factors to underestimating net energy. There is the assumption in clustering depots that they minimize distance to block starts, and the approximation of deadhead distance with Manhattan distance. In reality there may be longer routing to maneuver the various lakes, bridges and bottlenecks in the KCM system. The actual deadhead energy consumption rate may also be higher than estimated in the standard Manhattan cycle. Each of these would manifest as a lower modeled driving energy consumption compared to real-world values.

Under simple charging strategy assumptions the approximate power needs of the KCM network were also estimated. Figure 4.11 shows the number of active vehicles scheduled in the KCM GTFS, and the power demand of all charging vehicles per 15-minute period of the day. All vehicles were assumed to be electrified and charge immediately at baseline plug power until fully recharged when they return to their assigned depots.

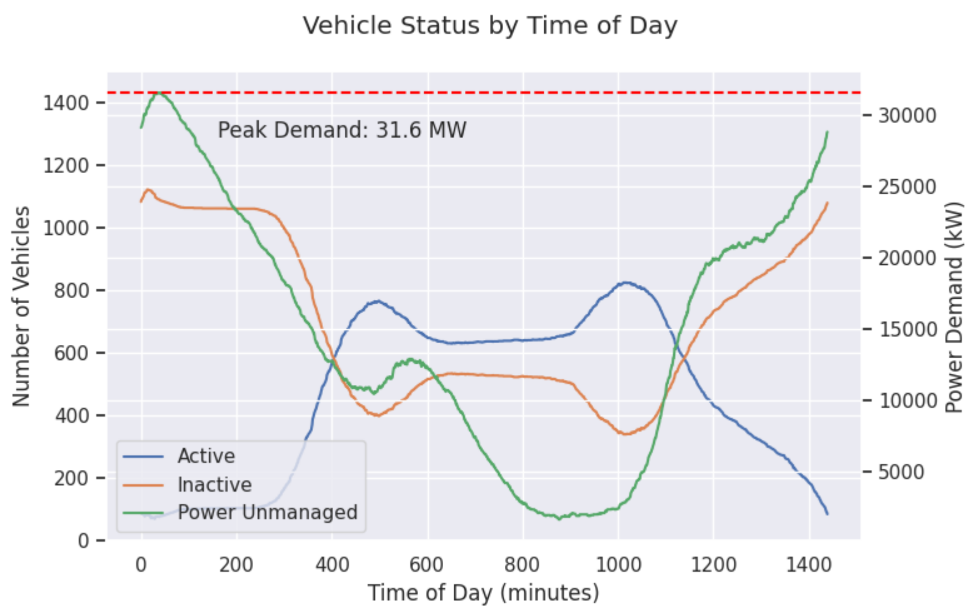

Figure 4.11: Active vehicles in the KCM network peak near 10AM and 5PM, with a large number of vehicles returning to depots after the PM peak. Higher baseline charging increases the peak demand and shifts it to match vehicle arrival times.

Assumptions of unmanaged charging lead to high overnight peak demand due to the large number of vehicles returning to depots after the PM peak. The peak power demand is about 31.6 MW, and occurs near midnight. As higher plug power is assumed, the peak would become higher and coincide more and more closely with PM peak vehicles arriving at the depot (a generally more expensive time of day). This is in direct tension with daily pullout needs of the fleet: With unmanaged charging and a 50 kW plug power, only about 75% of scheduled vehicles would be recharged by their pullout time. In order to electrify 95% of the fleet with unmanaged charging and meet daily pullout needs, a minimum plug power of 470 kW would be needed. This is on the upper range of available charger capacities, and many existing depots may not have the infrastructure to support this level of charging. This high upper tail is driven by long-duration blocks with relatively short scheduled down-times at the depot before departing.

Another way of framing charging needs is to divide the total network energy among the number of vehicle-hours available for charging at the depot. This more closely approximates the average power demand of the fleet under a scenario where managed charging is available. For KCM this is 21 kW per vehicle charging at the depot. In this case, the power curve would track the count of inactive vehicles in Figure 4.11. The peak power would occur when the most vehicles are at the base (i.e., still midnight) and would be about 23.7 MW. This significant reduction in peak power demand would require a more complex charging strategy, but would also reduce the overall cost of electrification by through lower peak power demand costs. Depending on electricity time of use rates, the greater mid-day consumption may reduce or increase costs.

The implication of these findings is that from an operational standpoint, many blocks can be quickly and easily electrified with no managed charging, no vehicle block swapping and no on-route charging. As a BEB fleet grows, the peak charging power required to meet all block pullouts would increase rapidly, at which point managed charging strategies could be deployed to reduce peak power demand. The tradeoff between the costs of administering a managed charging strategy and the costs of peak power demand is a key consideration for transit agencies looking to electrify their fleets.

A sensitivity analysis was performed to determine the impact of operational parameters on block energy consumption and charging needs. The results are shown in Figures 4.12 and 4.13.

Auxiliary power has a strong effect on the energy consumption of BEBs in the fleet. Depending on the level of utilization it can lead to a difference of up to 3.0 kWh/mi. This level of uncertainty is highly undesirable for fleet electrification, as it can increase the energy required to run a block by up to 50% (Figure 4.13). If underestimated, this could lead to stranded BEBs or missed pullouts due to insufficient charging capacity. As shown in Figure 4.12 much of the impact from auxiliary load is captured in the energy consumption estimates

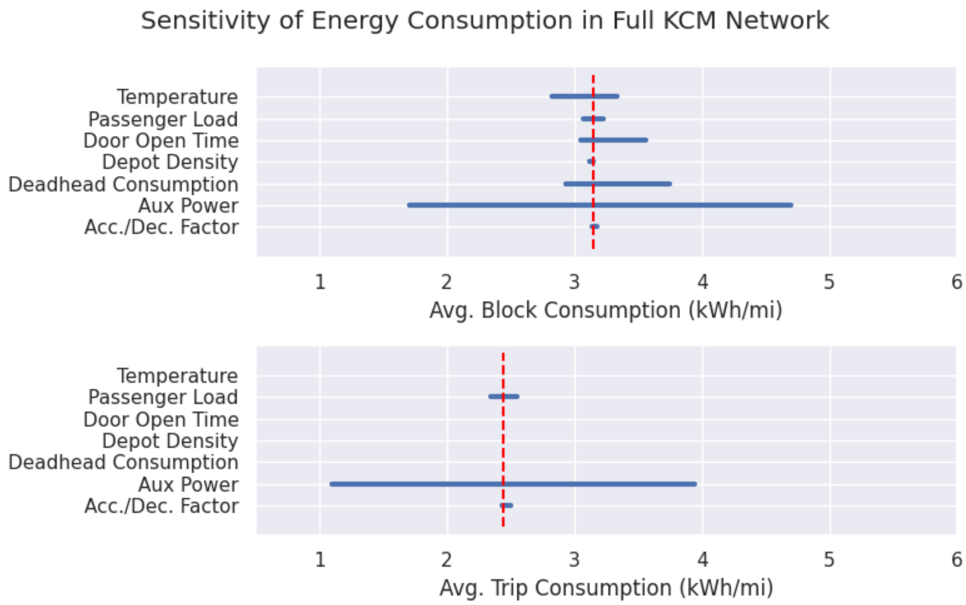

Figure 4.12: Sensitivity of block (Equation 4.1) and in-service ($\hat{C}_{ij}$) consumption to various operational parameters. Auxiliary loads manifest in the trip consumption and strongly affect overall block consumption.

for in-service trip driving ($\hat{C}_{ij}$). This indicates that reducing deadhead auxiliary use (P) may not have a large impact on net consumption (i.e., the layover times l_{ij} are not large for most trips).

Other parameters affecting the in-service trip consumption have a minimal effect. The implication of this is that one of the best ways to operationally reduce energy consumption for BEBs is to coach drivers to reduce HVAC usage, rather than driving habits. Figure 4.12 also shows that routes with high passenger loads are just as viable for electrification as others, as the increased weight generally doesn't impact consumption. This does not account for relationships with the number of passengers, outside temperature, HVAC usage and more that could be modeled with a complex heat transfer model.

Despite high uncertainty, Figure 4.13 shows that even under worst-case scenario for aux-

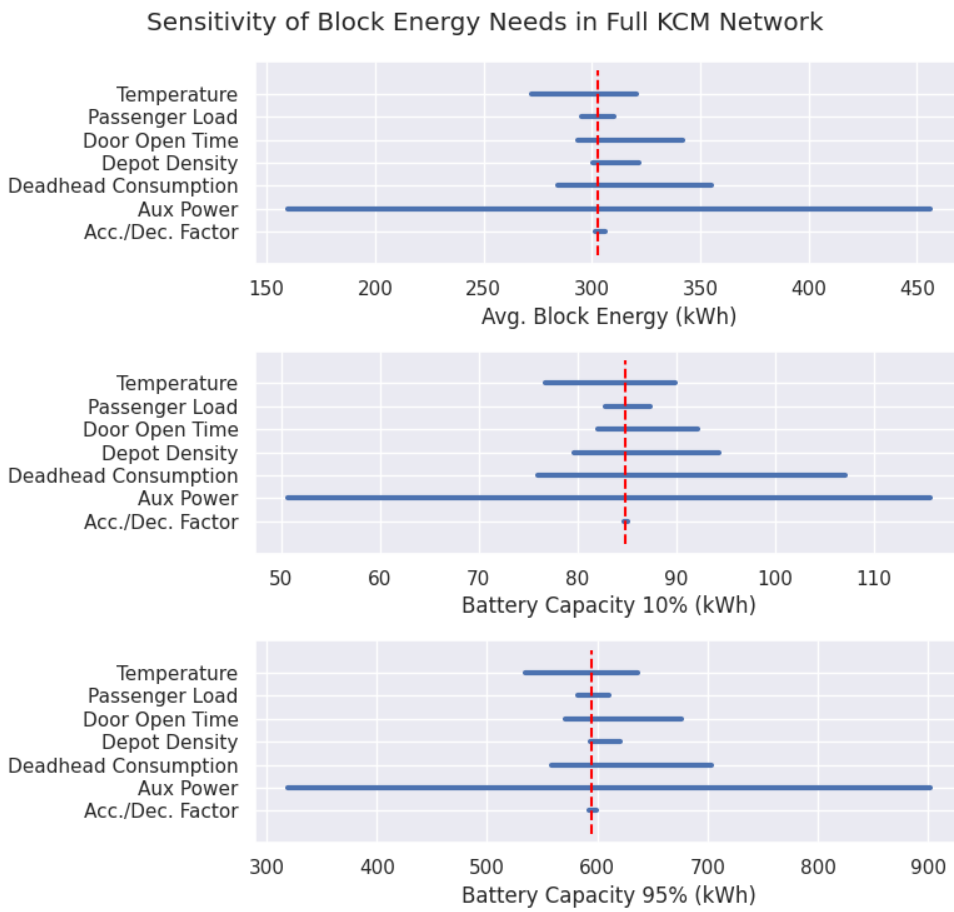

Figure 4.13: Even under worst case loads, 10% of blocks could be electrified with existing BEB technology. Current capacities are not sufficient for full fleet electrification.

iliary loads 10% of blocks could be electrified with a usable battery capacity of only 120 kWh. This is well within the capabilities of the BEB market today. However, to meet a full (95%) electrification scenario, many of the tested parameters would push blocks outside of that capable range. To meet fleet electrification goals KCM would need BEBs with higher ranges or diesel heaters and other operational treatments to offset auxiliary loads.

