ROADWAY SAFETY INSTITUTE

Human-centered solutions to advanced roadway safety

Improving Railroad Grade Crossing Safety: Accurate Prediction of Train Arrival Times for Emergency Response Management and Driver Decision Support

Daniel B. Work
William Barbour

Civil and Environmental Engineering
Vanderbilt University

Ren Wang

Civil and Environmental Engineering
University of Illinois Urbana-Champaign

Final Report

CTS 19-03

University of Minnesota
Driven to Discover℠

The University of Akron

Illinois
University of Illinois at Urbana-Champaign

Southern Illinois University Edwardsville

Western Michigan University

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
CTS 19-03		

4. Title and Subtitle	5. Report Date
Improving Railroad Grade Crossing Safety: Accurate Prediction of Train Arrival Times for Emergency Response Management and Driver Decision Support	February 2019
	6.

7. Author(s)	8. Performing Organization Report No.
Daniel B. Work, William Barbour, Ren Wang	

9. Performing Organization Name and Address	10. Project/Task/Work Unit No.
Civil and Environmental Engineering Institute for Software Integrated Systems Vanderbilt University 1025 16th Ave S; Suite 102 Nashville, TN 37212	CTS#2015057
	11. Contract (C) or Grant (G) No. DTRT13-G-UTC35

12. Sponsoring Organization Name and Address	13. Type of Report and Period Covered
Roadway Safety Institute Center for Transportation Studies University of Minnesota 200 Transportation and Safety Building 511 Washington Ave. SE Minneapolis, MN 55455	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

http://www.roadwaysafety.umn.edu/publications/

16. Abstract (Limit: 250 words)

Incidents at highway-rail grade crossings, locations where railroad tracks intersect surface streets at grade, are a primary driver of safety in rail transportation in the United States. Addressing safety at these locations through technology is a focus of the Federal Railroad Administration and United States Department of Transportation. Effective management of emergency response resources on the road network requires knowledge of when trains will arrive at grade crossings and temporarily disconnect emergency vehicles from parts of the community they serve. Generating estimated times of arrival (ETAs) for trains at grade crossings on a long time horizon can be used to proactively address surface transportation safety and emergency response management. This project investigates train delays to accurately estimate train arrival times at grade crossings to support in-vehicle driver alerts. The prediction of arrival times uses train-positioning information, properties of the train, properties of the network, and properties of potentially conflicting traffic on the network as input. A historical algorithm is developed to accurately model delays using train-positioning information and an online algorithm integrates real-time train position information into the forecasts. Amtrak data and over two years of CSX freight rail data are used to test and validate the proposed algorithms. Results on ETA prediction are presented for various sets of input features, machine learning algorithms, and prediction locations. ETAs at control points located close to grade crossings are dramatically improved over baseline algorithms, particularly for predictions made multiple hours from a crossing that are useful for proactive safety measures.

17. Document Analysis/Descriptors	18. Availability Statement
Railroad grade crossings, Railroad safety, Railroad vehicle operations, Forecasting, Emergency management, Machine learning	No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	72	

Improving Railroad Grade Crossing Safety: Accurate Prediction of Train Arrival Times for Emergency Response Management and Driver Decision Support

Daniel B. Work[1,3], William Barbour[1], and Ren Wang[2]

[1]*Department of Civil and Environmental Engineering, Vanderbilt University*
[2]*Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign*
[3]*email: dan.work@vanderbilt.edu*

Acknowledgments

The authors acknowledge funding by the Roadway Safety Institute, the University Transportation Center for US-DOT Region 5, which Includes Minnesota, Illinois, Indiana, Michigan, Ohio, and Wisconsin. Financial support was provided by the United States Department of Transportation's Office of the Assistant Secretary for Research and Technology (OST-R).

Contents

List of Figures

List of Tables

Executive Summary

Incidents at *highway-rail grade crossings*, locations where railroad tracks intersect surface streets at grade, are a primary driver of safety in rail transportation in the United States. Addressing safety at these locations through technology is a focus of the Federal Railroad Administration and United States Department of Transportation. Early warning systems can provide enhanced safety information on train arrivals at grade crossings and blocked crossings. Blocked crossings and resulting surface street traffic delays are concerns for emergency services vehicles and, secondarily, the general public. Effective management of emergency response resources on the road network requires knowledge of when trains will arrive at grade crossings and temporarily disconnect emergency vehicles from parts of the community they serve. Generating *estimated times of arrival* (ETAs) for trains at grade crossings on a long time horizon can be used to proactively address surface transportation safety and emergency response management.

