Artificial Intelligence for Smart Transportation

MICHAEL WILBUR, Vanderbilt University, USA
AMUTHEEZAN SIVAGNANAM, Pennsylvania State University, USA
AFIYA AYMAN, Pennsylvania State University, USA
SAMITHA SAMARANAYEKE, Cornell University, USA
ABHISHEK DUBEY, Vanderbilt University, USA
ARON LASZKA, Pennsylvania State University, USA

1 INTRODUCTION

There are more than 7,000 public transit agencies in the U.S. (and many more private agencies), and together, they are responsible for serving 60 billion passenger miles each year. Additionally, new on-demand modalities including ride-share, bike-share, and e-scooters have been introduced in recent years and transformed the transportation landscape in urban environments. A well-functioning transit system fosters the growth and expansion of businesses, distributes social and economic benefits, and links the capabilities of community members, thereby enhancing what they can accomplish as a society [6, 11, 15]. However, the explosion in transportation options and the complicated relationship between public and private offerings present myriad new challenges in the design and operation of these systems. There are also complex, and often competing, operational objectives that complicate the implementation of efficient services. Since affordable public transit services are the backbones of many communities, solving these problems and understanding state-of-the-art methods for AI-driven smart transportation has the potential to strengthen urban communities, address the climate challenge, and foster equitable growth.

Fundamentally, the design of a well-functioning transit system requires solving complex combinatorial optimization problems related to planning and real-time operations. These problems span many well studied fields, from classical line planning to offline and online vehicle routing problems (VRPs). While there are many ways to assess the performance of smart transportation systems, we largely focus on evaluating these systems in the context of optimizing *utilization* (i.e. ridership) and *efficiency* (i.e. reducing operational costs). Increasing utilization requires learning mobility patterns over wide geographical areas and adapting systems to better meet the demand for mobility. It also requires flexible mobility options such as on-demand and multi-modal transportation to adapt to demand in real-time and thus better serve potential passengers. Additionally, more efficient systems alleviate the impact on the environment by reducing emissions and can free resources by reducing costs. Lastly, *coverage* is an important consideration that often competes with utilization and efficiency. Therefore it is important to discuss optimization in the context of *ridership* versus *coverage*, the latter of which is an important consideration in the design of equitable and fair transportation. While the problem of ridership versus coverage is open-ended, there are ways in which these dimensions can be modelled such that transit agencies can better optimize over the objectives that matter most to them.

Efficient transportation systems require making decisions in real-time over large geographical areas and computationally intractable state-action spaces. Due to the intractability of these problems, traditional analytical methods fall short. Therefore, transit agencies have turned to computational approaches that enable large-scale data-driven optimization. AI-driven transportation can address this problem by learning complex abstractions of vast volumes of data to make decisions in a

Authors' addresses: Michael Wilbur, Vanderbilt University, USA; Amutheezan Sivagnanam, Pennsylvania State University, USA; Afiya Ayman, Pennsylvania State University, USA; Samitha Samaranayeke, Cornell University, USA; Abhishek Dubey, Vanderbilt University, USA; Aron Laszka, Pennsylvania State University, USA.

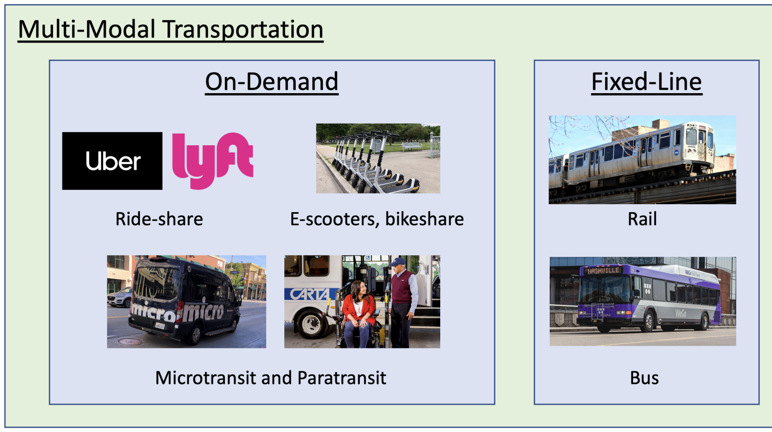


Fig. 1. Transportation modalities: there is a variety of mobility options in today's urban environments, provided by both public transit agencies and private mobility providers, which can be classified as *on-demand* or *fixed-line*. On-demand transportation provides direct service and is often accessed through a smart-phone application. Fixed-line transportation utilizes mass transit vehicles that service fixed routes and schedules. Multi-modal transportation is a broad categorization that includes any service that combines multiple modes of transit to serve passengers.

computationally efficient manner at scale and in real-time. However, it requires solving complex challenges in model representation, decision-making under uncertainty, as well as data ingestion and processing.

Since smart transportation is a broad field, we primarily focus on two high-impact modalities: fixed-line and on-demand transportation systems. Fixed-line transportation utilizes vehicles (bus or rail) that traverse pre-defined routes over the course of a day following fixed schedules. On-demand transportation is a flexible service in which users request rides either in real-time or ahead-of-time, and transportation is provided from origin to destination through ride-share, e-scooters, or bike-share. Additionally, many transit agencies have recently investigated the potential of multi-modal transportation, which provides a more flexible and adaptive transportation model that combines fixed-line and on-demand transportation.

This chapter discusses the primary requirements, objectives, and challenges related to the design of AI-driven smart transportation systems. The major content involves the following:

(1) Data sources and data management for AI-driven transportation.
(2) An overview of how AI can aid decision-making with a focus on transportation.
(3) Computational problems in the transportation domain and AI approaches.

2 CONTEXT

Large-scale adoption of smart phones and sensing technologies have revolutionized urban mobility in recent years. Led by companies such as Uber, Lyft and Via, new on-demand transportation options such as ride-share, e-scooters, and bike-share have been introduced to complement existing transportation services provided by public transit agencies. As the mobility landscape evolves, public transit agencies are tasked with managing ever more complex systems. To understand the relationship between transportation modalities, we can classify these systems as *on-demand*, *fixed-line*, and *multi-modal* as shown in Figure 1.

On-demand transportation provides direct service from origin to destination. These systems are often accessed through a smart-phone application or by calling an agency to schedule trips. Ride-share companies such as Uber and Lyft provide their own smart-phone applications for users to request rides in real-time. Microtransit and paratransit provide similar services but are distinguished by the use of high-capacity vehicles. In these systems, passengers typically share rides with several other users, and trip requests can be placed ahead-of-time (for example, a trip can be requested for later in the week at a specific time) or in real-time in the same way as ride-share. These services can be provided by private mobility providers or be managed by a public transit agency. From a city's perspective, microtransit services are available to all residents and can be thought of as a low-cost extension of their public transit system. They can be used for direct point-to-point travel as well as in hybrid transit systems, where the vehicles shuttle passengers to and from fixed-line transit [28]. Similarly, paratransit provides curb-to-curb service for passengers who are unable to use fixed-route transit (e.g. passengers with disabilities) and is often considered a social good or required by law.

Fixed-line transportation is the backbone of urban mobility across many urban centers. These services are provided by public transit agencies and consist of large, mass-transit vehicles such as buses or rail-cars that service fixed routes and schedules. A fixed route is a pre-defined sequence of locations that a vehicle will traverse with a scheduled arrival time at each location. Transit agencies aim to provide service at each stop at relatively frequent intervals throughout the day. Fixed-line services have high capacity and can transport passengers in a way that is more environmentally friendly compared to on-demand services or personal vehicles. However, in many cities fixed-line transportation struggles to attract a critical mass of passengers while providing adequate coverage to all areas of the city [37]. Therefore, optimizing these systems is important for increasing ridership and reducing costs in a way that properly serves the needs of passengers spread over wide geographical areas.