Cross-Sectional Results for International Networks

For a more complete picture of agency needs regarding fleet electrification, the energy modeling and sensitivity frameworks were extended to GTFS and GTFS-RT feeds from 40 international bus networks (Appendix .2). For each network, key technological requirements for fleet electrification were determined. These included the minimum battery capacity needed to meet 10/95% of block energy needs, and the minimum plug powers needed to meet similar block pullouts. These results examine how the constraints to fleet electrification vary across different networks, and how aspects of the network can affect these constraints.

First, nearly every agency in the study could electrify 10% of their blocks with unmanaged depot charging. Figure 4.14 shows the battery capacity required to electrify 10/95% of each agency's fleet. The ranges shown represent the range of auxiliary loads. Note that the GTFS standard lists "block_id" as an optional parameter. For agencies not reporting block IDs, each trip was considered a block. This will underestimate the net required energy for each vehicle in that network, and overestimate the amount of time available at the depot for charging.

To achieve full BEB fleet electrification with the design vehicle for this study (466 kWh battery capacity), every agency listing block IDs would need to operate at the lowest end of auxiliary power. This assumes baselines for other parameters such as temperature, driver acceleration behavior and passenger load which could further reduce the margin of error. Based on these findings, the current BEB market is inadequate for full fleet electrification. Many agencies are challenged with 2030 zero-emission goals which based on these results may be impossible without re-routes or other large operational changes. The difference in block energy use between agencies is largely driven by block distances and times. No relationship was found between block distance/time and the rate of energy consumption for that block.

Unmanaged charging is the simplest and most affordable strategy from an implementation standpoint. It does not require equipment or personnel outside of chargers. For an initial fleet rollout, very low charger power can be used. Figure 4.15 shows the range of charger

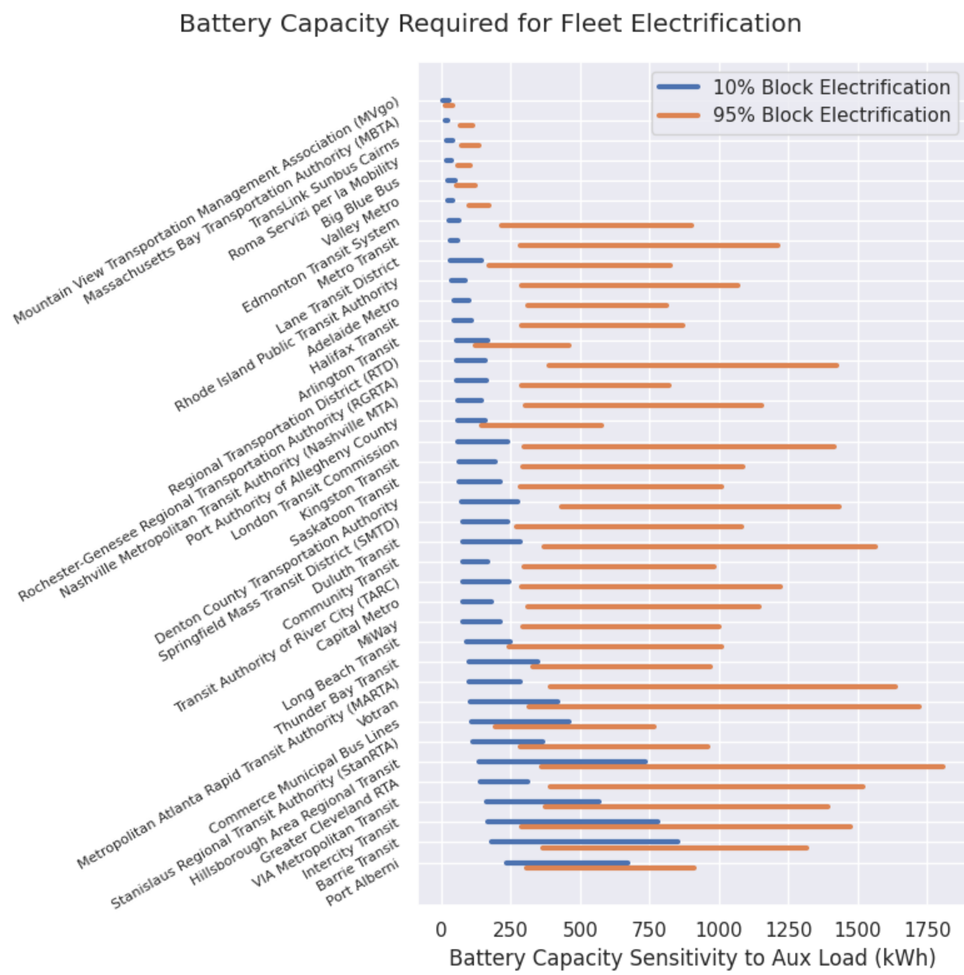

Figure 4.14: Nearly all agencies could electrify 10% of blocks with unmanaged depot charging, even under worst case auxiliary loads. Note that agencies not reporting block IDs are assigned one block per trip.

powers required to meet 10%. These blocks do not necessarily align with those electrified in Figure 4.14, but higher-energy blocks are mostly the result of longer distances and durations (as opposed to inefficiencies from traffic, etc.). These also lead to higher plug power due to higher energy needs and less depot charging time available. For full fleet electrification, the power required to meet block pullout becomes excessively high along with peak demand.

There is no choice but to transition to managed charging.

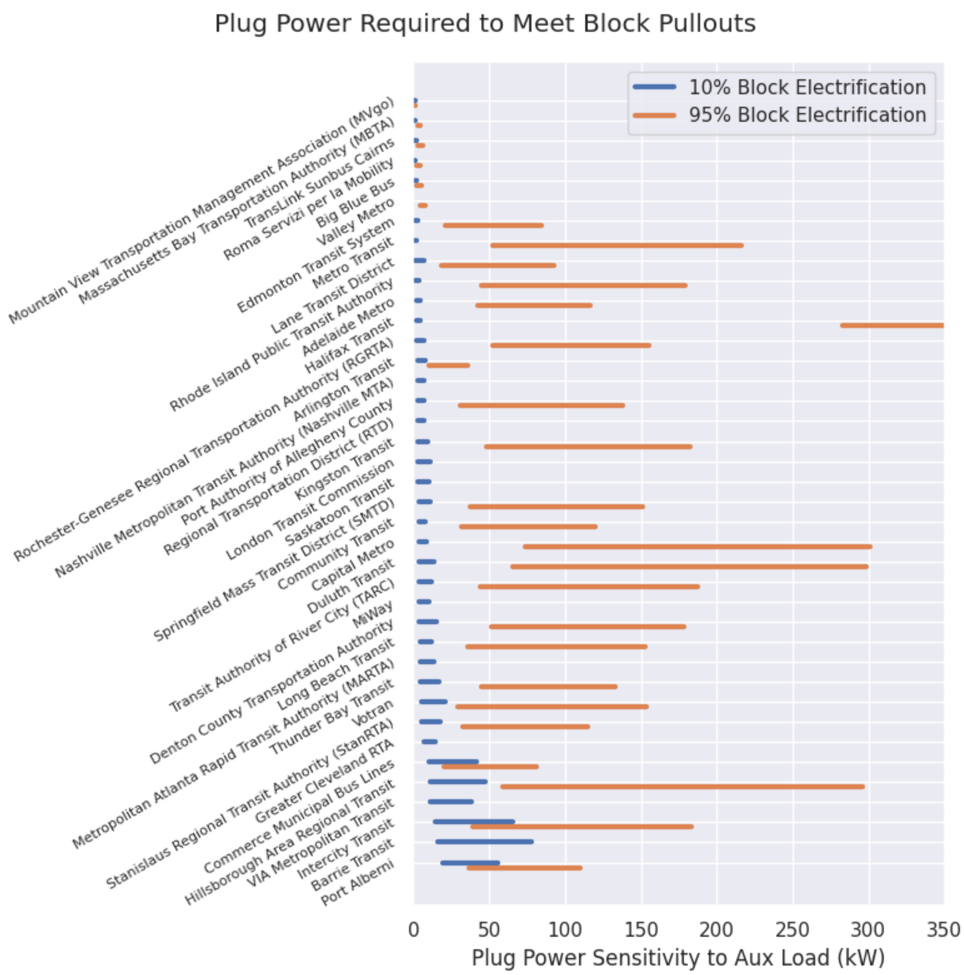

Figure 4.15: Nearly all agencies could meet pullouts for 10% of blocks with unmanaged depot charging, even under worst case auxiliary loads. Note that agencies not reporting block IDs are assigned one block per trip. Some agencies report block turnaround times of only a few minutes. This leads to extraordinarily high charging requirements. The range of reasonable plug powers covering most agencies is shown.

The ideal charging management strategy charges vehicles as slowly as possible, while still meeting block energy pullout needs. To approximate the average power demand from a managed charging strategy, the total energy of the network is divided by the number of vehicle-hours available for charging at the depot. This assumes a lower-bound best case scenario for charging power where all vehicle pullouts can be met by swapping vehicles between blocks.