This project investigates train delays to accurately estimate train arrival times at grade crossings to support in-vehicle driver alerts shown, for example, on personal navigation devices. However, variability of travel times on the U.S. freight rail network is high due to large network demands relative to infrastructure capacity, especially when traffic is heterogeneous. This project is the first look at the potential for high fidelity freight rail data to be used for arrival time prediction. The prediction of arrival times uses train-positioning information, properties of the train, properties of the network, and properties of potentially conflicting traffic on the network as input. The work is composed of two phases. The first phase focuses on the development of a historical algorithm to accurately model delays using train-positioning information. The second phase of the project develops online algorithms to integrate real-time train position information into the forecasts. Amtrak data and over two years of CSX freight rail data are used to test and validate the proposed algorithms. This report presents the data used in this problem and details on feature engineering and construction for ETA predictions. It also highlights findings on the dominant sources of runtime variability and the most predictive factors for ETA. Results on ETA prediction are presented for various sets of input features, machine learning algorithms, and prediction locations. ETAs at control points located close to grade crossings are dramatically improved over baseline algorithms, particularly for predictions made multiple hours from a crossing that are useful for proactive safety measures.

Chapter 1

Introduction

1.1 Grade crossing arrival times

In 2015, there were nearly 3,000 collisions between vehicles and trains that resulted in approximately 230 fatalities at *grade crossings*, locations where roadways and pathways cross railroad tracks at grade. In the United States today, there are 216,000 grade crossings (Federal Railroad Administration, 2018). Grade crossings are the second highest contributor to rail-related fatalities, after trespassing (Federal Railroad Administration Office of Safety Analysis, 2018); these two causes cover over 90% of rail-related fatalities.

Grade crossings are not just problematic because of safety concerns related to collisions with vehicles. Additionally, occupancy time at grade crossings can be large, which causes congestion on surface streets and, notably, delays and blockages for emergency vehicles. Notably systematic problems have been observed where congested tracks lead to grade crossing blockages of multiple hours (Surface Transportation Board, 2016). Legal disagreements have occurred between state and local entities attempting to assert control over rail grade crossings, but control has rested at the Federal level for train movements (Wronski, 2008). This is true even at grade crossings with roads owned and operated by cities and states.

The Federal Railway Administration (FRA) has undertaken research on the use of intelligent transportation system technology at grade crossings to enhance safety and warnings by providing more complete information from positive train control (United States Department of Transportation and Federal Railroad Administration, 2007). This technology framework, shown in Figure 1.1, was formalized in IEEE Standard 1570-2002 (IEEE Vehicular Technology Society, 2002). Emergency vehicle preemption of road signals is also well-studied and has shown to be effective and have minimal negative impacts, but emergency vehicles do not have the ability to preempt or otherwise influence rail operations (Nelson and Bullock, 2000). Any proactive intervention framework requires that *estimated times of arrival*

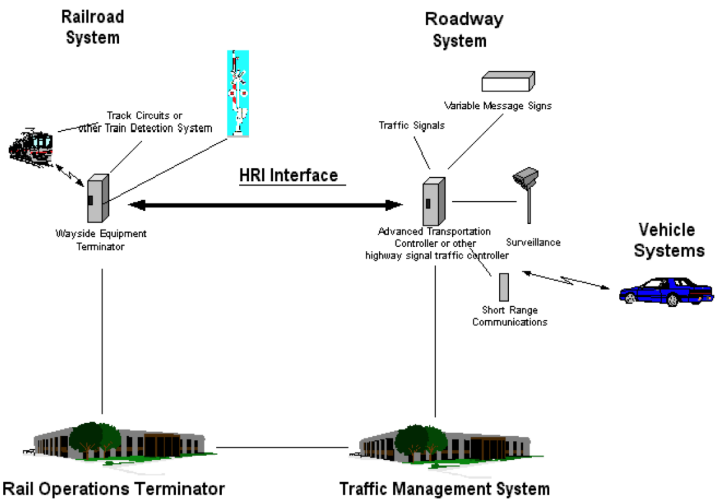


Figure 1.1: IEEE Standard 1570-2002 highway-rail intersection interface overview (IEEE Vehicular Technology Society, 2002).

(ETAs) of trains at grade crossings be generated, motivating our present work.

Grade crossing safety is a challenging problem because freight and passenger trains have large stopping and acceleration distances due to their mass and speed. This problem is compounded by shared corridors, high-speed rail, and increasingly long and heavy freight trains (Chadwick et al., 2014). Large freight trains can require up to a mile of emergency stopping distance, well outside the range of what is reasonable to prevent most grade crossing incidents. Additional risks are created by collisions, as they have the potential to cause train derailments (Chadwick et al., 2012).

Compliance with grade crossing safety measures is not absolute and enforcement is difficult. Drivers do not have good comprehension of safety marking and devices installed at grade crossings (Richards and Heathington, 1988). Moreover, drivers will fail to notice or obey grade crossing warnings (Meeker et al., 1997). There are numerous means by which to improve driver behavior and, thus, safety at grade crossings. Results from Project Lifesaver have shown a positive effect by reducing collisions at grade crossings through awareness and education (Savage, 2006). Detection and enforcement by video camera has also been an area of study (Kim and Cohn, 2004). Predictive models are able to identify particularly problematic or at-risk crossings from a safety standpoint (Medina and Benekohal, 2015).