As a response to these challenges, public transit agencies have begun revamping their operations in a way that integrates on-demand services with fixed-line transit to provide dynamic coverage in areas where demand is spread out spatio-temporally, making it inefficient to serve these areas with high-capacity, fixed-line services. We can refer to this as *multi-modal* transportation. There are many open challenges to implementing multi-modal transportation systems that are outside the scope of this work. Instead, we focus on optimization methods for on-demand and fixed-line services. *Optimization in this context aims to improve ridership and efficiency while providing adequate service to residents across socio-economic groups.* By optimizing these services, we can better understand how these systems can be integrated more effectively. The most promising way to optimize transportation services is through data collection and analysis. Data enables transit agencies to improve their understanding of their systems to better inform decision-making as well as leverage recent advances in AI to optimize their services, which is the focus of this work.

3 BACKGROUND

Effective solutions are of critical importance in that poorly designed mobility applications can make for unhappy users, increase costs, and in the case of safety critical applications, can affect the livelihood of residents that depend on the system. The goal of these systems is to learn models from data that can aid optimization and support more efficient mobility options. AI-driven smart transportation systems therefore require integrating *data*, *learning*, and *optimization* as shown in Figure 2. Each one of these components presents unique challenges that must be addressed when designing smart transportation systems. In this section, we summarize each of these components and how they can be combined to create smart transportation applications.

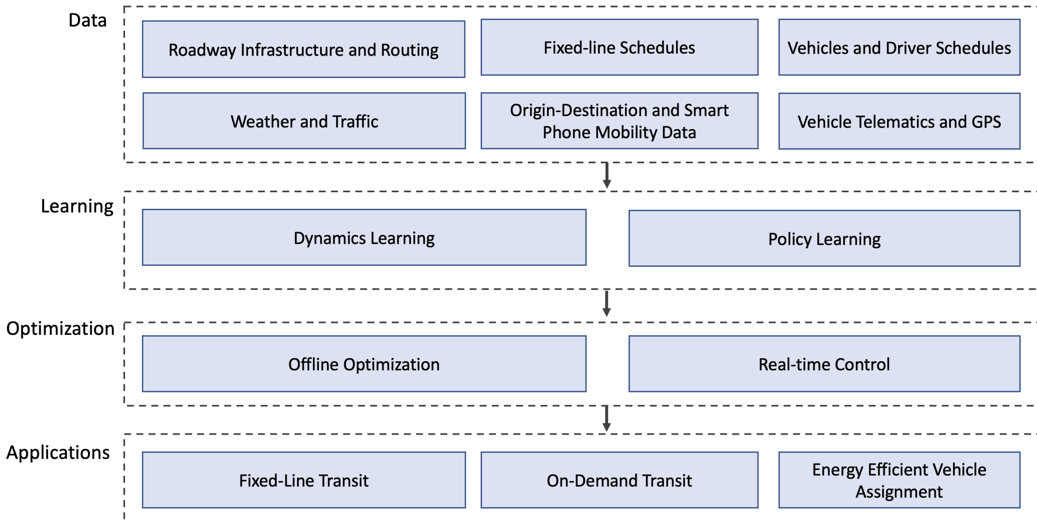


Fig. 2. AI-driven smart transportation requires integrating data, learning, and optimization to provide smart transportation services and applications for transit agencies and users. These applications aim to optimize *utilization* (i.e. increase ridership) and *efficiency* (i.e. reduce operational costs and emissions).

3.1 Data Sources and Sensing

The "data" component refers to the need for high-quality data sources that can be used to generate datasets for training predictive models and to generate data online for real-time applications. In the transportation domain, this data is mostly streamed from various sensors and is spatio-temporal in nature. Therefore, sensor data typically includes a timestamp indicating when the reading was taken, the value of the reading, as well as a spatial component representing where this sensor is located. Sensors can be statically located along roadways or can be dynamic in that they travel over time, e.g. GPS receiver within a vehicle or bus. The spatio-temporal nature of these data sources presents challenges in efficient storage, synthesis, and data retrieval [3, 36, 40]. Synthesis requires processing data in a variety of domain-specific formats and at irregular intervals. Raw sensor streams must often be enriched by joining them with infrastructure data, such as the identifier of the road along which a vehicle is travelling.

Data sources can be classified as static or real-time. Static data are datasets related to the road networks, GIS layout, and schedules for the operating regions, while real-time data consists of data streaming from sensors and collected during real-time operations. In this sense, the static data is not fully "static." For instance, both roadway information and fixed-line schedules can change over time. However, the static data in this context is not required to be updated daily in the way that real-time data is. The most common forms of static data used for building smart transportation systems include the following.

- *Roadway Infrastructure and Routing Graphs:* Transportation models require a way to represent the roadway network. A commonly used roadway network data source is OpenStreetMap (OSM). OSM is community-driven and open-source with updates provided by the OSM community with a wide geographic footprint [14]. Data can be directly downloaded from various mirrors [35] and is used as the mapping data for a variety of commercial-grade applications. OSM in its raw form consists of *nodes*, which are points with a geographic

position (latitude, longitude); *ways*, which are ordered lists of nodes representing linear features such as streets; and *relations*, which define relationships between nodes and ways. Together, these features represent a graph that can be used for mapping or routing. There are numerous open-source projects that provide routing services such as shortest path calculation and travel time estimation using OSM data, including the Open Source Routing Machine (OSRM), OpenTripPlanner, and Valhalla [20, 25].
- *Transit Schedules:* The most widely used schema for representing fixed-line routes and schedules is the General Transit Feed Specification (GTFS) [13]. GTFS describes a standard format for public transportation schedules and associated geographic information. This includes standard representations for routes, trips, stops, and scheduled arrival times at each stop for buses and subway networks. It also includes geo-spatial shape objects representing routes so that fixed-line transit can be mapped to the operating region. Public transit agencies maintain and publish their own schedules in GTFS format for public consumption. In this way, developers and researchers can write code that consumes GTFS data in an interoperable way that can work across transit agencies and regions, provided that the transit agencies maintain GTFS-compliant data sources.
- *On-Demand Driver Runs:* Some forms of on-demand transportation, particularly those managed directly by public transit agencies, may include scheduling information for vehicles and drivers. For instance, in the case of micro-transit and para-transit, a transit agency maintains a list of available drivers for a day and the start and end times of their runs. It may also maintain the list of available vehicles including the capacity of each vehicle and which driver-run corresponds to each available vehicle. This contrasts with many commercial ride-share services, such as Uber or Lyft, where there are no fixed schedules for drivers.

Advances in sensing technologies have led to an explosion of real-time streaming data in urban environments. These sensors allow us to monitor transportation systems in real-time and act as a bridge between the physical and cyber domains. They allow us to quantify and monitor the system through data. Real-time data then serves two purposes. First, we can collect sensor data over a time range to generate offline datasets. These offline datasets are joined with the static data and are used to train AI models. Second, the learned models then ingest the sensor data at inference time to make predictions and aid decision-making processes.