One strategy to reduce individual block energy needs and peak demands is to run more vehicles with shorter blocks. Figure 4.16 shows the relationship between the average block distance and average managed charging rate (per-inactive vehicle) in the study networks (a similar relationship holds for average block duration and charging rate). The average block distance is a function of deadheading and scheduled trip distances. The average charge rate is a function of the energy consumed by the full network and the duration of the blocks. When block distance is high, vehicles consume more energy. When block duration is high, vehicles have less time to charge at the depot. These compounding factors create high plug powers at the depot.

Therefore, as shown in Figure 4.16, average and peak power can be greatly reduced by agencies running shorter blocks. This may be accomplished by splitting blocks and running more BEBs (or swapping BEBs between blocks). This decision would have to be weighed against costs for implementing block swapping or purchasing additional vehicles. Given that BEBs typically have high capital and low operating costs, the extra vehicles may not justify the power savings.

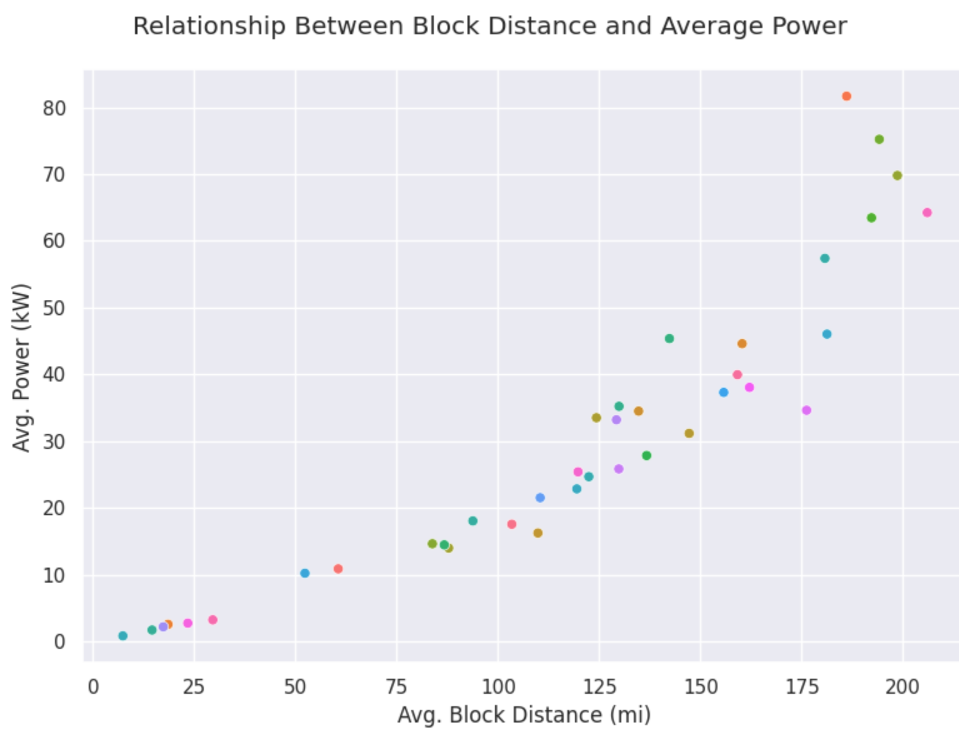

Figure 4.16: Decreasing block distance through block splitting or re-routing can greatly reduce average and peak power demands. Vehicles spend more time at the depot and can charge more slowly under a managed scenario. This comes at a tradeoff to operating more vehicles and having greater deadhead miles. Note that agencies not reporting block IDs are assigned one block per trip.

Implications on Viability BEB Fleet Electrification

The results demonstrate that for all transit agencies in the study, even under highly conservative estimates of auxiliary power loads most networks can support 10% block electrification with little to no managed charging. This makes BEB pilot projects relatively straightforward, but not necessarily cost effective. To evaluate the costs for a full fleet transition the capital and operational costs of the pilot BEBs would have to be weighed against potential federal funding, future BEB technology, and the specific needs of the network.

To support full electrification, BEB capacities of at least 500-1000 kWh are needed. This can be reduced through operational approaches such as driver acceleration and limiting door open time. However, by far the greatest source of uncertainty remains the auxiliary load. As a greater proportion of the fleet is electrified, peak demand and charger power to meet block pullouts will necessitate managed charging. Under best-case scenarios for managed charging, plug power as low as 30 kW may be possible.

To meet net-zero goals, vehicle battery capacities will need to improve significantly. Current charger powers may be sufficient to meet full-electrification goals with adequate plugs and managed charging strategies. Purchasing additional BEBs to run the same quantity of service miles would greatly reduce average block distance and thus average and peak charging power. Running these shorter blocks with lower capacity buses could also decrease consumption and total energy needs by using lighter vehicles with smaller batteries. Ultimately, the cost of these strategies must be weighed carefully against their benefits if transitioning to a fully electric fleet.

4.6 Conclusions

This work proposed a new method for developing BEB drive cycles specific to any transit network and demonstrated that method on a fleet electrification study of 40 international bus networks. The core data came from GTFS and GTFS-RT which are standardized formats that can be readily collected for most transit networks. The study evaluated the shortcomings of the GTFS-RT data format for constructing drive cycles and the implications of using the standard for energy analysis. It also evaluated the sensitivity of network energy consumption and charging needs related to key operational parameters. Drive cycle findings were validated against individually-collected phone and GNSS receiver recordings onboard KCM buses. Network energy findings were validated against reported consumption for existing KCM BEBs.

The results show that the GTFS-RT standard is a useful tool for constructing drive cycles, but that the low resolution of the data can lead to underestimation of energy consumption by 0.03 to 0.25 kWh/mi (depending on the amount of post-processing for the validation and GTFS-RT data). This is especially true for cycles with sporadic acceleration events characteristic of urban bus routes. Routes where low speeds and large elevation changes control BEB power consumption lead to estimates on the more accurate side of this range. Also, GTFS-RT uses low resolution ridership data which can be used to model weight from passenger loads. Passenger load effects on consumption were evaluated through sensitivity rather than reported GTFS-RT values. However they were found to potentially impact average block consumption by up to 0.20 kWh/mi.

Results from the cross-sectional agency analysis revealed that with current technology, most agencies can electrify 10% of their blocks. This assumes completely unmanaged depot charging; the simplest operational strategy for BEB charging. As the proportion of blocks to be electrified grows, usable energy capacity becomes the main limiting factor. Capacities of 500-1000 kWh for the given design vehicle would be needed to electrify all blocks; these are on the upper limit of the market today. To meet block pullouts under increasing electrification, managed charging is essential. Under perfect efficiency, most agencies incur an average power demand of 30 kW per-vehicle at the depot. The compounded relationship between average power demand and average block distance means that more vehicles serving the same number of service miles may be a viable way to reduce peak and average power demand. Future work might examine the costs of this tradeoff in greater detail.

Limitations of the study were assumptions regarding depot locations, deadhead distances and charging strategies. In future work, more precise locations and routing decisions for deadhead trips could be modeled to improve energy consumption estimates. Managed charging strategies, including on-route opportunity charging could greatly reduce the BEB capacity needed to meet block pullout needs. They may also reduce peak power demands and required depot charging power. Ultimately, block energy consumption as calculated in this study could be used as a constraint in blocking and scheduling optimization software. However it is unlikely to be sufficient on its own to fully inform a fleet electrification strategy. Future work could include a more detailed analysis of the cost implications of different charging strategies, and the impact of different operational parameters on the total cost of ownership of a BEB fleet.

Chapter 5

CONCLUSIONS

5.1 Summary

Tools for planning and analyzing the operations of bus transit systems are primarily licensed and closed source. This can limit their applicability to research and precipitates home-grown solutions unique to individual systems. Open and standardized bus data are readily available to support these analyses, but they come with some caveats to precision. Much prior work has demonstrated the value of these data in providing customer-facing utilities such as realtime arrival information. However few have applied them to agency planning or operations analyses. This work strived to address this gap by exploring open source data solutions and their limitations for bus transit operational analyses.

First, the segment delay analysis in Chapter 2 provided a system-wide tool for visualizing bus delays aggregated to segments. For system-wide operations analysis, there are many closed source tools available. For example Tripspark, PTV, Remix and Transloc offer comprehensive dashboards which track realtime performance metrics [168, 85, 170, 165]. These tools also include varying levels of integration with sensor collection hardware, routing algorithms and scheduling. Some open source tools such as Pantograph and geOps visualize realtime measured or scheduled vehicle movements but do not offer analysis of the operations or historic information [17, 59]. This study used GTFS-RT to identify locations in the network where targeted transit priority treatments could generate the greatest benefit. It found that one of the key challenges with open realtime bus data which is their relatively low temporal resolution. Aggregating observations from many trips and routes revealed network characteristics, but delays to individual trips were lost in the realtime feeds. In practice, this makes realtime feeds a useful reporting and comparison tool for aggregate performance metrics (e.g., freeflow travel times, route-level average delays). However they lack the precision required for analysis of specific vehicles (e.g., stop times, causes of the delays for individual bus trips).

The generalizable travel time modeling in Chapter 3 tested methods for forecasting bus travel times across networks, on new routes and with varying levels of data availability. It used a geospatial data mining approach to capture aggregate information from a large dataset of bus trajectories and attempt to draw predictions for vehicles on specific routes and trips. Bus travel time prediction is well explored in both open and closed source tools. For example closed source tools such as Tripspark and Transloc use short-horizon travel time prediction to provide realtime arrival information [168, 165]. However when reviewing the literature in Chapter 3, very few open source models were found capable of forecasting across networks. Most were tested on a handful of routes from a single network, or even only a few trips. This study found that with only a handful of fine-tuning samples from a target network, a bus travel time model trained on copious data from a source network can generalize with performance nearly as good as a model trained on the target. It also demonstrated how the model can be used as a planning tool for routes in the source network with no prior data, by mining information from other nearby bus trips. While impractical for agencies, this approach to travel time forecasting offers benefits to researchers and planners who often seek to quickly and easily draw comparisons across many transit networks.