Safety at grade crossings has been, and continues to be, a priority of the FRA and the United States Department of Transportation (USDOT). In 2015, the FRA announced a partnership with Google to incorporate grade crossing locations into mapping data that many drivers use to navigate (Tumulty, 2015). Brady (2003) incorporated potentially blocked rail crossings into routing during emergency response and management. Also related is the problem of facility location planning for effective emergency response to incidents that include railroad-related events (Ouyang et al., 2018). These large concerns over grade crossing safety demonstrate the importance of proactively addressing safety at grade crossings with respect to the impending arrival of freight trains. For early warning systems or advanced

routing systems to be useful for passenger, commercial, and emergency vehicles, accurate ETAs must be generated for train arrivals at grade crossings. Arrival data at grade crossings is generally not a subject of collection by railroads or public agencies. Additionally, there still exists a technological gap to anticipate train arrivals at grade crossings on a longer time horizon, up to multiple hours. ETA prediction on the necessary longer time horizons requires more than just real-time positioning of a train. It requires the consideration of each individual train, the larger rail network, and interactions between trains on the network.

1.2 Arrival time estimates

Given the importance of ETA prediction for proactive safety, we briefly describe the causes and characteristics of variability on rail networks.

The rail network in the United States has significant infrastructure capacity limitations that cause congestion of the rail traffic. Few rail corridors contain exclusively double (or more) track that allows simultaneous bi-directional traffic (Murali et al., 2010). In comparison, the double and triple track railroads in Europe provide for double the train density of U.S. rail networks (Oliver Wyman, 2016). Many U.S. corridors contain a single track with short sections of double track known as *sidings*, where trains may meet or pass each other. These *movements* (i.e., meets, passes) are implemented in the railroad signaling system but directed by human dispatchers. Dispatchers are experienced with working on specific track corridors, but movements on sidings require planning and precise timing to achieve efficient operations (Vromans et al., 2006; Kecman and Goverde, 2013). Freight volume is expected to increase in the United States, so either infrastructure capacity must be increased or operational improvements must be made to increase capacity (Cambridge Systematics, 2007; Weatherford et al., 2008; Association of American Railroads, 2013).

In addition to the track infrastructure constraints, there are numerous other factors that can contribute to variability of the runtime on a track segment. Traffic heterogeneity and the train priority differences directly influence both the runtime of trains and also the variability in the runtime (Dingler et al., 2009, 2010). Physical characteristics of trains such as the length, tonnage, and power further influence the runtime due to track grade, track curvature, and siding lengths (Dingler et al., 2009). The ability of a train to complete a trip and exit the *line of road* (i.e., the track segments connecting distant terminals) is also influenced by the degree of congestion in the arrival terminal. This is compounded by the possible actions required for the train in the terminal, such as refueling, inspection, switching of cars, or crew change (Dingler et al., 2009; Higgins et al., 1995). Railroad operating strategies such as dynamically scheduled trains and maximizing train length are particularly vulnerable to delay (Lu et al., 2004; Mu and Dessouky, 2011).

In the presence of runtime variability, ETAs are also useful for railroads to improve real-time decision making and the efficiency of the network (Hertenstein and Kaplan, 1991; Hallowell and Harker, 1998). For example, future train schedules can be continually updated to provide new train plans to allow traffic to flow smoothly between terminals

on the network (Kraay and Harker, 1995). Although there are many techniques available to derive optimal schedules (see Goverde (2005) for a thorough review), the schedule may be very sensitive to delays when the network is near capacity. High capacity utilization leads to more complex dispatching where small delays are created, leading to larger deviations from the train plan (Khoshniyat and Peterson, 2017); this is referred to as *knock-on delay* (Vromans et al., 2006; Murali et al., 2010; Goverde and Meng, 2011).

Highly variable runtimes increase operational uncertainty for the railroad and for other transportation systems that are affected by them at grade crossings and otherwise. On the rail network, propagation of delay to other trains is significant (D'Ariano and Pranzo, 2009), and there are large direct costs incurred due to additional operating time alone (Lovett et al., 2015). Delays on the rail network directly influence non-rail transportation services, such as emergency vehicles as already mentioned, if trains must stop and block traffic at grade crossings (Estes and Rilett, 2000). If accurate, real-time ETAs are made available, revisions to the operating plan can be implemented, and surface street transportation services can be re-routed.

1.3 Problem statement and related work

The main focus of the present work is the prediction problem for ETAs at grade crossings on U.S. freight railroads at a long time horizon using real-time data, for use in safety-critical early warning and decision support systems. The estimation problem requires new ETAs to be produced as time elapses and the train progresses down the line of road. Each time the train reaches one of a number of fixed locations on the track, data is collected and a new estimated travel time to a given grade crossing is produced.