The data itself comes in a variety of formats and is typically defined as high-velocity and high-volume. Therefore, the problem of ingesting, processing, storing, and interpreting real-time sensor data in the transportation domain inherits the traditional problems associated with Internet of Things (IoT) analytics [23]. Commonly used real-time data sources in the transportation domain are as follows.

- *Vehicle Telematics and GPS:* Vehicle telematics typically consist of physical sensing kits installed in vehicles. These devices can record vehicle dynamics such as speed, acceleration, and tractive or brake torque of a vehicle or flow of vehicles [8]. They can also record power consumption and generation which can vary between types of vehicles (e.g. electric, diesel, and hybrid vehicles). AI-driven smart transportation relies heavily on vehicle positioning and trajectory tracking using GPS. GPS readings are streamed in real-time either through GPS-specific devices installed in the vehicle or through mobile applications on smart phones and tablets. In addition to monitoring vehicle locations, GPS trajectories can also be tracked for transit users through user-defined smart-phone applications.
- *Origin-Destination and Smart-Phone Data:* Many optimization applications for smart transportation require origin-destination (OD) datasets which encompass trips for users throughout the day. At a minimum, OD data requires an origin location (where the user originates)

and a destination location (where the user ends) for each trip. Optionally these locations can be timestamped in a way that represents when the user started and or ended the trip. Historical data may also include trajectories which represent geo-points along the trip for the transit user. One publically available source of OD-data is the Longitudinal Employer-Household Dynamics (LEHD) Origin-Destination Employment Statistics (LODES), which is provided by the United States Census Bureau [7]. LODES data includes information related to census groups and tracts where residents live and work, which can be used to investigate demographics and trips between spatial regions within a city or state. There are also commercially available datasets derived from anonymized smart-phone GPS data, which can aid OD generation by provided traffic related to points-of-interest (POIs) [27].

- *GTFS Realtime:* GTFS is an open data format used by many transit agencies to represent public transportation schedules and associated geographic information. GTFS Realtime is an extension to the static GTFS schedules that provides up-to-date information about current arrival times, vehicle locations, occupancy, and service alerts to help users better plan their trips [12]. The structured nature of GTFS Realtime makes it easy to monitor fixed-line public transit performance in the context of the transit schedules and routes. GTFS Realtime has been incorporated into a variety of commercial-grade applications including Google Maps. GTFS Realtime is typically provided by transit agencies themselves. Compared to raw GPS data, GTFS Realtime is a structured format that is interoperable between agencies and telematics kit providers.
- *Weather:* Weather conditions can have a significant impact on transportation systems including on energy usage and travel delays [39]. Raw weather data can be accessed for weather stations throughout the United States from the National Oceanic and Atmospheric Administration (NOAA). Common weather features include temperature, precipitation, atmospheric pressure, wind speed, and wind direction. Various APIs, such as Meteostat [21], provide easy access to historical as well as real-time NOAA weather data for simple integration into AI pipelines.
- *Traffic:* Traffic congestion and road segment speeds are important features for estimating travel time, routing, and energy prediction in smart transportation applications. Typically these datasets include real-time travel speeds or travel time along roadway segments that must be tied in some way to the roadway infrastructure. HERE Technologies provides an API for speed recordings for segments of major roads in the form of timestamped speed recordings [34]. Every road segment is identified by a unique Traffic Message Channel identifier (TMC ID). Each TMC ID is also associated with a list of latitude and longitude coordinates, which describe the geometry of the road segment. Additionally, INRIX [16] provides TMC-level traffic information that also is mapped to OSM segments and is therefore easily integrated with OSM-based applications.

3.2 Optimization for Transportation

Smart transportation systems require solving computational problems that optimize various objectives, such as ridership, costs, or emissions over time. Optimization problems in this setting can be broadly classified in two categories: offline and real-time. In this section, we will discuss these two settings and the role that learning plays in solving these computationally challenging problems.

3.2.1 Offline optimization problems. Offline optimization problems are often solved ahead of time in a batch setting, and they are common in logistics and supply-chain optimization. For example, a package delivery company may need to assign routes to drivers to deliver items the next day. This class of problems is fundamentally combinatorial optimization. Broadly, combinatorial optimization involves finding an optimal solution from a set of feasible solutions, where the set of feasible solutions is discrete or can be reduced to a discrete set; many of these problems are NP-hard. In transportation settings, there are often strict constraints that must be satisfied. For example, in the case of package delivery, there may be certain times of day when the package must arrive at a specific location or a fixed number of vehicles that are available. Therefore, a feasible solution is a solution that satisfies all of the constraints, and an optimal solution is a feasible solution that maximizes the objective. Combinatorial optimization can be modelled as an integer linear program or mixed-integer linear program (MILP) and solved using state-of-the-art heuristic search approaches, such as branch-and-bound or branch-and-cut. While these methods are exact in that they are guaranteed to converge to optimal solutions, they unfortunately reduce to potentially enumerating all feasible solutions, which poses scalability issues.

3.2.2 Real-time control (online optimization problems). Conversely, real-time control (or online planning) refers to optimization problems where decisions must be made sequentially over time. Sequential decision-making is commonly formalized as a Markov Decision Process (MDP) [26]. An MDP can be specified by a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, p(s_o)\}$:

- Set of states $\mathcal{S}$ and distribution over the starting state $p(s_0)$.
- Set of actions $\mathcal{A}$.
- Transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ takes as input the current state $s_t \in \mathcal{S}$ and an action $a_t \in \mathcal{A}$ at timestep t and returns the next state $s_{t+1} \in \mathcal{S}$.
- Reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$.
- The policy $\pi : \mathcal{S} \to \mathcal{A}$ controls the agent's behaviour by selecting an action given the current state.

The *environment* consists of the transition function and reward function. At each timestep t the environment takes s_t, a_t and returns the next state from $\mathcal{T}(s_{t+1}|s_t, a_t)$ and a scalar reward from $\mathcal{R}(s_t, a_t)$. The objective of sequential decision-making is to select actions given a policy $\pi(a_t|s_t)$ to optimize *cumulative* reward over time.

3.3 Learning

The fundamental advantage of learning-based AI methods is their ability to handle high-dimensional state-space and to *learn* decision procedures/control algorithms from data rather than from models. This is advantageous in the transportation domain, where real-world state-action spaces are high-dimensional and manifest as intractable state-spaces for mathematical modeling and analysis. Therefore, we can employ state-of-the-art AI methods to learn abstract representations of these problems to tackle real-world transportation problems. In this section, we discuss two broad categories of learned models: *descriptive* models and *generative* models.