Last, there are analyses outside of existing closed and open source toolsets that require agencies to contract solutions or perform in-house research. These benefit immensely from open source data and code such that agencies can collaborate and benefit from other's findings. The energy drive cycle analysis constructed on open data standards in Chapter 4 builds on prior literature in analyzing the energy demand of electric buses. Prior work in this topic focused on aggregating historic realtime data to static route segments [178, 51, 68]. This work constructed predictive models based on the historic realtime data to generate drive cycles for new segments and networks. Based on findings from Chapters 2 and 3 related to challenges with non-aggregated GTFS-RT data, it examined the accuracy of low resolution realtime data in modeling energy demands with a manually collected high resolution baseline. It then tested a model for network-wide block energy consumption with real world data from the KCM network, and extended the model to a cross sectional agency analysis. It found that the limitations of the open data standard had a relatively minor impact on network energy calculations. A cross-sectional analysis of energy consumption across agencies identified some of the greatest barriers to BEB adoption.

All together, these findings point to 1) The viability of open and standardized transit data sources in supporting operational analyses for both agencies and researchers and 2) The potential benefits of increasing the level of detail and required variables reported in these standards.

5.2 Limitations

The three studies in this work were targeted to specific tasks whereas most transit operations and planning software offer a suite of tools. Evaluating the ability of open bus data to fully replace more detailed ITS data collected by agencies would require applying and comparing these tools side-by-side. Ultimately, this limits the applicability of these findings. The true benefit of these open data is in examining metrics across many networks, which falls mostly in the realm of research and planning. Individual transit agencies may look to others as benchmarks, but still benefit greatly from the detailed monitoring and analysis systems built on their high resolution ITS data. In future work, more detailed analyses such as anomaly detection, stop performance, re-route design, scheduling and more could be built on the limitations of open standards, then compared to state of the art tools.

This work did not cover the SIRI and NeTEx standards in detail. Conceptually they are similar to GTFS-RT and GTFS, but with slightly different goals, parameters and data format. Extending this work to SIRI and NeTEx would offer a greater pool of cities with static and realtime data for the analyses in Chapters 3 and 4. Building generalizable tools for both standards may also help bridge the gap between them. Furthermore, some of the criticism leveled at GTFS-RT and GTFS feeds centers around lack of conformance to the standard. A more comprehensive comparison of adherence across both sets of standards may reveal potential improvements for either side.

There are also some limitations specific to each study. The analyses of transit performance metrics in Chapter 2 were relatively simplistic. They did not include stop performance or attempt to model the effects of targeted infrastructure improvements on performance. The travel time models used for Chapter 3 did not include map-matching for the training data, which may have reduced their accuracy in some cases. Pairing trips to the road network may have also supported more detailed tagging features from OSM, rather than relying on embeddings. Many assumptions were made to support the general energy analysis in Chapter 4, but most critical was the lack of managed charging, plug configurations or other charging strategies to lower peak energy demands. In a realistic scenario, the cost-savings from optimizing opportunity charging, managed depot charging and plug configurations likely outweighs the cost of their implementations. More realtime data could have been collected, and the precision of the GPS measurements could have been validated with respect to the urban environment (e.g., does the loss of precision in urban canyons lead to systematic errors in the drive cycle energy?). A variety of design vehicles could also have been tested to establish whether the current BEB market meets the blocking needs of agencies. Heat losses and auxiliary loads also could have been modeled in greater detail, given their large impact on trip energy consumption.

5.3 Future Work

Several interesting avenues for future research arose in the completion of this work. First, calibrating the energy models (which used predicted travel times) involved careful tuning of BEB design vehicle parameters and post-processing of the drive cycles. Ultimately, the travel time model was essentially used as a proxy for most-likely energy consumption on a trajectory. In future work, one might train a forecast model directly on the estimated energy consumption. This would require collection of many drive cycles to use as training data. RoutE (Route Energy Prediction Model) is an NREL tool which attempts to solve this problem. Second, this work focused only on required components of the GTFS-RT standard. Optional components such as measured speed and passenger loads are available in some feeds. With these features, more complex analyses could be built which better estimate energy consumption. Last, one of the key benefits of the models tested in Chapter 3 was the ability to generalize predictions to new routes within the source network without any training data. This could be used to develop energy-efficient re-routes, or to more precisely model deadhead energy consumption between depots and trips for the energy models in Chapter 4.

BIBLIOGRAPHY

[1] Alana Aamodt, Karlynn Cory, and Kamyria Coney. Electrifying transit: A guidebook for implementing battery electric buses. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2021.

[2] Hatem Abdelaty and Moataz Mohamed. A framework for beb energy prediction using low-resolution open-source data-driven model. *Transportation Research Part D: Transport and Environment*, 103:103170, 2022.

[3] Federal Transit Administration. 2021 capital expenses. Technical report, Federal Transit Administration, 2021.

[4] Federal Transit Administration. 2021 national transit summaries and trends. Technical report, Federal Transit Administration, 2021.

[5] Zack Aemmer, Andisheh Ranjbari, and Don MacKenzie. Measurement and classification of transit delays using gtfs-rt data. *Public Transport*, 14:263–285, 2022.

[6] Zack Aemmer, Sondre Sørbø, Alfredo Clemente, and Massimiliano Ruocco. Generalization strategies for improving bus travel time prediction across networks. *Journal of Urban Management*, 2024.

[7] Environmental Protection Agency. Dynamometer drive schedules. https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules. Accessed: 2023-09-13.

[8] Environmental Protection Agency. Fast facts: Us transportation sector greenhouse gas emissions. Technical report, Environmental Protection Agency, 2021.

[9] Ali Saadon Al-Ogaili, Agileswari Ramasamy, Tengku Juhana Tengku Hashim, Ahmed N Al-Masri, Yap Hoon, Mustafa Neamah Jebur, Renuga Verayiah, and Marayati Marsadek. Estimation of the energy consumption of battery driven electric buses by integrating digital elevation and longitudinal dynamic models: Malaysia as a case study. *Applied Energy*, 280:115873, 2020.

[10] Jaime Allen, Juan Muñoz, and Juan de Dios Ortúzar. Understanding public transport satisfaction: Using maslow's hierarchy of (transit) needs. *Transport Policy*, 81, 06 2019.

[11] Aaron Antrim and Sean J. Barbeau. Opening the Door to Multimodal Applications: Creation, Maintenance and Application of GTFS Data. In *Opening the Door to Multimodel Applications*, volume TRB 96th Annual Meeting Compendium of Papers, Washington DC, United States, 2017.

[12] Aaron Antrim, Sean J Barbeau, et al. The many uses of gtfs data–opening the door to transit and multimodal applications. *Location-Aware Information Systems Laboratory at the University of South Florida*, 4, 2013.

[13] B. P. Ashwini, R. Sumathi, and H. S. Sudhira. Bus Travel Time Prediction: A Comparative Study of Linear and Non-Linear Machine Learning Models. *Journal of Physics: Conference Series*, 2161(1):012053, 2022. Publisher: IOP Publishing.

[14] Sean J Barbeau. Quality control-lessons learned from the deployment and evaluation of gtfs-realtime feeds. In *97th Annual Meeting of the Transportation Research Board, Washington, DC*, 2018.

[15] Sean J. Barbeau, Minh Pham, Jorge Adorno Nieves, and Robert L. Bertini. Improving Transportation Performance Measurement via Open "Big Data" Systems – Phase 1 Transit. Technical report, Center for Transportation, Equity, Decisions & Dollars (CTEDD), Arlington, Texas, 2020.

[16] Richard Barnes, Senaka Buthpitiya, James Cook, Alex Fabrikant, Andrew Tomkins, and Fangzhou Xu. BusTr: Predicting Bus Travel Times from Real-Time Traffic. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 3243–3251, New York, NY, USA, 2020. Association for Computing Machinery.

[17] Hannah Bast, Patrick Brosi, and Sabine Storandt. Real-time movement visualization of public transit data. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '14, page 331–340, New York, NY, USA, 2014. Association for Computing Machinery.

[18] John Bates, John Polak, Peter Jones, and Andrew Cook. The Valuation of Reliability for Personal Travel. *Transportation Research Part E: Logistics and Transportation Review*, 37(2):191–229, 2001.

[19] Justin Beaudoin and CYC Lin Lawell. Is public transit's "green" reputation deserved?: Evaluating the effects of transit supply on air quality. *University of California at Davis Working Paper*, 2016.

[20] Bentley. Cube access accessibility planning and analysis software. https://www.bentley.com/software/cube-access/. Accessed: 2023-09-13.

[21] Annabell Berger, Andreas Gebhardt, Matthias Müller-Hannemann, and Martin Ostrowski. Stochastic Delay Prediction in Large Train Networks. In *Stochastic Delay Prediction in Large Train Networks*, pages 100–111, Saarbrücken, Germany, 2011. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[22] Mathew Berkow, John Chee, Robert Bertini, and Christopher Monsere. Transit Performance Measurement and Arterial Travel Time Estimation Using Archived AVL Data. In *ITE District 6 Annual Meeting*, page 6, Washington DC, United States, 2007. Institute of Transportation Engineers.

[23] Robert L. Bertini and Ahmed El-Geneidy. Generating Transit Performance Measures with Archived Data. *Transportation Research Record*, 1841(1):109–119, 2003. Publisher: SAGE Publications Inc.

[24] Robert L. Bertini and Sutti Tantiyanugulchai. Transit Buses as Traffic Probes: Use of Geolocation Data for Empirical Evaluation. *Transportation Research Record*, 1870(1):35–45, 2004.

[25] Larry A. Bowman and Mark A. Turnquist. Service Frequency, Schedule Reliability and Passenger Wait Times at Transit Stops. *Transportation Research Part A: General*, 15(6):465–471, 1981.

[26] Mário Braga, Maribel Yasmina Santos, and Adriano Moreira. Integrating public transportation data: creation and editing of gtfs data. In *New Perspectives in Information Systems and Technologies, Volume 2*, pages 53–62. Springer, 2014.

[27] Aaron Brooker, Jeffrey Gonder, Lijuan Wang, Eric Wood, Sean Lopp, and Laurie Ramroth. Fastsim: A model to estimate vehicle efficiency, cost and performance. Technical report, SAE Technical Paper, 2015.

[28] Caliper. Transcad transportation planning software. https://www.caliper.com/tcovu.htm. Accessed: 2023-09-13.