To produce the ETA estimate, a variety of routinely collected and maintained data sources available to freight railroads are used. This includes track geometry data (containing grade and curvature information, single and multi-track territory, length of sidings, etc.), historical runtime profiles of all trains, properties of all trains (such as length and tonnage), and crew records. We demonstrate and evaluate prediction methods with commonly-used control point timing data from the railroad, because grade crossing arrival data is not available. These control point data represent the best available large-scale data source for U.S. freight rail that has been available for research. Using other data sources, such as GPS or dispatch data, grade crossing prediction models can be constructed equivalently to the models demonstrated for control points. We explain this modeling choice and its implications further in Section 3.4 and Chapter 7.

Several methodologies to produce ETAs are available, including microscopic simulation (Petersen and Taylor, 1982; Şahin, 1999; Marinov and Viegas, 2011), analytical approaches (Assad, 1980), and data-driven techniques (Bonsra and Harbolovic, 2012). Due to the complexity of the freight rail network (which limits the accuracy of analytical abstractions) and the difficulty in capturing all delay inducing factors in a simulation based model (e.g., decisions made

by human dispatchers, special cases involving priority elevation, unplanned maintenance, and weather), a data-driven approach is proposed in this work (Li et al., 2014). This approach is made possible through access to a large and comprehensive freight rail dataset also described in the report. Well designed data-driven techniques are able to generalize to similar but unseen scenarios to those represented in the training dataset, making them useful for prediction of ETAs during typical operations (Marković et al., 2015). Note, however, that the methods may not extrapolate well to rare and extreme events such as heavy network disruptions, especially when few or no examples exist in the training data.

Many ETA prediction methods for rail transit, buses, and cars have also relied on data-driven algorithms (Altinkaya and Zontul, 2013; Mori et al., 2015) similar to those discussed for freight rail. Liu et al. (2017) achieved station-level arrival time prediction improvements for urban rail transit using neural networks. Sun et al. (2018) also used neural networks in the prediction of large delay events for transit services. Cats and Loutos (2016) evaluated real-time bus arrival prediction schemes with respect to real-world performance and found that deficiencies in prediction accuracy still exist in practice. Zhang et al. (2016) studied travel time characteristics of emergency vehicles and noted the importance of vehicle routing, which is affected by grade crossing arrival time prediction.

There are, however, fundamental differences in operations between rail transit, buses, and cars, and the rail freight traffic considered in our work. Rail and bus transit operations are characterized by frequent stops where delays occur due to the passenger boarding and alighting process (Chien et al., 2002). Buses are also delayed en-route between stations due to traffic signals and other vehicular traffic, which are delay factors for cars as well. Importantly, in the bus system, the vehicular traffic represents an external disturbance to the system. Finally, rail transit vehicles, buses, and cars are generally physically homogeneous (e.g., they have similar performance characteristics and consequently similar dynamics) to other vehicles in their respective classes. These properties are in contrast to freight rail traffic, where the trains are quite heterogeneous with respect to tonnage, power, length, and priority, all of which affect centralized dispatching decisions and, ultimately, delays.

Several lines of research are related to the problem of freight rail ETA prediction. We briefly summarize the most closely related works, and direct the interested reader to the comprehensive reviews available in the works by Bonsra and Harbolovic (2012) and Gorman (2009). The majority of freight trains operate according to a schedule that is constructed in an offline manner and robust to some random unplanned disturbances (Mu and Dessouky, 2011; Vromans et al., 2006; Khoshniyat and Peterson, 2017). When extreme disturbances cause the original schedule to deteriorate, online rescheduling measures must be implemented to account for the delay and to maintain robustness to further delay (D'Ariano et al., 2007; Hallowell and Harker, 1998; Khadilkar et al., 2017). Numerous efforts are aimed at understanding and quantifying the causes of delay that influence scheduling, rescheduling, and predictability (Murali et al., 2010; Chen and Harker, 1990; Dingler et al., 2010). Delay is typically formulated in terms of deviation from a train schedule or historical performance, but it can be extended to arrival time prediction for individual trains (Bonsra and Harbolovic, 2012).