3.3.1 Descriptive Models. Descriptive models utilize supervised learning to learn abstract representations of the model environment and are often referred to in the AI literature as discriminative models. The models are trained with labeled data and aim to capture the conditional probability $p(Y|X)$ for input features X and output labels Y. That is, predictive models aim to learn a function $f : X \to Y$ that maps the input features to output labels. Supervised learning can be separated into two types of problems (or *tasks*): classification and regression. Classification tasks aim learn a function f that maps the input features to labels that are discrete categories. For example, given the features of a transit route and the current state-of-charge for an electric bus, we may want to predict whether the vehicle can service the route without having to re-charge. This would be a binary classification problem. On the other hand, regression tasks map the input features to a continuous label space. For example, given the features of a transit route and the current state-of-charge for an

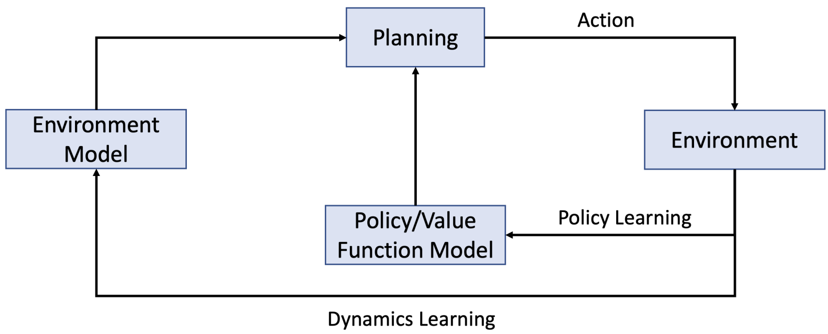


Fig. 3. Real-time planning for transportation can be modelled as a sequential decision-making problem. Planning requires making decisions over intractably large state-action spaces in stochastic settings. Learning can aid planning by either 1) learning the dynamics of the environment or 2) directly learning or use information from a policy/value network to improve the planning procedure.

electric bus, we may simply want to predict the energy that will be used for this vehicle to service the route. There are many state-of-the-art methods for learning predictive models, including deep neural networks (DNN).

3.3.2 Generative Models. On the other hand, generative models are a class of statistical models that learn the distribution of the environment $P(X)$, or in tasks where labelled data is available, learn the joint probability of $P(X, Y)$. Generative models provide a condensed representation of the environment for building simulators, learning policies through reinforcement learning, or being queried at inference time to simulate future scenarios. Generative models in the transportation domain are particularly important for modelling demand. For example, we might create a generative demand model for representing the spatio-temporal distribution of trip requests for on-demand transportation, which can aid decision-making when assigning vehicles to service new requests in real-time.

3.4 AI-Driven Optimization

There are many ways in which we can utilize descriptive and generative models to aid optimization in the transportation domain. For example, take the high-level description of a sequential decision problem, as shown in Figure 3, in the context of on-demand ride-share. In on-demand ride-share, at each timestep a new batch of trip requests arrives and must be assigned to vehicles that can service these new requests. The *planning* component refers to algorithms or methods responsible for this assignment at each timestep. In this case, the *environment* refers to the transportation system itself and includes roadways, traffic congestion, weather, trip requests, etc., all of which generate data. The planner and environment interact sequentially, and as discussed in Section 3.2, the goal is to select actions that maximize the objective (costs, emissions, and utilization) over time.

Transportation systems are computationally complex, which makes traditional analytical methods intractable in this setting. Therefore, we can utilize AI and machine learning to learn condensed representations through data to aid planning. In Figure 3, we highlight two ways to achieve this.

3.4.1 Dynamics Learning. Dynamics learning refers to directly learning representations related to the environment itself. Recall from Section 3.2.2 that the environment consists of the transition function $\mathcal{T}$ and reward function $\mathbf{R}$. Using these two functions, the environment takes as input the current state s_t and action a_t and returns the next state s_{t+1} and reward r_t. The goal of dynamics learning is to learn the transition function $\mathbb{P}[s_{t+1}|s_t, a_t]$ and/or the reward function $\mathbb{P}[r_t|s_t, a_t]$ through experience. For example, in the context of on-demand ride-share, we could learn a generative model of demand or a traffic congestion model. These environment models can be used to investigate how certain decisions will play out over time and contextualize decisions in light of future scenarios through *planning*. Planning in this context can be a greedy algorithm that simply selects the action with the greatest predicted reward using the reward function or use a combination of the transition and reward functions to look-ahead through online search. A common online search method in this setting is Monte Carlo tree search (MCTS), which we discuss in later sections. This combination of model-learning and planning is often referred to as model-based RL in the RL literature [22].

3.4.2 Policy Learning. Alternatively, recent advances in model-free reinforcement learning aim to directly learn policies or value functions that can be embedded directly in the planning component. This can be done with or without a model of the environment. For example, recall that the policy π is responsible for selecting an action given the current state with the goal of maximizing reward over time. Therefore a policy $\mathbb{P}[a_t|s_t]$ can be parameterized and learned directly through experience or simulation. Alternatively, we can learn a value function that predicts the future reward for a state given a policy $v_\pi(s) = \mathbb{E}_\pi[r_{t+1} + r_{t+2} + \dots |S_t = s]$. Model-free RL has proven effective in numerous application domains including learning to play video games or other camera-input applications where learning a model of the environment is intractable [2]. However, as we will discuss, in the transportation domain a model of the environment can often be learned. In the following sections, we discuss concrete examples of how to combine environment models, policies, and planning to solve unique problems in the transportation domain.

4 COMPUTATIONAL PROBLEMS AND AI APPROACHES

In this section, we discuss some key computational problems in which AI has been used in recent work related to transportation.

4.1 On-demand Paratransit and Microtransit

Advances in sensing and smart phone adoption have led to rapid expansion of new demand-responsive modes of transportation in recent years such as ride-share (Uber, Lyft), e-scooters, and bike-share. In the case of ride-share, users request trips through a smart-phone application and a vehicle provides direct transportation to the user's requested destination. As users have increasingly utilized demand-responsive services in this way, public transportation agencies are looking for ways to make their own transportation options more demand-responsive and dynamic. One way in which transit agencies can make their systems more responsive is through microtransit. Microtransit offers door-to-door service with high-capacity vehicles. Similarly, paratransit service is a socially beneficial curb-to-curb transportation service provided by public transit agencies for passengers who are unable to use fixed-route transit (e.g. passengers with disabilities). Paratransit is often required by law and comes with its own unique set of design challenges.

Both paratransit and microtransit are forms of ride-pooling, which can be defined as a direct transportation service that allows trips to be pooled together in a way that users can share rides with other passengers. By allowing passengers to share trips, ride-pooling has the potential to overcome inefficiencies with traditional ride-share—particularly by reducing vehicle miles travelled (VMT) [30, 32]. However, the design and implementation of ride-pooling systems has numerous challenges. First, the problem is computationally complex and involves solving complex allocation problems in real-time. Second, the environment is highly uncertain and dynamic in that traffic, weather, and the distribution of requests can change rapidly over time.

Ride-pooling is a dynamic version of the vehicle routing problem (DVRP) with stochastic trip requests. In this setting, some customer requests may be known at the time of planning while others are unknown, and some stochastic information may be available about potential future requests. When a new trip request or batch of trip requests arrive, these requests must be assigned to vehicles that can serve these requests without violating the VRP constraints. Both microtransit and paratransit include standard constraints, such as that pickups must occur before dropoffs, and each passenger can be served by only one vehicle. Since paratransit primarily focuses on serving passengers with disabilities, paratransit services operate under the Americans with Disabilities Act (ADA), which enforces time windows as a hard constraint, unlike on-demand microtransit services, thereby requiring strict adherence to such constraints which must be taken into account.

There are many highly efficient myopic solutions available for the ride-pooling problem. State-of-the-art myopic assignment utilizes shareability graphs to generate a large number of feasible solutions and then batch assigns requests to vehicles through an integer linear program (ILP) [1]. These methods are highly efficient and can solve the assignment problem to near-optimality at each decision-epoch, but since they do not take into account future information, they may not be optimal when evaluated over the course of a full day. However, the availability of stochastic information and environment models presents an opportunity to utilize AI to learn more non-myopic solutions that can make decisions in the context of expected future demand and environment conditions. There are two primary methods for designing non-myopic solutions to the DVRP. First, we can utilize historical trip requests and demand to build a simulator that can be used to train a value function that encodes the value of a future state in the context of expected reward over the entire day [17]. This approach can be categorized as model-free reinforcement learning and is computationally efficient at run-time. However, it requires a large amount of high-quality data for the simulator and can perform poorly when real-time conditions deviate greatly from historical patterns. The second approach is to utilize dynamics learning to create a generative demand model that can be used with online search to evaluate current actions in the context of future scenarios [38]. This method is more adaptive to changing environment dynamics at run-time since traffic and demand models can be updated based on current conditions. However, this approach is less scalable since it requires computationally intensive online search to run at each decision epoch.