[29] Nick Caros, Xiaotong Guo, Anson Stewart, and John Attanucci. Systematic Bus Corridor Identification and Performance Measure Aggregation: Method and Case Studies. https://annualmeeting.mytrb.org/OnlineProgramArchive/Details/15905, 2021.

[30] Avishai Ceder. Urban transit scheduling: framework, review and examples. *Journal of urban planning and development*, 128(4):225–244, 2002.

[31] Alternative Fuels Data Center. Low and zero emission public transportation funding. https://afdc.energy.gov/laws/12849. Accessed: 2023-09-13.

[32] Alternative Fuels Data Center. Public transportation research, demonstration, and deployment funding. https://afdc.energy.gov/laws/11552. Accessed: 2023-09-13.

[33] Fabian Cevallos, Xiaobo Wang, Zhenmin Chen, and Albert Gan. Using avl data to improve transit on-time performance. *Journal of Public Transportation*, 14(3):21–40, 2011.

[34] Laura Cecilia Cham. *Understanding bus service reliability : a practical framework using AVL/APC data*. Thesis, Massachusetts Institute of Technology, 2006.

[35] Kevin Chandler, Erin Eberts, and Leslie Eudy. Interim evaluation results from new flyer/allison hybrid buses at king county metro. Technical report, National Renewable Energy Lab., Golden, CO (US), 2005.

[36] Zebin Chen, Xiaolin Xiao, Yue-Jiao Gong, Jun Fang, Nan Ma, Hua Chai, and Zhiguang Cao. Interpreting trajectories from multiple views: a hierarchical self-attention network for estimating the time of arrival. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2771–2779, 2022.

[37] Daniel Chindamo and Marco Gadola. What is the most representative standard driving cycle to estimate diesel emissions of a light commercial vehicle? *IFAC-PapersOnLine*, 51(5):73–78, 2018.

[38] Gabriel Chiriac, Dumitru Dorin Lucache, Costică Nițucă, Alin Dragomir, and Seeram Ramakrishna. Electric bus indoor heat balance in cold weather. *Applied sciences*, 11(24):11761, 2021.

[39] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[40] Eva Chondrodima, Harris Georgiou, Nikos Pelekis, and Yannis Theodoridis. Public transport arrival time prediction based on gtfs data. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 481–495. Springer, 2021.

[41] Christy Coghlan, Sina Dabiri, Brian Mayer, Mitch Wagner, Eric Williamson, Michael Eichler, and Naren Ramakrishnan. Assigning Bus Delay and Predicting Travel Times using Automated Vehicle Location Data:. *Transportation Research Record*, 2673(3):624–636, 2019.

[42] Antonio Comi and Antonio Polimeni. Bus Travel Time: Experimental Evidence and Forecasting. *Forecasting*, 2(3):309–322, 2020. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[43] Conveyal. Conveyal analysis. https://conveyal.com/learn. Accessed: 2023-09-13.

[44] Filip Covic and Stefan Voß. Interoperable smart card data management in public mass transit. *Public Transport*, 11(3):523–548, 2019.

[45] Sean Óg Crudden and Simon Berrebi. An open-source framework to implement kalman filter bus arrival predictions. *Networks and Spatial Economics*, 23(2):429–443, 2023.

[46] Graham Currie and Mahmoud Mesbah. Visualization of Geographical Information System and Automatic Vehicle Location Data to Explore Transit Performance. *Transportation Research Record*, 2216(1):59–66, 2011. Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[47] Alan Danaher, James Wensley, Amy Dunham, Ted Orosz, Ryan Avery, Kenneth Cobb, Kari Watkins, Carly Queen, Simon Berrebi, Marlene Connor, and Jim McLaughlin. Minutes Matter: A Bus Transit Service Reliability Guidebook. Transit Cooperative Research Program 215, National Academies of Sciences, Engineering, and Medicine, 2020.

[48] Juan de Oña, Rocio de Oña, and Francisco Calvo. A Classification Tree Approach to Identify Key Factors of Transit Service Quality. *Expert Systems with Applications*, 39(12):11164–11171, 2012.

[49] A. Deebika Shree, J. Anusuya, and S. Malathy. Real time bus tracking and location updation system. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pages 242–245, 2019.

[50] Guy Desaulniers and Mark D Hickman. Public transit. *Handbooks in operations research and management science*, 14:69–127, 2007.

[51] Erica E Eggleton, Ryan Carlin, Harrison J Goldwyn, Lynna Truong, Danny Ilioiu, and Daniel T Schwartz. Route_dynamics–an open source python package for analyzing route-specific energy and power requirements that influence battery performance in electric bus fleets. *Available at SSRN 4158241*, 2022.

[52] Ahmed M El-Geneidy, Jessica Horning, and Kevin J Krizek. Analyzing transit service reliability using detailed data from automatic vehicular locator systems. *Journal of Advanced Transportation*, 45(1):66–79, 2011.

[53] Tom Elliott and Thomas Lumley. Modelling the travel time of transit vehicles in real-time through a gtfs-based road network using gps vehicle locations. *Australian & New Zealand Journal of Statistics*, 62(2):153–167, 2020.

[54] Esri. Public transit tools. https://github.com/Esri/public-transit-tools. Accessed: 2023-09-13.

[55] Eurotech. Eurotech's duramar and king county metro - bringing rapid service and reliability to riders, 2015. Accessed: 2024-04-01.

[56] Wei Fan and Zegeye Gurmu. Dynamic Travel Time Prediction Models for Buses Using Only GPS Data. *International Journal of Transportation Science and Technology*, 4(4):353–366, 2015.

[57] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. Ssml: Self-supervised meta-learner for en route travel time estimation at baidu maps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2840–2848, 2021.

[58] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2697–2705, 2020.

[59] Kona Farry. Pantograph: Puget sound map. https://pantographapp.com/pugetsound/map. Accessed: 2023-09-13.

[60] Wei Feng and Miguel Figliozzi. Developing a Bus Service Reliability Evaluation and Visualization Framework Using Archived AVL / APC Data. In *Developing a Bus Service Reliability Evaluation and Visualization Framework Using Archived AVL / APC Data*, Santa Barbara, California, United States, 2012. Institute of Transportation Engineers.

[61] Wei Feng and Miguel Figliozzi. Empirical Findings of Bus Bunching Distributions and Attributes Using Archived AVL/APC Bus Data. In *Empirical Findings of Bus Bunching Distributions and Attributes Using Archived AVL/APC Bus Data*, pages 4330–4341. American Society of Civil Engineers, 2012.

[62] Michael Fernandes, Logan Walls, Sean Munson, Jessica Hullman, and Matthew Kay. Uncertainty displays using quantile dotplots or cdfs improve transit decision-making. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[63] Brian Ferris, Kari Watkins, and Alan Borning. OneBusAway: A Transit Traveler Information System. In *OneBusAway: A Transit Traveler Information System*, volume 35 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Book Series*, pages 92–106. LNICST, 2010.

[64] Code for America. Transitmix: Design your perfect bus system. https://www.govloop.com/community/blog/transitmix-design-your-perfect-bus-system/. Accessed: 2023-09-13.

[65] European Committee for Standardization. Netex overview. https://netex-cen.eu/overview/. Accessed: 2023-09-13.

[66] Center for Urban Transportation Research. awesome-transit. https://github.com/CUTR-at-USF/awesome-transit. Accessed: 2023-09-13.

[67] Renaud Foucart and Qian Cher Li. The role of technology standards in product innovation: Theory and evidence from uk manufacturing firms. *Research Policy*, 50(2):104157, 2021.

[68] Anaissia Franca. *Electricity consumption and battery lifespan estimation for transit electric buses: drivetrain simulations and electrochemical modelling*. PhD thesis, University of Victoria, 2018.

[69] Yonah Freemark and Lindiwe Rennert. Surmounting the fiscal cliff. *Urban Institute*, 2023.

[70] Liping Fu and Laurence R. Rilett. Real-Time Estimation of Incident Delay in Dynamic and Stochastic Networks:. *Transportation Research Record*, 1603(1):99–105, 1997. Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[71] Peter G. Furth, Brendon Hemily, Theo H. J. Muller, and James G. Strathman. Using Archived AVL-APC Data to Improve Transit Performance and Management. Technical Report 113, National Academies of Sciences, Engineering, and Medicine, 2006.

[72] Michal S Gal and Daniel L Rubinfeld. Data standardization. *NYUL Rev.*, 94:737, 2019.

[73] Birgitta Gatersleben and David Uzzell. Affective appraisals of the daily commutecomparing perceptions of drivers, cyclists, walkers, and users of public transport. *Environment and Behavior*, 39:416–431, 05 2007.

[74] Liping Ge, Malek Sarhani, Stefan Voss, and Lin Xie. Review of transit data sources: Potentials, challenges and complementarity. *Sustainability*, 13:11450, 10 2021.

[75] Dean Gesch, Michael Oimoen, Susan Greenlee, Charles Nelson, Michael Steuck, and Dean Tyler. The national elevation dataset. *Photogrammetric engineering and remote sensing*, 68(1):5–32, 2002.

[76] Antoine Giraud, Martin Trépanier, Catherine Morency, and Félix Légaré. Data fusion of apc, smart card and gtfs to visualize public transit use. Technical report, CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise ..., 2016.

[77] John M. Gleason and Darold T. Barnum. Toward Valid Measures of Public Sector Productivity: Performance Measures in Urban Transit. *Management Science*, 28(4):379–386, 1982. Publisher: INFORMS.

[78] Gautam Raj Godavarthi, Ravi Sekhar Chalumuri, and Senathipathi Velmurugun. Measuring the Performance of Bus Rapid-Transit Corridors Based on Volume by Capacity Ratio. *Journal of Transportation Engineering*, 140(10):04014049, 2014. Publisher: American Society of Civil Engineers.

[79] Jodi Godfrey, Austin Sipiora, Lisa Staes, et al. Guidebook for deploying battery electric buses. Technical report, United States. Department of Transportation. Federal Transit Administration, 2023.

[80] Juan Godfrid, Pablo Radnic, Alejandro Vaisman, and Esteban Zimányi. Analyzing public transport in the city of buenos aires with mobilitydb. *Public Transport*, 14(2):287–321, 2022.