Several works have proposed to empirically produce delay or runtime estimates using historical data for passenger rail networks. Kecman and Goverde (2013) propose an ETA prediction framework for passenger rail arrival time prediction using track occupancy data for conflict evaluation. In Kecman and Goverde (2015), track occupancy variables are used along with schedule and delay data in the real-time prediction of running time and dwell time estimation for passenger rail in the Netherlands. Statistical models including robust linear regression, tree-based non-linear regression, and random forests are each applied to running time and dwell time estimation and an emphasis is placed on the importance of location-specific models. Chapuis (2017) uses artificial neural networks to predict arrival times of frequent passenger trains using historical train and station delays. Compared to the proposed work, the ETAs are evaluated in the Netherlands and France, respectively, on high-priority passenger traffic (Furtado, 2013; Pouryousef et al., 2015), which also operates with higher punctuality compared to the freight or passenger traffic in the United States (Amtrak, 2016). Marković et al. (2015) use support vector regression on passenger railways in Serbia in order to identify relationships between delay at a station and various internal factors (i.e., related to the train and to the railroad) and external factors. The predictive ability of SVR is compared to that of an artificial neural network and is shown to have better performance and maintain interpretability of the model. Wang and Work (2015) estimate passenger rail delays on the Amtrak passenger rail network in the United States using vector regression techniques and only historical runtimes between passenger stations. The regression problems are formulated in both a historical and online perspective, but the feature set for prediction is limited and does not contain any data on the freight traffic, which constitutes the majority of traffic on the shared line of road in the United States. Online methods presented for passenger rail, accessible because of the data stream created by station arrivals and departures, have not been fully extended to freight rail. Additionally, the magnitude of delay and ETA error for passenger rail is typically on the order of minutes, while delay and ETA error for non-priority freight rail traffic may exceed multiple hours.

The most closely related estimation works on freight trains are the works of Gorman (2009) and Bonsra and Harbolovic (2012). In Gorman (2009), an econometric analysis of free-running and congestion-related factors are used to identify the primary causes of delay. The data is partitioned by geographic area and priority groupings. Congestion-related factors, such as meets, passes, and overtakes, are found to have the largest effect on delay. Bonsra and Harbolovic (2012) predict runtimes for individual freight trains in an offline setting using regression. Prediction improvements are attained when estimated at the time of departure. The regression model used train and network parameters and a historical runtime averaging technique for evaluating model performance.

1.4 Outline and contributions

The main contribution of this work is to show how to pose the ETA prediction problem at grade crossings on a rail network as a machine learning regression problem and to provide results indicative of predictive performance

across a range of time horizons and various machine learning algorithms. ETA updates occur at fixed timing points on the network and can be generated corresponding to any other timing point, which can be grade crossings, or to major destinations on the network. We provide practical insights by highlighting the datasets available to perform the prediction and describing some of the feature engineering required when the feature vectors change in time and space. We present a set of data features and several machine learning regression algorithms used to achieve more accurate ETAs than common statistical methods yield. Finally, the resulting models are discussed in detail with respect to their performance.

Due to multiple approaches taken in this work, we first present a single origin-destination modeling framework that is straightforward to understand, followed by a unified all-origin framework that can leverage larger training datasets and predict ETAs from multiple locations in a single model. Both of these models are applied first to ETA predictions made to train destinations at yards and terminals, so that we may benchmark machine learning techniques against existing algorithms used by the railroads and compare the performance of various machine learning algorithms. We then use these same models, which outperform existing techniques used in practice, to make predictions to grade crossings. Also included in this report is preliminary work on ETA prediction for passenger trains. This work is included in Appendix A. While it demonstrated improved prediction accuracy, the publicly available Amtrak data was not sufficiently informative, because it has course resolution and does not cover freight trains operating on the same tracks that are known to have a significant impact on passenger train performance.

Specifically, the remainder of the report is organized as follows. Chapter 2 presents the framework used to process and operate on the various data sources and the preliminaries for the machine learning regression. Chapter 3 discusses the datasets and the work that is necessary to process the data for use in the machine learning framework. Chapter 4 details the model experiments that are conducted as well as the means by which to evaluate them. Chapter 5 describes the process for tuning and evaluating one example of the origin-destination models, which is then extended to models across the full testing route; results are given for models using select feature combinations. Chapter 6 formulates and tests a unified framework to combine models and leverage larger quantities of data, and also compares performance of a variety of algorithms. Chapter 7 directly addresses the problem of prediction of ETAs at grade crossings using the lessons learned on prediction to a single destination. We provide a summary and discuss future lines of research in Chapter 8. Appendix A discusses the ETA prediction for passenger trains using both a historical and online data-driven approach with Amtrak data.

Chapter 2

Framework and Problem Formulation

This chapter briefly describes the machine learning framework used for ETA estimation. It reviews general machine learning terminology and parameters specific to the algorithms, both used later during analysis and discussion.

2.1 ETA Machine learning framework

The problem of predicting an estimated time of arrival for a train from an origin point to a destination point on the rail network is posed as a supervised machine learning regression problem. The goal of the regression problem is to predict the true runtime $y(i) \in \mathbb{R}$ of a train i given the properties of train i, the network, and other traffic on the network, which are contained in the feature vector $x(i) \in \mathbb{R}^n$. Given a dataset of m trains with true runtimes $Y = [y(1), y(2), y(3), \cdots, y(m)]^T \in \mathbb{R}^m$ and corresponding feature vectors $X = [x(1), x(2), x(3), \cdots, x(m)]$, where $X \in \mathbb{R}^{n \times m}$, the machine learning regression problem is to find a mapping $f_w \colon \mathbb{R}^n \to \mathbb{R}$ parameterized by w such that $f_w(x(i))$ is an accurate predictor of $y(i)$. In general, supervised machine learning regression uses a set of *training data* $\{X_{tr}, Y_{tr}\}$ (where the subscript *tr* is used to indicate the training data) to learn the function f_w, by minimizing a prediction error measure between $f_w(X_{tr})$[1] and Y_{tr} over the m records in the training data.