4.2 Prediction of Energy Usage

Transportation accounts for 28% of total energy consumption in the U.S.[10], leading to significant environmental consequences such as urban air pollution, greenhouse gas emissions. Shifting from personal vehicles to public transit systems offers a substantial opportunity to reduce energy usage and environmental impact. However, public transit itself consumes substantial energy, with U.S. bus services alone emitting around 19.7 million metric tons of CO2 annually [24]. The adoption of electric vehicles (EVs) can help decrease the environmental impact, but their higher costs result in a combination of EVs, hybrids, and internal combustion engine vehicles (ICEVs) in transit agency fleets. This creates a challenging optimization problem for transit agencies: determining which vehicles should be assigned to specific transit trips. Since the advantages of EVs over ICEVs vary depending on factors like route and time of day (e.g. EVs are more beneficial in slower traffic with frequent stops and less advantageous on highways), the assignment of vehicles can significantly affect energy use and, consequently, environmental impact. The heart of operational optimization of a transit agency lies in the challenge of precisely forecasting the electricity and fuel consumption of transit vehicles. Accurate prediction of electricity and fuel consumption for transit vehicles is difficult due to various factors like vehicle type, traffic conditions, and road characteristics. Leveraging sensor-based technologies, data analytics, and machine learning can address these challenges, allowing the development of a comprehensive framework for route-level energy prediction in public transit.

Within this subsection, we introduce an innovative framework designed for data-driven offline energy consumption prediction at the route level [4, 5]. Our framework is specifically tailored for mixed-vehicle transit fleets, considering diesel (ICEV), hybrid and electric vehicles to accommodate the operational diversity of the fleet. To evaluate its efficacy, we utilize real-world data obtained from the bus fleet of a public transit authority from a midsize city in U.S.

4.2.1 Data Processing Framework. Before applying machine-learning models for energy usage prediction, we must process the time series data recorded from the vehicles by cleaning it, generating samples with a fixed-dimension feature space, and incorporating data from other sources, including traffic and weather data. For each bus, the recorded data is a series of datapoints, numbered $i = 0, 1, 2 \ldots$, where each datapoint is a tuple of a timestamp TS_i, a location L_i, etc. For EVs, each datapoint i includes a battery current A_i, a battery voltage V_i, a battery state of charge (%), and a charging cable status (0 or 1). For diesel and hybrid vehicles, instead of battery data, datapoint contains fuel usage in gallons. First, we filter out specific data points in our analysis to achieve our objective of predicting energy usage. First, we exclude data recorded when a bus is either waiting in the garage or undergoing charging. We also remove data points where the charging cable status indicates active charging (for electric vehicles).

In order to create labels for our supervised energy prediction model, we estimate the energy usage from the recorded vehicle data. For diesel buses, we can determine the amount of fuel consumed between two consecutive data points by calculating the change in the total fuel used. For electric buses, estimating energy usage is more intricate. The state of charge values, recorded with limited precision, can be used to approximate energy usage. However, for accurate values, we rely on estimates derived from recorded battery current (A) and voltage (V) values. Instantaneous power usage (in Watts) at any given time can be computed as the product of A and V. We estimate the energy used (in Joules), denoted as E_i, between consecutive data points $(i - 1)$ and (i) using $E_i = A_i \cdot V_i \cdot (TS_i - TS_{i-1})$. Here, TS_i represents the timestamp of data point i in seconds. This formula provides highly accurate estimates since current and voltage values are recorded at least once per second. To ensure the validity of our estimates, we conducted comparisons against changes in SoC across numerous data points, confirming their unbiased nature.

One of the important steps in the data processing framework is *cleaning and mapping the GPS location to roads*. The recorded vehicle locations, based on GPS, inherently contain noise. Some locations may fall on streets or parking lots where buses cannot traverse, posing challenges for accurate distance calculations and integration with other data sources. To mitigate this noise, we combine the recorded vehicle locations (L_i) with a street-level map obtained from OpenStreetMap (OSM) [14]. OSM represents roads using disjoint segments known as OSM features, each with unique identifiers and properties. To align the vehicle locations with the street-level map, we employ two novel methods: a heuristic algorithm and a machine-learning approach. For mapping recorded locations (L_i) to OSM segments (road segments), we consider nearby OSM segments based on geographical distance. We create an R-tree spatial index for the street-level map and intersect it with a bounding disk around L_i. This intersection yields a set of potential road segments, which we filter based on road types suitable for buses. For heuristic approach, we count the number of preceding and following locations near each nearby segment and select the segment with the most nearby locations. For machine-learning approach, rather than directly outputting the correct segment from the nearby candidate segments, we estimate their likelihood using a regression model. The model incorporates variables such as distance between the location and the candidate OSM segment, road type, and distances to the set of preceding and following locations. The model outputs a value

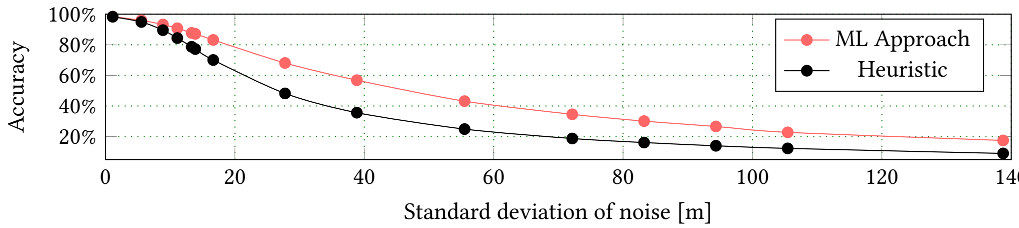


Fig. 4. Accuracy of mapping noisy locations to road segments using heuristic and machine-learning approach. Accuracy is measured as the fraction of locations that are mapped to the correct road segment.

between 0 and 1, indicating the likelihood of correct mapping. We apply the regression model to each nearby candidate segment and select the one with the highest likelihood for mapping the location.

For evaluating the accuracy of different approaches for mapping noisy locations to OSM features, we first generate routes using a street-level map, selecting locations precisely on the roads, and then introduce random noise to these locations (noise is generated from a two-dimensional Gaussian distribution with zero mean)[1]. The standard deviation of the noise is varied from 1 meter to 140 meters in both directions. Finally, we map the noisy locations to road segments using both the heuristic and machine-learning approaches with decision tree regression. Accuracy is measured as the ratio of correctly mapped locations. In Fig. 4, the accuracy of both approaches is compared across different levels of noise. At the lowest noise level (1.1 meters), both the heuristic algorithm and the machine-learning approach achieve accuracy above 98%. As expected, the accuracy of both approaches decreases as the noise level increases. However, the machine-learning approach outperforms the heuristic algorithm at higher noise levels.