[81] Google. Gtfs realtime overview. `https://developers.google.com/transit/gtfs-realtime`. Accessed: 2023-09-13.

[82] Google. Gtfs static overview. `https://developers.google.com/transit/gtfs`. Accessed: 2023-09-13.

[83] Piotr Gramacki, Kacper Leśniara, Kamil Raczycki, Szymon Woźniak, Marcin Przymus, and Piotr Szymański. Srai: Towards standardization of geospatial ai. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 43–52, 2023.

[84] Piotr Gramacki, Szymon Woźniak, and Piotr Szymański. Gtfs2vec: Learning gtfs embeddings for comparing public transport offer in microregions. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data*, GeoSearch'21, page 5–12, New York, NY, USA, 2021. Association for Computing Machinery.

[85] PTV Group. Ptv visum. https://www.ptvgroup.com/en-us/products/ptv-visum. Accessed: 2023-09-13.

[86] Mark E. Hallenbeck, Eric J. Howard, Dmitri Zyuzin, R.P. Avery, and Miki E. Verma. Use of electronic fare transaction data for corridor planning. In *Use of electronic fare transaction data for corridor planning.*, 2017.

[87] Erfan Hassannayebi, Ali Farjad, Alireza Azadnia, Mehrdad Javidi, and Raghavendra Chunduri. A data analytics framework for reliable bus arrival time prediction using artificial neural networks. *International Journal of Data Science and Analytics*, pages 1–20, 2023.

[88] Yuye He, Sebastien Blandin, Laura Wynter, and Barry Trager. Analysis and Real-Time Prediction of Local Incident Impact on Transportation Networks. In *Analysis and Real-Time Prediction of Local Incident Impact on Transportation Networks*, pages 158–166, Shenzhen, China, 2014. IEEE.

[89] Cristina Heghedus, Antorweep Chakravorty, and Chunming Rong. Neural Network Frameworks. Comparison on Public Transportation Prediction. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 842–849, 2019.

[90] Tomislav Hengl, L Leal Parente, Josip Krizan, and Carmelo Bonannella. Continental europe digital terrain model at 30 m resolution based on gedi, icesat-2, aw3d, glo-30, eudem, merit dem and background layers. *Version Dataset v3. 0. Zenodo*, 2020.

[91] Kristian Henrickson. *A Framework for Understanding and Addressing Bias and Sparsity in Mobile Location-Based Traffic Data*. PhD thesis, University of Washington, 2018.

[92] Edward L Hillsman, Sean J Barbeau, et al. Enabling cost-effective multimodal trip planners through open transit data. Technical report, National Center for Transit Research (US), 2011.

[93] Kei Hiroi, Hitomi Imai, and Nobuo Kawaguchi. Dynamic arrival time estimation model and visualization method for bus traffic. *Intelligent Transport Systems for Everyone's Mobility*, pages 155–173, 2019.

[94] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. HetETA: Heterogeneous Information Network Embedding for Estimating Time of Arrival. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 2444–2454, New York, NY, USA, 2020. Association for Computing Machinery.

[95] Larson Transportation Institute. About the bus research and testing center. https://www.altoonabustest.psu.edu/about/index.aspx. Accessed: 2023-12-28.

[96] Shah Mohammad Mominul Islam, Arshad Adam Salema, Mohammed Zeehan Saleheen, and Joanne Mun Yee Lim. The influence of shifting the electric bus charging routine on the techno-economic performance of a solar-powered bus depot. *Energy*, 239:122316, 2022.

[97] Jerald Jariyasunant, Daniel B Work, Branko Kerkez, Raja Sengupta, Steven Glaser, and Alexandre Bayen. Mobile transit trip planning with real-time data. *eScholarship*, 2011.

[98] Ranhee Jeong and R Rilett. Bus arrival time prediction using artificial neural network model. In *Proceedings. The 7th international IEEE conference on intelligent transportation systems (IEEE Cat. No. 04TH8749)*, pages 988–993. IEEE, 2004.

[99] Takuya Kawatani, Tsubasa Yamaguchi, Yuta Sato, Ryotaro Maita, and Tsunenori Mine. Prediction of Bus Travel Time over Intervals between Pairs of Adjacent Bus Stops Using City Bus Probe Data. *International Journal of Intelligent Transportation Systems Research*, 19(2):456–467, 2021.

[100] Hyun Kim and Yena Song. An Integrated Measure of Accessibility and Reliability of Mass Transit Systems. *Transportation*, 45(4):1075–1100, 2018.

[101] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[102] Gergely Kocsis and Imre Varga. Extracting mass transportation networks from general transit feed specification datasets. In *COMPLEXIS*, pages 85–91, 2022.

[103] Andrew J Kotz, Eric Miller, Andrea Watson, and Kenneth J Kelly. Transit bus electrification evaluation from gps speed traces. In *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, pages 1069–1074. IEEE, 2020.

[104] Parviz A Koushki, Mohammed A Ali, Bapatla P Chandrasekhar, and Mohammad Al-Sarawi. Exposure to noise inside transit buses in kuwait: Measurements and passenger attitudes. *Transport Reviews*, 22(3):295–308, 2002.

[105] Anand V Kulkarni, Rekha R Sapre, and Chirag P Sonchal. Gps based methodology for drive cycle determination. *SAE transactions*, pages 233–239, 2005.

[106] Narumon Kunama, Mudtana Worapan, Santi Phithakkitnukoon, and Merkebe Demissie. Gtfs-viz: Tool for preprocessing and visualizing gtfs data. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 388–396, 2017.

[107] Abdullah Kurkcu, Fabio Miranda, Kaan Ozbay, and Claudio T. Silva. Data visualization tool for monitoring transit operation and performance. In *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 598–603, 2017.

[108] Jinxuan Lai, Lei Yu, Guohua Song, Pei Guo, and Xumei Chen. Development of city-specific driving cycles for transit buses based on vsp distributions: Case of beijing. *Journal of Transportation Engineering*, 139(7):749–757, 2013.

[109] Antti Lajunen. Lifecycle costs and charging requirements of electric buses with different charging methods. *Journal of cleaner production*, 172:56–67, 2018.

[110] Wuwei Lan, Yanyan Xu, and Bin Zhao. Travel Time Estimation without Road Networks: An Urban Morphological Layout Representation Approach. http://arxiv.org/abs/1907.03381, 2019. arXiv:1907.03381 [cs].

[111] Catherine T Lawson, Alex Muro, and Eric Krans. Forecasting bus ridership using a "blended approach". *Transportation*, 48:617–641, 2021.

[112] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[113] Jiezhang Li, Wanyi Zhou, Zebin Chen, and Yue-Jiao Gong. Geo-Attention Network for Traffic Condition Prediction and Travel Time Estimation. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '21, pages 654–657, New York, NY, USA, 2021. Association for Computing Machinery.

[114] Dea Lierop, Madhav Badami, and Ahmed El-Geneidy. Transport reviews what influences satisfaction and loyalty in public transport? a review of the literature what influences satisfaction and loyalty in public transport? a review of the literature. *Transport Reviews*, 38:52–72, 01 2018.

[115] Wei-Hua Lin and Robert L. Bertini. Modeling Schedule Recovery Processes in Transit Operations for Bus Arrival Time Prediction. *Journal of Advanced Transportation*, 38(3):347–365, 2010.

[116] Xi Lin, Yequan Wang, Xiaokui Xiao, Zengxiang Li, and Sourav S. Bhowmick. Path Travel Time Estimation using Attribute-related Hybrid Trajectories Network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 1973–1982, New York, NY, USA, 2019. Association for Computing Machinery.

[117] Luying Liu, Andrew Kotz, Aditya Salapaka, Eric Miller, and William F Northrop. Impact of time-varying passenger loading on conventional and electrified transit bus energy consumption. *Transportation Research Record*, 2673(10):632–640, 2019.

[118] Yanjun Liu, Hui Zhang, Jianmin Jia, Baiying Shi, and Wei Wang. Understanding urban bus travel time: Statistical analysis and a deep learning prediction. *International Journal of Modern Physics B*, 37(04):2350034, 2023. Publisher: World Scientific Publishing Co.

[119] Sean Lopp, Eric Wood, and Adam Duran. Evaluating the impact of road grade on simulated commercial vehicle fuel economy using real-world drive cycles. Technical report, SAE Technical Paper, 2015.

[120] Kai Lu, Jiangtao Liu, Xuesong Simon Zhou, and Baoming Han. A review of big data applications in urban transit systems. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–18, 02 2020.

[121] Jiaman Ma, Jeffrey Chan, Sutharshan Rajasegarar, and Christopher Leckie. Multi-attention graph neural networks for city-wide bus travel time estimation using limited data. *Expert Systems with Applications*, 202:117057, 2022.

[122] Shuai Ma, Modi Jiang, Peng Tao, Chengyi Song, Jianbo Wu, Jun Wang, Tao Deng, and Wen Shang. Temperature effect and thermal impact in lithium-ion batteries: A review. *Progress in Natural Science: Materials International*, 28(6):653–666, 2018.

[123] Shuai Ma, Meng Lin, Tzu-En Lin, Tian Lan, Xun Liao, François Maréchal, Yongping Yang, Changqing Dong, Ligang Wang, et al. Fuel cell-battery hybrid systems for mobility and off-grid applications: A review. *Renewable and Sustainable Energy Reviews*, 135:110119, 2021.

[124] Michael Mandelzys and Bruce Hellinga. Identifying Causes of Performance Issues in Bus Schedule Adherence with Automatic Vehicle Location and Passenger Count Data:.

Transportation Research Record, 2143(1):9–15, 2010. Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[125] Jean Damascène Mazimpaka and Sabine Timpf. Trajectory data mining: A review of methods and applications. *Journal of spatial information science*, 2016(13):61–99, 2016.

[126] Daniel L Mendoza, Martin P Buchert, and John C Lin. Modeling net effects of transit operations on vehicle miles traveled, fuel consumption, carbon dioxide, and criteria air pollutant emissions in a mid-size us metro area: Findings from salt lake city, ut. *Environmental Research Communications*, 1(9):091002, 2019.

[127] Parametrix Microgrid Labs. King county metro transit system electrification plan - phase 1 battery electric bus service energy modeling and analysis report. Technical report, King County Metro, 2024.