The machine learning model f_w must generalize (i.e., make good predictions on data that has not been used to train the model), in order to maintain high accuracy on new data and to avoid *overfitting* the training data. To test the degree of generalization, the accuracy of the prediction is assessed on hold out *test data* $\{X_{te}, Y_{te}\}$, which is not used to train the model.

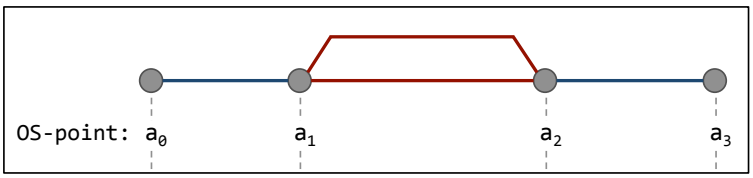
[1] With a slight abuse of notation, we overload the function f_w to also operate on the entire dataset X_{tr}.

2.2 Generating ETAs on a freight rail network for grade crossings

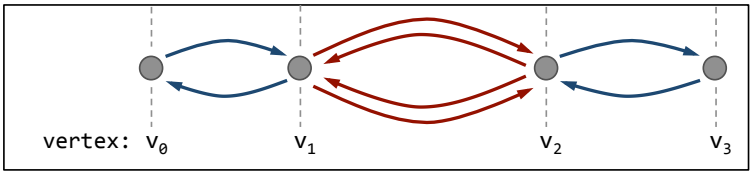
The central difficulty of posing the train ETA prediction problem into the framework above stems from the fact that many of the features used for prediction change in time and in space as the train moves towards the destination. For example, the amount of traffic on the line of road will change as trains enter and leave the line of road. The number of available sidings on a route in single track territory also changes across the route and as other trains occupy or vacate sidings. If a single model is used for all origin-destination predictions in the network, it may be difficult to predict area-specific delays (e.g., due to local dispatching decisions and route characteristics) that may not occur throughout the network; results pertaining to origin-destination specificity are discussed in Section 5.2. Moreover, because some features change over time (as described above), while others may not (e.g., train priority), construction of an unbiased training dataset is a nontrivial engineering task. For example, one cannot simply create a new training data point each time a single property of a train changes (e.g., corresponding to a new feature vector) without biasing the training data, since the feature vector still corresponds to a single train trip.

To address these difficulties, we propose to build a distinct regression model for each origin-destination pair for which predictions are required, where the ETA corresponds to the estimated time of arrival at the next destination (i.e., major terminal) for trains passing the corresponding origin point on the network. The resulting models are all of the same form and differ only in feature weights and hyper-parameters. Because the models are independent, each model can be trained using all trips that pass between the corresponding origin-destination pair by constructing features according to the state of the train and network at the time the train reaches the origin point. Localized and geography-specific performance effects may be captured in the individual models without explicitly constructing them in the feature vector. For example, longer travel times will be observed for heavy or under-powered trains in areas of high track grade. Feature construction can also vary between models (e.g., in dimensionality) since each uses a custom dataset built for the origin-destination pair.

The primary disadvantage of building a model for each origin-destination pair in the network is the number of models required. In a rail network with k nodes, at minimum k^2 origin-destination predictors could be required (possibly more if multiple paths exist between each origin-destination pair). In contrast to road networks where the number of nodes and the number of viable paths between any two node pairs may be large, rail networks have fewer nodes and less path redundancy. In practice, few locations are relevant destination points from a given origin because a single route (excluding small deviations for sidings) is typically used to connect two points on the network. The freight rail network in the United States is sparsely connected in most regions, particularly with regard to high volume routes; isolated corridors connect major terminal points on the network where most crew changes and switching work occur. Therefore, the number of origin-destination paths for which predictions are required is tractable. In the area of study in this work, there exist 35 points that can serve as origin points. This results in 35 ETA updates for a train



(a) Track infrastructure representation assembled from GIS data.



(b) Vertex-edge graph constructed from track infrastructure data.

Figure 2.1: Graph vertices align with OS-points and each track segment between them is represented by a graph edge in each direction. Double track areas and sidings, therefore, are represented by four graph edges.

traversing the corridor. There are less than four practical destination points from each origin point, which results in at most 140 predictors as opposed to $35^2 = 1{,}225$. For a realistic sized network, we estimate a total of 10,000 models are necessary for all ETA predictions.