Once we map the location traces to road segments, we can generate samples by dividing the time series data into segments based on the traveled road segments. Each sample represents a maximal continuous travel on a specific road segment and includes the starting and end locations, starting and end times, and the sum of energy used during the travel. To accurately calculate the distance traveled for each sample, we obtain the geometry of the corresponding road segment from OSM and identify the line segments that the bus actually traveled. The distance traveled for each sample is calculated based on the partial distance on the starting line segment, the complete distance of the intermediate line segments, and the partial distance on the ending line segment.

Finally, we incorporate additional features for the prediction model. These features encompass elevation changes within the samples, weather-related attributes like temperature, humidity, visibility, precipitation, and wind speed, as well as traffic data such as the speed ratio between actual speed and free-flow speed.

4.2.2 Prediction Models. For prediction of energy usage, we apply three different machine-learning models: artificial neural network, linear regression, and decision tree regression. The input of the energy prediction models (i.e. training or test set) is a set of samples, where each sample is a tuple of distance travelled, various road-type features, elevation change, various weather features, various traffic features, and energy used as the target feature. Before training and testing, we map categorical variables (e.g. road type) into sets of binary features using one-hot encoding. We train all three models to minimize *mean squared error* (MSE).

[1]The synthetic location traces are solely used for evaluating the mapping approaches, as we require ground-truth segments. However, for training and evaluating energy-use prediction, we utilize real location traces obtained from vehicles.

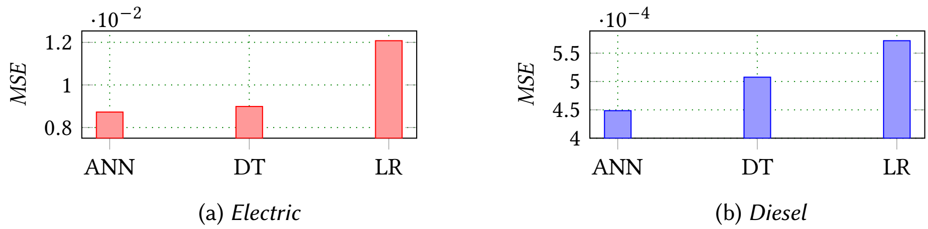


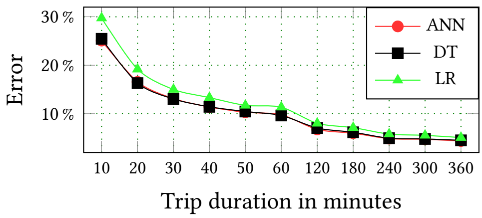
Fig. 5. Comparison of different energy prediction models based on *mean squared error* (MSE) for electric and diesel vehicles.

For our neural-network based models, we use feed-forward deep neural networks with one input layer, multiple hidden layers, and one output layer. Sigmoid activation is used in the hidden layers, and linear activation is used in the output layer for both diesel and electric. The models are optimized with the *Adam* optimizer [18] and a learning rate of 0.001, implemented using Keras [9], a high-level API of TensorFlow. We use standard multiple linear regression and decision tree regression [29] as other prediction models. For implementing the models, we use the implementation provided by the *scikit-learn* Python library. Decision tree regression builds a tree structure based on the training samples, where each node represents a decision based on the value of a feature variable, and leaf nodes provide predictions. We opted for neural networks due to their exceptional predictive capabilities, a fact supported by our numerical findings. In comparison, linear and decision tree regression techniques do not exhibit the same level of performance, although their outcomes are more straightforward to comprehend and articulate. For instance, linear regression highlights the direct relationship between input variables and target features, providing a clear understanding of their influence.

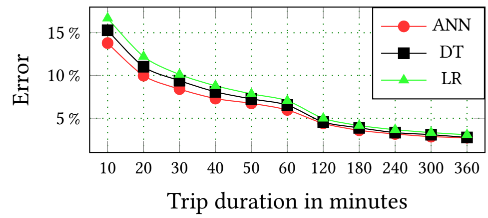
4.2.3 Numerical Evaluation.

Comparison of Prediction Models for Short trips. We assess the performance of three machine-learning models in predicting energy usage for short trip segments. These segments pose a challenge due to their limited distance and duration. Fig. 5 displays the results in terms of mean squared error (MSE) for all three models for both diesel and electric vehicles. The artificial neural network (ANN) demonstrates superior performance compared to the other models, both for electric and diesel vehicles.

Comparison of Prediction Models for Longer Trips. We also examine the performance of our models in predicting energy usage for longer trips. To accomplish this, we segment our time series into extended durations, ranging from 10 minutes to 6 hours. For each trip duration, we generate a collection of examples and utilize our models to predict energy consumption for each example. Subsequently, we compare the sum of these predictions with the actual energy usage recorded for the entire trip. Fig. 6 shows the relative prediction error for trips of various lengths. When analyzing various trip lengths, we observe that the average error values, computed across multiple trips, tend to be lower for longer trips. This is an expected outcome because with larger sample sizes, the individual errors of numerous samples cancel out each other when using an unbiased prediction model. In the case of diesel vehicles, we discover that the artificial neural network (ANN) consistently outperforms the other models across all trip lengths, exhibiting significant improvement. However, for electric vehicles, both the ANN and decision tree (DT) models perform equally well for the majority of trip lengths.



(a) Electric



(b) Diesel

Fig. 6. Energy prediction error for longer trips, consisting of many samples, with neural network (ANN), decision tree (DT), and linear regression (LR).

4.3 Energy-Efficient Vehicle Assignment

Providing affordable public transit services is crucial for communities to ensure residents carry out daily activities such as employment, education, and other services without interruption. Unfortunately, transit agencies are facing inefficiency in utilizing transit resources, which results in high operating costs and environmental impact. Further, replacing traditional internal-combustion engine vehicles (ICEVs) with electric vehicles (EVs) can reduce energy costs and environmental impact. But due to the high upfront costs of EVs, transit agencies tend to use mixed fleets of vehicles (i.e. EVs and ICEVs). To efficiently utilize the mixed fleets of vehicles, transit agencies must optimize how they assign each vehicle to serve a transit trip and ensure EVs are charged in a timely manner for uninterrupted services, which is a challenging problem for transit agencies with large transit networks [31]. Because the advantage of using EVs inplace of ICEVs differs based on the route and time of the day. For example, EVs can be good at more traffic with frequent stops compared to be used on highways. Further, transit agencies have limited charging capabilities (i.e. number of charging poles, maximum power to consume at charging station to avoid high peak loads on the electric grid) which makes the problem more challenging.

4.3.1 Problem Formulation. We begin the mathematical formulation by defining a transit network, which includes a set of locations $\mathcal{L}$ such as bus stops, garages, and charging stations. Transit agency operates a set of *buses* $\mathcal{V}$, with the assistance of the transit network to serve the day-to-day transit services. Each bus $v \in \mathcal{V}$ belongs to a *vehicle model* $M_v \in \mathcal{M}$, and $\mathcal{M}$ is the set of all vehicle models in operation. Further, we divide vehicle models $\mathcal{M}$ into two disjoint subsets based on the energy source as liquid-fuel models $\mathcal{M}^{\text{gas}}$ (e.g. diesel, hybrid) and electric models $\mathcal{M}^{\text{elec}}$. Electric model ($m \in \mathcal{M}^{\text{elec}}$) has limited battery capacity (C_m) and may require charging during operational hours. On the other hand, liquid-fuel models can operate without refueling throughout the day. Every day the transit agencies must serve a set of *transit trips* $\mathcal{T}$ using its buses. Each trip $t \in \mathcal{T}$ starts from the origin $t^{\text{origin}} \in \mathcal{L}$ at t^{start}, then passes through a set of stops at fixed arrival and departure times, and finally reaches the destination $t^{\text{destination}} \in \mathcal{L}$ at t^{end}.