[128] MobilityData. General transit feed specification. https://gtfs.org/resources/visualizations/. Accessed: 2023-09-13.

[129] MobilityData. General transit feed specification - multimodal. https://gtfs.org/resources/multimodal/. Accessed: 2023-09-13.

[130] Richard Neitzel, Robyn RM Gershon, Marina Zeltser, Allison Canton, and Muhammad Akram. Noise levels associated with new york city's mass transit systems. *American journal of public health*, 99(8):1393–1399, 2009.

[131] Jorge Adorno Nieves, Brian D. Torres Alvarado, Sean J. Barbeau, and Miguel A. Labrador. Evaluating Models for Estimating Schedule Deviation in Public Transit. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 275–280, 2021.

[132] Thummaporn Nimpanomprasert, Lin Xie, and Natalia Kliewer. Comparing two hybrid neural network models to predict real-world bus travel time. *Transportation Research Procedia*, 62:393–400, 2022.

[133] Gabriel Nowacki. Development and standardization of intelligent transport systems. *International Journal on Marine Navigation and Safety of Sea Transportation*, 6(3):403–411, 2012.

[134] Khandker M. Nurul Habib, Lina Kattan, and Md Tazul Islam. Model of Personal Attitudes Towards Transit Service Quality. *Journal of Advanced Transportation*, 45(Modeling and Optimization of Transportation Systems):271–285, 2011.

[135] Society of Automotive Engineers. J-3105: Electric vehicle power transfer system using conductive automated connection devices. https://www.sae.org/standards/content/j3105_202001/. Accessed: 2023-09-13.

[136] US Department of Transportation. Battery electric bus basics. https://www.transportation.gov/urban-e-mobility-toolkit/e-mobility-basics/bus. Accessed: 2023-09-13.

[137] Bureau of Transportation Statistics. National transportation statistics. Technical report, Bureau of Transportation Statistics, 2021.

[138] OpenMobilityData. Transitfeeds, 2024.

[139] OpenTopography. Alos world 3d 30 meter dem, 2021. Accessed: 2024-04-01.

[140] OpenTripPlanner. Opentripplanner, multimodal trip planning. https://www.opentripplanner.org/. Accessed: 2023-09-13.

[141] OpenTripPlanner. Opentripplanner deployments worldwide, 2024. Accessed: 2024-04-01.

[142] Optibus. Optibus for operators. https://www.optibus.com/solutions/optibus-for-operators/. Accessed: 2023-09-13.

[143] Yuming Ou. AI for Real-Time Bus Travel Time Prediction in Traffic Congestion Management. In Fang Chen and Jianlong Zhou, editors, *Humanity Driven AI: Productivity, Well-being, Sustainability and Partnership*, pages 63–84. Springer International Publishing, 2022.

[144] Doug J Parker. *AVL systems for bus transit: Update*, volume 73. Transportation Research Board, 2008.

[145] Jiankun Peng, Jiwan Jiang, Fan Ding, and Huachun Tan. Development of driving cycle construction for hybrid electric bus: A case study in zhengzhou, china. *Sustainability*, 12(17):7188, 2020.

[146] Niklas Christoffer Petersen, Filipe Rodrigues, and Francisco Camara Pereira. Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Systems with Applications*, 120:426–435, 2019.

[147] Xidong Pi, Mark Egge, Jackson Whitmore, Zhen Sean Qian, and Amy Silbermann. Understanding transit system performance using avl-apc data: An analytics platform with case studies for the pittsburgh region. *Journal of Public Transportation*, 21(2):19–40, 2018.

[148] Postsavee Prommaharaj, Santi Phithakkitnukoon, Merkebe Getachew Demissie, Lina Kattan, and Carlo Ratti. Visualizing public transit system operation with gtfs data: A case study of calgary, canada. *Heliyon*, 6(4), 2020.

[149] Mohammed A. Quddus, Washington Y. Ochieng, and Robert B. Noland. Current Map-Matching Algorithms for Transport Applications: State-of-the Art and Future Research Directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.

[150] R5R. R5 - rapid realistic routing. https://r5py.readthedocs.io/en/stable/. Accessed: 2023-09-13.

[151] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud JG Van Sloun, and Yonina C Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

[152] Yuecheng Rong, Jun Liu, Zhilin Xu, Jian Ding, Chuangming Zhang, and Jiaxiang Gao. Buswte: Realtime bus waiting time estimation of gps missing via multi-task learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 554–570. Springer, 2022.

[153] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

[154] S Seki, C Hendrickson, and D Stine. Which alternative fuel technology is best for transit buses. Technical report, Carnegie Mellon University, 2016.

[155] Faruk Serin, Yigit Alisan, and Adnan Kece. Hybrid time series forecasting methods for travel time prediction. *Physica A: Statistical Mechanics and its Applications*, 579:126134, 2021.

[156] Arun Sharma, Zhe Jiang, and Shashi Shekhar. Spatiotemporal data mining: A survey. *arXiv preprint arXiv:2206.12753*, 2022.

[157] Yibin Shen, Cheqing Jin, Jiaxun Hua, and Dingjiang Huang. Ttpnet: A neural network for travel time prediction based on tensor decomposition and graph embedding. *IEEE Transactions on Knowledge and Data Engineering*, 34(9):4514–4526, 2020.

[158] Yindong Shen, Jia Xu, and Zhongyi Zeng. Public transit planning and scheduling based on avl data in china. *International Transactions in Operational Research*, 23(6):1089–1111, 2016.

[159] Jiuh-Biing Sheu, Yi-Hwa Chou, and Allen Chen. Stochastic Modeling and Real-Time Prediction of Incident Effects on Surface Street Traffic Congestion. *Applied Mathematical Modelling*, 28(5):445–468, 2004. Publisher: Elsevier.

[160] Daniel Steiner, Hartwig Hochmair, and Gernot Paulus. Quality Assessment of Open Realtime Data for Public Transportation in the Netherlands. *Journal for Geographic Information Science*, 3(1):579–588, 2015.

[161] James G. Strathman, Kenneth J. Dueker, Thomas Kimpel, Rick Gerhart, Ken Turner, Pete Taylor, Steve Callas, David Griffin, and Janet Hopper. Automated Bus Dispatching, Operations Control, and Service Reliability: Baseline Analysis:. *Transportation Research Record*, 1666(1):28–36, 1999. Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[162] Gregory Tassey. Standardization in technology-based markets. *Research policy*, 29(4-5):587–602, 2000.

[163] Hing Yan Tong and KaWai Ng. Development of bus driving cycles using a cost effective data collection approach. *Sustainable Cities and Society*, 69:102854, 2021.

[164] Luan Tran, Min Y. Mun, Matthew Lim, Jonah Yamato, Nathan Huh, and Cyrus Shahabi. DeepTRANS: a deep learning system for public bus travel time estimation using traffic forecasting. *Proceedings of the VLDB Endowment*, 13(12):2957–2960, 2020.

[165] Transloc. Transloc fixed route bus transit. https://transloc.com/solutions/fixed-route/. Accessed: 2023-09-13.

[166] Transmodel. Siri standard. https://www.transmodel-cen.eu/siri-standard/. Accessed: 2023-09-13.

[167] Wichai Treethidtaphat, Wasan Pattara-Atikom, and Sippakorn Khaimook. Bus arrival time prediction at any distance of bus route using deep neural network model. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 988–992. IEEE, 2017.

[168] Tripspark. Tripspark fixed route software. https://www.tripspark.com/fixed-route-software/. Accessed: 2023-09-13.

[169] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[170] Via. Geospatial tools. https://ridewithvia.com/solutions/remix/explore. Accessed: 2023-09-13.

[171] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, 04 2018.

[172] Senzhang Wang, Jiannong Cao, and S Yu Philip. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 34(8):3681–3700, 2020.

[173] Kari Watkins. Onebusaway: A ten year retrospective of an open source real-time information system. *CASPT*, 2018.

[174] Kari Edison Watkins, Brian Ferris, Alan Borning, G. Scott Rutherford, and David Layton. Where Is My Bus? Impact of Mobile Real-Time Information on the Perceived and Actual Wait Time of Transit Riders. *Transportation Research Part A: Policy and Practice*, 45(8):839–848, 2011.

[175] Kari Edison Watkins, Brian Ferris, and G Scott Rutherford. Explore: An attraction search tool for transit trip planning. *Journal of Public Transportation*, 13(4):111–128, 2010.

[176] Wehrmeyer. Mapnificent. https://www.mapnificent.net/. Accessed: 2023-09-13.

[177] Nate Wessel and Michael J. Widener. Discovering the Space–Time Dimensions of Schedule Padding and Delay from GTFS and Real-Time Transit Data. *Journal of Geographical Systems*, 19(1):93–107, 2017.

[178] Graham Wilson and Curran Crawford. Fleet-wide bus drivecycle modelling and energy estimation using real-time transit feed data. *Available at SSRN 3982560*, 2021.

[179] Maxwell Woody, Maryam Arbabzadeh, Geoffrey M Lewis, Gregory A Keoleian, and Anna Stefanopoulou. Strategies to limit degradation and maximize li-ion battery service lifetime-critical review and guidance for stakeholders. *Journal of Energy Storage*, 28:101231, 2020.

[180] Szymon Woźniak and Piotr Szymański. Hex2vec: Context-aware embedding h3 hexagons with openstreetmap tags. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 61–71, 2021.

[181] Jianqing Wu, Bo Du, Zengyang Gong, Qiang Wu, Jun Shen, Luping Zhou, and Chen Cai. A gtfs data acquisition and processing framework and its application to train delay prediction. *International Journal of Transportation Science and Technology*, 12(1):201–216, 2023.

[182] Jianqing Wu, Qiang Wu, Jun Shen, and Chen Cai. Towards Attention-Based Convolutional Long Short-Term Memory for Travel Time Prediction of Bus Journeys. *Sensors*, 20(12):3354, 2020. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[183] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The world wide web conference*, pages 2181–2191, 2019.

[184] Bin Yu and Zhongzhen Yang. A Dynamic Holding Strategy in Public Transit Systems with Real-Time Information. *Applied Intelligence*, 31(1):69–80, 2009.