In order to map spatiotemporal train data to the network topology, infrastructure data can be reconstructed into a directed graph format, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of directed edges (Kecman and Goverde, 2015). Vertices are points where the track merges and splits (e.g., endpoints of sidings). Grade crossings are each associated with a single vertex. Data on passing trains is recorded at *OS-points*, which are fixed locations $v \subset \mathcal{V}$; OS-points serve as the origin points in the origin-destination models. Directed edges represent track segments across which trains travel between OS-points and a direction of travel. This alignment between track infrastructure and the constructed graph is shown in Figure 2.1. In the figure, OS-points are denoted a_0, a_1, a_2, and a_3 with corresponding graph vertices v_0, v_1, v_2, and v_3. All tracks are delineated between these OS-points with multiple tracks such as the main line and siding between a_1 and a_2 remaining distinct. Pairs of directed edges representing each delineated track segment allow different runtimes and feature values in each direction of travel, which is necessary when properties such as grade are considered (Bonsra and Harbolovic, 2012). This results in two directed graph edges for a single track and four or more edges for an area of multiple tracks such as the siding between vertices v_1 and v_2. In this formulation, all trains can be routed on the graph across their unique path considering track usage (e.g., siding track versus main line track). Data can be gathered on the behavior of trains for each directed edge with respect to speed, grade, train occupancy, and other location/direction specific attributes. Also, features that consider estimates of the positions of multiple trains and track topology can be mined from this data.

2.3 Support vector regression

The regression problem of predicting an ETA from a vector of features is proposed to be solved via *support vector regression* (SVR), introduced in (Drucker et al., 1997). Support vector regression is a popular machine learning algorithm grounded in statistical learning theory and for which training is efficient due to the convexity of the training problem. The optimal model parameters in linear SVR are straightforward to interpret and are unique (Burges and Crisp, 2000), which can be invaluable in the application of the algorithm. Additionally, the SVR formulation provides for extension to nonlinear regression via kernel functions. The intent of this work is not to demonstrate superiority of SVR to other algorithms, but to apply a well-studied algorithm to the data-driven ETA prediction. The precise algorithm that should be implemented in a live production system would depend on performance and additional practical factors, such as computation time, memory requirements, and more. Other applicable algorithms include linear ridge regression (Hoerl and Kennard, 1970), elastic net regression (Zou and Hastie, 2005), kernel ridge regression (Saunders et al., 1998), random forests (Kecman and Goverde, 2015), and neural networks (Marković et al., 2015), to name a few.

The training step in a generalized regression problem may be written as:

$$\min_{w} \quad L(f_w(X) - Y) + \|w\|, \tag{2.1}$$

where L is a loss function measuring the quality of the predicted output, $f_w(X)$, relative to the true output, Y, and the feature weights w are penalized via a norm to avoid overfitting the training data. This characteristic is informally referred to as model *flatness* (Basak et al., 2007).

SVR is a special case of (2.1) and uses a two-norm on w and in the simple case assumes an affine predictor of the form $f_w(x) = w^T x + b$, where $w \in \mathbb{R}^n$ and the offset $b \in \mathbb{R}$. SVR uses an *ε-insensitive loss function* $|\cdot|_\varepsilon$, which penalizes prediction residuals $r = y - f_w(x)$ larger than a threshold defined by ε. The loss function is constructed as (Cortes and Vapnik, 1995):

$$|r|_\varepsilon = \begin{cases} 0 & \text{if } |r| \leq \varepsilon \\ |r| - \varepsilon & \text{otherwise.} \end{cases} \tag{2.2}$$

The ε-insensitive loss function quantifies the distance between a prediction and the band created by $y \pm \varepsilon$. For a vector of residuals, the sum of the element ε-insensitive losses are used as the loss function.

The training step in SVR can be reformulated as computing the weights w and offset b by solving the following

convex optimization problem:

$$\begin{aligned} \underset{w,b,\xi,\xi^*}{\text{minimize}} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}(\xi(i)+\xi^*(i)) \\ \text{subject to} \quad & y(i) - w^T x(i) - b \leq \varepsilon + \xi(i), \quad \forall i \\ & w^T x(i) + b - y(i) \leq \varepsilon + \xi^*(i), \quad \forall i \\ & \xi(i),\ \xi^*(i) \geq 0, \quad \forall i \end{aligned} \tag{2.3}$$

where $\xi, \xi^* \in \mathbb{R}^m$ are variables introduced to rewrite the ε-insensitive loss (2.2) as linear inequalities. The total ε-insensitive loss (i.e., accuracy of model fit) is balanced against model flatness by a scalar factor C.

The optimization problem (2.3) can be solved via the dual problem, yielding the optimal dual variables $\alpha, \alpha^* \in \mathbb{R}^m$. These dual variables are related to the feature weights such that $w = \sum_{i=1}^{m}(\alpha(i) - \alpha^*(i))x(i)$. This results in a predictor of the form:

$$f(x) = \sum_{i=1}^{m}(\alpha(i) - \alpha^*(i))x(i)^T x + b; \tag{2.4}$$

see (Scholkopf and Smola, 2001) for a comprehensive description.