The transit agency installs a set of *charging poles* $\mathcal{CP}$ (generally at the depot) to charge electric buses. To better facilitate the operation of each charging pole $cp \in \mathcal{CP}$, the day can be divided into uniform-length disjoint *time slots* $\mathcal{S}$, where each slot $s \in \mathcal{S}$ begins at s^{start} and ends at s^{end}. We combine the set of charging poles and slots and obtain the charging slots $\mathcal{CS} = \mathcal{CP} \times \mathcal{S}$. Each electric bus ($v \in \mathcal{V} \wedge M_v \in \mathcal{M}^{\text{elec}}$) can be charged at any unassigned charging slot ($cs \in \mathcal{CS}$) at a time, as long the electric bus v can reach the charging pole without running out of charge in the battery.

In between two transit trips $x_1, x_2 \in \mathcal{T} \bigcup \mathcal{CS}$ (or between a transit trip and a charging slot, or between two charging slots for electric buses), the bus may be required to move from destination for the previous trip x_1 to the origin of the following trip x_2. Such trips can be defined as *non-service trips* since they are not directly contributing to serving passengers but are required for the operation of the buses throughout the day. Each on-service trip can be defined by $T(l_1, l_2)$ where the trips start at the location $l_1 \in \mathcal{L}$ and end at the location $l_2 \in \mathcal{L}$.

Solution. We define the solution as a set of trip assignments where each transit trip is assigned to exactly one bus $v \in \mathcal{V}$ and a set of charging assignments of the electric bus to the combination of a charging pole $cp \in \mathcal{CP}$ and specific slot $s \in \mathcal{S}$ on the charging pole.

Constraints. The solution must respect the following real-world constraints: Every bus can serve only one trip at a time; A bus can serve two trips consecutively (or an electric bus can serve a trip and assign to a charging), if the bus has enough time to move from the destination of the first trip to origin of the second trip before the start time of the second trip; Every electric bus needs to have enough energy before serving every assigned trip; Only one electric bus can be charged in particular charging pole at a particular slot; the energy charged by electric bus is capped at the maximum capacity of the battery.

Objective. In the energy-efficient vehicle assignment, we aim to reduce the energy use of transit vehicles. We express our objective as

$$\min_{\mathcal{A}} \sum_{v \in \mathcal{V}:\, M_v \in \mathcal{M}^{\text{gas}}} K^{\text{gas}} \cdot e(\mathcal{A}, v) + \sum_{v \in \mathcal{V}:\, M_v \in \mathcal{M}^{\text{elec}}} K^{\text{elec}} \cdot e(\mathcal{A}, v)$$

where $e(\mathcal{A}, v)$ be the amount of energy used by bus v for all trips completed at the end of the day and K^{gas} and K^{elec} denote the unit costs of energy used for liquid-fuel and electric vehicles

4.3.2 Computational Approaches. One way to solve the above problem is using an integer program (IP). Although the IP provides the optimal solution for smaller problem instances, the problem becomes computationally intractable for larger problem instances. In this sub-section, we introduce a greedy (to provide an initial solution) and simulated annealing-based metaheuristics approach that solves the larger problem instances in polynomial time.

Greedy Algorithm. In this algorithm, we follow an iterative process where we greedily choose an assignment between all possible assignments based on assigning a bus v to a transit trip or charging slot x instead of the base energy cost to serve the request. Biased cost comprises three main costs (i) base energy cost (base energy cost for charging slot is 0), (ii) the cost associated with moving the bus from the destination of the previous trip to the origin of the current trip, or vice versa, (iii) the wait time between the end of the previous trip to the beginning of the current trip or vice versa. The algorithm at the beginning computes the biased cost for the pairs of vehicles and trips. Then follows the iterative process, choosing the assignment with the lowest biased cost in each iteration. Then the algorithm updates the costs of remaining unassigned trips with respect to the vehicle used in the previous assignment. The algorithm follows the iterative process until all the requests are assigned or fail to find a solution.

Simulated Annealing Algorithm. The algorithm starts with the solution from the greedy algorithm and performs an iterative random search where the algorithm generates a random neighbor to the current solution in each iteration. Generating the random neighbor follows an iterative process where the algorithm randomly chooses the two vehicles in each iteration, splits their operational hours into two halves, and tries to swap the assignment between two vehicles. The algorithm to generate a random neighbor terminates after enough swapping iterations are performed. Since simulated annealing is an anytime algorithm, it can be configured to run until a fixed time.

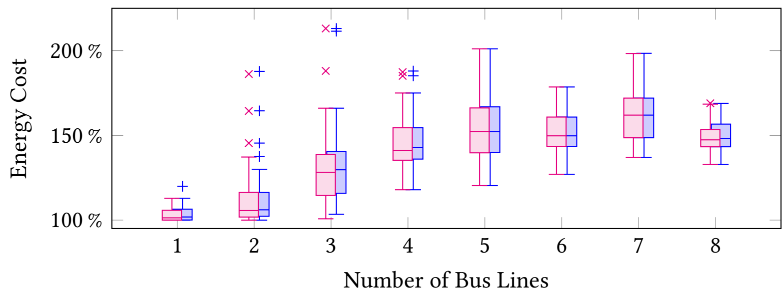


Fig. 7. Energy cost for operating transit services based on the assignment obtained from simulated annealing (■) and the greedy algorithm (■) compared to optimal assignments using IP.

4.3.3 Numerical Results. This section discusses the numerical results using the computational approach presented in the earlier sections. First, we obtain the GTFS, vehicles, and charging pole data from a real-transit agency (i.e. Chattanooga Area Regional Transportation Authority (CARTA)). To evaluate the algorithm's effectiveness presented earlier, we obtain real-world GTFS data from CARTA. Since routes for the non-service trips are not specified as part of GTFS, we obtain the routes using Google Directions API. Based on the GTFS data and routes obtained from Google Directions API, we compute the energy consumption for serving the transit and non-service trips using EVs and ICEVs based on the energy estimation described in the earlier Section 4.2. After that, we feed the energy optimization into our computational approaches and run experiments.

Solution quality. Based on the experiments on how well our heuristic and meta-heuristic minimize the energy cost compared to the Integer program, we observe that our solution approaches can restrict the solution quality to be within the range of 1.5 to 1.6, even for larger instances. For reference, you can look at the Fig. 7, which provides the energy cost of the solution obtained by our heuristic approaches compared to IP in percentages when serving 1-12 bus lines with exactly 10 trips using 3 EVs and ICEVs 5 times as the number of bus-lines capped at 50 ICEVs (i.e. for the scenario with 11 and 12 bus lines we only consider 50 ICEVs)

5 FUTURE CONSIDERATIONS

Efficient transportation systems require making decisions in real-time in complex environments. Historically, public transit systems rely on human intuition and analysis to design fixed schedules ahead of time. New demand-responsive modalities such as ride-share, microtransit, and paratransit use control algorithms to match trip requests to vehicles in real-time to provide on-demand service. Next generation systems increasingly aim to leverage AI to automate and increase efficiency across modalities using data. In this chapter, we discussed methodologies for leveraging AI in transportation and provided some concrete examples of computational problems in this space.