[185] Yuan Yuan, Chunfu Shao, Zhichao Cao, Zhaocheng He, Changsheng Zhu, Yimin Wang, and Vlon Jang. Bus Dynamic Travel Time Prediction: Using a Deep Feature Extraction Framework Based on RNN and DNN. *Electronics*, 9(11):1876, 2020. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[186] Wong Seng Yue, Koh Keng Chye, and Cheong Wan Hoy. Towards Smart Mobility in Urban Spaces: Bus Tracking and Information Application. In *AIP Conference Proceedings*, volume 1891, page 020145. AIP, 2017.

[187] Tianchi Zhang, Mei-Hwa Chen, and Catherine Lawson. General transit feed specification data visualization. In *2014 22nd International Conference on Geoinformatics*, pages 1–6, 2014.

[188] Xinhuan Zhang, Les Lauber, Hongjie Liu, Junqing Shi, Meili Xie, and Yuran Pan. Travel time prediction of urban public transportation based on detection of single routes. *PLOS ONE*, 17(1):e0262535, 2022. Publisher: Public Library of Science.

[189] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

[190] Tuqiang Zhou, Wanting Wu, Liqun Peng, Mingyang Zhang, Zhixiong Li, Yubing Xiong, and Yuelong Bai. Evaluation of urban bus service reliability on variable time horizons using a hybrid deep learning method. *Reliability Engineering & System Safety*, 217:108090, 2022.

[191] Lijun Zhu, Weibo Wei, Yang Xia, and Liping Fu. Bus Travel Time Prediction Based on Multi-Source Data Fusion. In *Proceedings of the 5th International Conference on Big Data Technologies*, ICBDT '22, pages 97–102, New York, NY, USA, 2022. Association for Computing Machinery.

.1 Appendix A

Table 1: Academic Bus and Bus-Adjacent Travel Time Prediction Models

Reference	Model	Bus-Specific	Full-Network	Multiple-Network	Data Scope	Data Source
Open Source						
[171]	DeepTTE			X	Chengdu; 9 million; Beijing; 3 million	Taxi
[110]	DeepI2T		X	X	Shanghai; 15 million, Porto; 1.3 million	Taxi, Satellite
[84]	gtfs2vec	X	X	X	48 European Cities	GTFS
[146]	ConvLSTM	X			Copenhagen Movia; 1.2 million; 1 route	AVL
[53]	Particle Filter	X	X		Auckland	GTFS-RT/API
[36]	HierETA			X	Beijing, Guangzhou	Didi
[57]	ER-TTE				Taiyuan, Huizhou, Hefei; 2.5 million	Baidu Maps
[157]	TTPNet			X	Beijing; 3.3 million Shanghai; 9.7 million	Taxi
[94]	HetETA				Shenyang; 4.8 million	Didi
Closed Source						
[16]	Bustr	X	X	X	Undisclosed; 40 million records	GTFS-RT/API
[152]	BusWTE	X	X	X	Xiamen; 4 million Nanjing; 4 million	Taxi
[13]	Various ML	X			Tumakuru; 1 route; 600 trajectories	AVL
[190]	VMD-LSTM	X			Suzhou; 1 route; 387 trajectories	AVL
[121]	MAGTTE	X			Xi'an; 3 routes	POI
[132]	Kalman Filter+LSTM	X			HOCHBAHN Germany; 4 routes	Agency
[164]	DeepTRANS	X	X		LAMTA; 62,000 trajectories	AVL, Loop
[113]	Geo-Attention				Shenzen; 200,000 trajectories	Gaia
[116]	AtHy-TNet				Singapore; 11 million trajectories	Bus Card, Taxi
[191]	MSDFTTP			X	Asia; 17,000 trajectories	Travel Survey, Weather
[56]	HA, KF, LR, ANN	X			Macae; 1 million records; 1 route	AVL
[42]	Time Series Decomposition	X	X	X	Rome, Lyiv	AVL
[188]	Kalman Filter	X			Madison; 1 route; 2,000 records	Bus Card, AVL
[89]	BPNN, LSTM	X	X		Stavanger; 3,000 trips	Agency
[143]	HA, TS, REG, KAL, LSTM	X			Sydney; 1 route	GTFS-RT/API
[182]	ConvLSTM+Attention	X	X		Sydney; all routes	GTFS-RT/API
[131]	DT, KNN, KF	X	X	X	Tampa, St Petersburg	GTFS-RT/API
[99]	GBDT	X	X		Kyushu University; 180,000 records	Sensor
[118]	Kalman Filter+LSTM, GBDT	X			Jinan; 1.3 million records; 4 routes	AVL
[155]	3-Layer Time Series	X			Istanbul; 1 route	Agency
[185]	RNN+Attention	X		X	Guangzhou and Shenzhen; 1 route	Taxi, AVL

.2 Appendix B

Table 2: International Transit Networks with Open Bus Feeds

Provider	Min Lon	Max Lon	Min Lat	Max Lat	Timezone	EPSG
London Transit Commission	-81.36311	-81.137591	42.905244	43.051188	America/Toronto	32617
Barrie Transit	-79.74063237	-79.61089569	44.3218044	44.42020676	America/Toronto	32617
Big Blue Bus	-118.549205	-118.237266	33.929498	34.075669	America/Los_Angeles	32611
Mountain View Transportation Management Association (MVgo)	-122.111591	-122.0475836	37.3876555	37.4314288	America/Los_Angeles	32610
Capital Metro	-97.9911	-97.370389	30.147252	30.587897	America/Chicago	32614
Regional Transportation District (RTD)	-105.584149	-104.669533	39.456877	40.206556	America/Denver	32613
Metro St. Louis	-90.662094	-89.874657	38.469747	38.825646	America/Chicago	32615
Intercity Transit	-122.977127	-122.483109	46.93304	47.16091	America/Los_Angeles	32610
Duluth Transit	-92.290788	-92.009608	46.654703	46.855417	America/Chicago	32615
Votran	-81.506774	-80.847351	28.856002	29.383995	America/New_York	32617
Nashville Metropolitan Transit Authority (Nashville MTA)	-87.379122	-86.29724	35.7416	36.525005	America/Chicago	32616
Transit Authority of River City (TARC)	-85.904275	-85.502994	38.056361	38.383253	America/Kentucky/Louisville	32616
Metropolitan Atlanta Rapid Transit Authority (MARTA)	-84.669803	-84.083398	33.432372	34.105822	America/New_York	32616
Springfield Mass Transit District (SMTD)	-89.75837	-89.527032	39.667827	39.898621	America/Chicago	32616
Metro Transit	-89.563863	-89.244818	42.987718	43.176537	America/Chicago	32616
Port Authority of Allegheny County	-80.258865	-79.70545	40.273078	40.667597	America/New_York	32617
Massachusetts Bay Transportation Authority (MBTA)	-71.848488	-70.276583	41.581289	42.797837	America/New_York	32619
Arlington Transit	-77.162277	-77.049105	38.839365	38.925147	America/New_York	32618
Rochester-Genesee Regional Transportation Authority (RGRTA)	-78.141093	-76.99341	42.526438	43.297343	America/New_York	32617
Adelaide Metro	138.445316	139.036087	-35.339223	-34.571043	Australia/Adelaide	32754
TransLink Sunbus Cairns	145.663264	145.786494	-17.104064	-16.743567	Australia/Brisbane	32755
BC Transit (Victoria Regional Transit System)	-123.795435	-123.275822	48.33716	48.697757	America/Vancouver	32610
Edmonton Transit System	-113.934884	-113.218139	53.302688	53.704407	America/Edmonton	32612
Saskatoon Transit	-106.760746	-106.550137	52.078661	52.202189	America/Regina	32613
Hamilton Street Railway	-80.031354	-79.692514	43.156343	43.326073	America/Toronto	32617
MiWay	-79.797795	-79.512874	43.469292	43.838069	America/Toronto	32617
Kingston Transit	-76.66738	-76.43906	44.2175	44.27897	America/Toronto	32618
Halifax Transit	-63.858243	-63.28321	44.565098	44.886253	America/Halifax	32620
Thunder Bay Transit	-89.38849	-89.1819	48.34626	48.47868	America/Toronto	32616
Port Alberni	-124.85047	-124.78265	49.21847	49.27743	America/Vancouver	32610
Commerce Municipal Bus Lines	-118.264252	-118.127642	33.965568	34.063075	America/Los_Angeles	32611
Roma Servizi per la Mobilità	12.107767	12.789177	41.648593	42.4415	Europe/Rome	32633
Stanislaus Regional Transit Authority (StanRTA)	-121.898961218	-120.74953	37.25713	37.954866	America/Los_Angeles	32610
Valley Metro	-112.872185	-111.660913	32.382706	33.713075	America/Phoenix	32612
Long Beach Transit	-118.447726	-118.063516	33.742939	34.069347	America/Los_Angeles	32611
Hillsborough Area Regional Transit	-82.584777	-82.210817	27.70568	28.195743	America/New_York	32617
Greater Cleveland RTA	-81.96736	-81.437811	41.227828	41.637051	America/New_York	32617
Central Ohio Transit Authority	-83.162651	-82.726672	39.830163	40.150506	America/New_York	32617
Lane Transit District	-123.3561683	-122.1152746	43.7849362	44.2226569	America/Los_Angeles	32610
Rhode Island Public Transit Authority	-71.829808	-71.154292	41.372687	42.016993	America/New_York	32619
Denton County Transportation Authority	-97.31863987	-96.926476	32.92199276	33.253687	America/Chicago	32614
VIA Metropolitan Transit	-98.711	-98.306921	29.27663	29.650497	America/Chicago	32614
Community Transit	-122.38471	-121.601001	47.59201	48.278539	America/Los_Angeles	32610
OVapi	3.383789	13.227539	51.272226	53.742214	Europe/Oslo	32630
Translink South East Queensland	152.124238	153.543374	-28.172196	-26.159390	Australia/Brisbane	32756
Translink North Stradbroke Island	153.401170	153.543723	-27.503240	-27.394988	Australia/Brisbane	32756
Translink Maryborough Hervey Bay	152.558177	152.908899	-25.560525	-25.183988	Australia/Brisbane	32756
Translink Bowen	148.225334	148.265372	-20.018563	-19.970855	Australia/Brisbane	32755
Translink Innisfail	145.988544	146.076331	-17.543952	-17.490769	Australia/Brisbane	32755