A major consideration for transit agencies is to increase utilization of fixed-line public transit. Public transit is highly efficient in terms of energy and emissions per passenger particularly compared to private cars and ride-share. Therefore, to combat climate change it is important to continue investigating ways to increase ridership on fixed-line services. One important future consideration is the potential of multi-modal, hybrid transportation systems that combine on-demand systems with fixed-line service in a way that uses on-demand mobility to serve the first-mile-last-mile (FMLM) to and from fixed-line. These systems have the potential to combine the best parts of demand-responsive and fixed-line services in a way that increases fixed-line utilization by expanding access to more residents.

Autonomy in general has great potential to improve mobility options in urban environments. We already discussed autonomy at the system level, e.g. in the context of designing real-time ride-share and ride-pooling algorithms or optimizing trip-vehicle assignment for fixed-line. Moving forward, there is great potential for improving control systems for autonomous vehicles. At a system level, autonomous vehicles have the potential to reduce accidents as well as traffic, thereby increasing safety and efficiency [19, 33]. As these systems become more capable it will be important for transit agencies to adapt to solve increasingly complicated control and optimization problems that arise from operating mixed fleets that include conventional as well electric and autonomous vehicles.

ACKNOWLEDGEMENTS

The authors will like to acknowledge the collaborative efforts of Dr. Ayan Mukhopadhyay, Vanderbilt University and Mr. Philip Pugliese from Chattanooga Area Regional Transportation Authority. The work presented here has been sponsored in part by the National Science Foundation through 1952011 and Department of Energy through awards DE-EE0008467 and DE-EE0009212. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and Department of Energy.

REFERENCES

[1] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[2] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[3] Berkay Aydin, Vijay Akkineni, and Rafal A Angryk. Modeling and indexing spatiotemporal trajectory data in non-relational databases. In *Managing Big Data in Cloud Computing Environments*, pages 133–162. IGI Global, 2016.

[4] Afiya Ayman, Amutheezan Sivagnanam, Michael Wilbur, Philip Pugliese, Abhishek Dubey, and Aron Laszka. Data-driven prediction and optimization of energy use for transit fleets of electric and ice vehicles. *ACM Transactions on Internet Technology*, 22(1):7:1–7:29, February 2022.

[5] Afiya Ayman, Michael Wilbur, Amutheezan Sivagnanam, Philip Pugliese, Abhishek Dubey, and Aron Laszka. Data-driven prediction of route-level energy use for mixed-vehicle transit fleets. In *6th IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 41–48, September 2020.

[6] Eda Beyazit. Evaluating social justice in transport: lessons to be learned from the capability approach. *Transport Reviews*, 31(1):117–134, 2011.

[7] The United States Census Bureau. Longitudinal employer-household dynamics, 2023.

[8] Yuche Chen, Guoyuan Wu, Ruixiao Sun, Abhishek Dubey, Aron Laszka, and Philip Pugliese. A review and outlook on energy consumption estimation models for electric vehicles. *SAE International Journal of Sustainable Transportation, Energy, Environment, & Policy*, 2(1):79–96, March 2021.

[9] François Chollet et al. Keras. https://keras.io, 2015.

[10] EIA. U.S. Energy Information Administration: Use of energy explained – energy use for transportation (2018). https://www.eia.gov/energyexplained/use-of-energy/transportation.php, Accessed: May 31st, 2020, 2018.

[11] Federal Highway Administration. Status of the nation's highways, bridges, and transit: 2002 conditions and performance report, 2003.

[12] Google. General transit feed specification (GTFS) real-time overview, 2022.

[13] Google. General transit feed specification (GTFS) static overview, 2022.

[14] Mordechai Haklay and Patrick Weber. OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[15] David Harvey. *Social justice and the city*, volume 1. University of Georgia press, 2010.

[16] INRIX. Inrix traffic, 2022.

[17] Waldy Joe and Hoong Chuin Lau. Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In *Conference on Automated Planning and Scheduling (ICAPS)*, volume 30, pages 394–402, 2020.

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

[19] Nathan Lichtlé, Eugene Vinitsky, Matthew Nice, Benjamin Seibold, Dan Work, and Alexandre M Bayen. Deploying traffic smoothing cruise controllers learned from trajectory data. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2884–2890. IEEE, 2022.

[20] MapZen. Valhalla, 2023.

[21] Meteostat. Historical weather and climate data. https://meteostat.net/en Raw data: NOAA, Deutscher Wetterdienst, 2020.

[22] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 2023.

[23] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 20(4):2923–2960, 2018.

[24] Office of Transportation and Air Quality. Fast facts: U.S. transportation sector greenhouse gas emissions 1990–2017. Technical Report EPA-420-F-19-047, U.S. Environmental Protection Agency, June 2019.

[25] Project-OSRM. Open source routing machine, 2022.

[26] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[27] SafeGraph. Safegraph data documentation, 2023.

[28] Mauro Salazar, Federico Rossi, Maximilian Schiffer, Christopher H Onder, and Marco Pavone. On the interaction between autonomous mobility-on-demand and public transportation systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2262–2269. IEEE, 2018.

[29] Scikit-learn Developers. Sklearn.tree.decisiontreeregressor. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html, Accessed: May 31st, 2020.

[30] Thibault Séjourné, Samitha Samaranayake, and Siddhartha Banerjee. The price of fragmentation in mobility-on-demand services. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(2):1–26, 2018.

[31] Amutheezan Sivagnanam, Afiya Ayman, Michael Wilbur, Philip Pugliese, Abhishek Dubey, and Aron Laszka. Minimizing energy use of mixed-fleet public transit for fixed-route service. In *35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 14930–14938, February 2021.

[32] Amutheezan Sivagnanam, Salah Uddin Kadir, Ayan Mukhopadhyay, Philip Pugliese, Abhishek Dubey, Samitha Samaranayake, and Aron Laszka. Offline vehicle routing problem with online bookings: A novel problem formulation with applications to paratransit. In *31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3933–3939, July 2022.

[33] Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, Hannah Pohlmann, Fangyu Wu, Benedetto Piccoli, et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221, 2018.

[34] HERE Technologies. Here traffic, 2022.

[35] Jochen Topf and Frederik Ramm. Geofabrik, 2020.

[36] Shaohua Wang, Yang Zhong, and Erqi Wang. An integrated gis platform architecture for spatiotemporal big data. *Future Generation Computer Systems*, 94:160–172, 2019.

[37] Michael Wilbur, Afiya Ayman, Amutheezan Sivagnanam, Anna Ouyang, Vincent Poon, Riyan Kabir, Abhiram Vadali, Philip Pugliese, Daniel Freudberg, Aron Laszka, et al. Impact of covid-19 on public transit accessibility and ridership. *Transportation Research Record*, page 03611981231160531, 2020.

[38] Michael Wilbur, Salah Uddin Kadir, Youngseo Kim, Geoffrey Pettet, Ayan Mukhopadhyay, Philip Pugliese, Samitha Samaranayake, Aron Laszka, and Abhishek Dubey. An online approach to solve the dynamic vehicle routing problem with stochastic trip requests for paratransit services. In *13th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 147–158, May 2022.

[39] Michael Wilbur, Ayan Mukhopadhyay, Sayyed Vazirizade, Philip Pugliese, Aron Laszka, and Abhishek Dubey. Energy and emission prediction for mixed-vehicle transit fleets using multi-task and inductive transfer learning. In *2021 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 502–517, September 2021.

[40] Michael Wilbur, Philip Pugliese, Aron Laszka, and Abhishek Dubey. Efficient data management for intelligent urban mobility systems. In *Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems (DI-CPS'21)*, pages 22–26, May 2021.