When the ETAs in the training data are not linearly related to the features, an alternative strategy is to transform the training data into a much higher dimensional feature vector denoted by $\Phi(x)$, where $\Phi\colon \mathbb{R}^n \to \mathbb{R}^N$ with $N >> n$, which can then be used for regression. The new predictor becomes:

$$f(x) = \sum_{i=1}^{m}(\alpha(i) - \alpha^*(i))\Phi\left(x(i)\right)^T \Phi(x) + b. \tag{2.5}$$

Interestingly, it is not necessary to explicitly define the mapping to the high dimensional space, since only the inner product $\Phi^T\Phi$ is needed in the regression function. The inner product can instead be defined through a kernel function $K(x(i), x(j)) = \Phi\left(x(i)\right)^T \Phi(x(j))$. The use of the kernel function directly in (2.5) is known as the *kernel trick* (Burges, 1998) in machine learning. In the present work, we adopt the *radial basis function* (RBF) kernel (Boser et al., 1992) of the form:

$$K(x(i), x(j)) = \exp\left(-\frac{1}{2\sigma^2} \left\| x(i) - x(j) \right\|_2^2\right), \tag{2.6}$$

where σ is a parameter controlling the decay rate of the kernel, effectively limiting the influence that any single observation may have on the trained model.

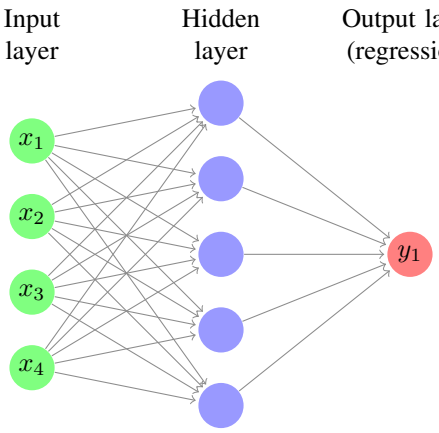


Figure 2.2: Simple Feed-forward neural network with one hidden layer.

2.4 Random forest regression

Random forest regression is an ensemble algorithm that constructs a series of regression trees using randomly sampled subsets of the training data for each tree and a subset of available data features for splitting within trees (Breiman, 1984, 2001). Regression trees are constructed by splitting data samples at each node in the tree according to values of input features. Each node resulting from the split more effectively isolates data samples with similar output values. The best split is determined by a minimization of resulting prediction error. Nodes are no longer split when the number of samples in the node falls below the minimum or the decrease in prediction error falls below a defined threshold. The predicted output value for each terminal node in the tree is calculated from the corresponding training samples that terminated in the node. The predictions made by individual trees are averaged to arrive at the ensemble prediction. Combining many weak learner regression trees in the random forest predictor has shown to be an effective methodology and helps avoid overfitting (Liaw and Wiener, 2002; Oruganti et al., 2016).

2.5 Deep feed-forward neural network model

A neural network consists of multiple neurons organized in layers, with individually weighted connections between neurons in adjacent layers. At minimum, a feed-forward neural network consists of three distinct layers: the input layer, one hidden layer, and an output layer consisting of one node for regression, as shown in Figure 2.2, or multiple nodes for classification. A deep feed-forward neural network has multiple hidden layers. The depth of the model is determined by the number of hidden layers. After being processed at the hidden layer nodes, their outputs are forwarded to the output layer which then makes a prediction, according to its activation function. This feed-forward process is characteristic of the neural network and is used for making predictions, given an input vector. The training phase of a neural network is comprised of selecting the optimal weights for each of the connections between the

neurons. More specifically, given the input vector at the input layer, and the known output that actually occurred, the problem is defined as choosing the weights for the connections between neurons so as to minimize the error between the prediction and actual observation. Gradient-based optimization is used for training the neural network and choosing weights.

Chapter 3

Feature construction and data cleaning steps

This chapter discusses the process of combining and mining datasets to be used in feature construction. The data used in this work is described first, before describing the features that are calculated from the datasets which are subsequently used to train the machine learning ETA algorithm. Due to the proprietary nature of the data, some values are reported in relative terms.

3.1 Description of raw data

This work uses a collection of datasets describing the rail network and operations from December 1, 2014 through January 31, 2017 inclusive. It consists of freight train movement, train car operations, crew, and locomotive data in the CSX Transportation network extracted from dispatching, operations, and signaling data.

The movement data consists of records generated at OS-points between terminals. The data includes the track on which the train was reported and the time at which the train triggered the OS-point. This dataset also contains information about track mileage covered, direction of travel, and the next destination at which the train is scheduled to stop. OS-points have spacing between 1 and 10 miles and typical temporal spacing between 1 and 20 minutes. Typical runtimes for the full route vary between 5 hours and the maximum crew time of 12 hours.

The train car operations data details the actions performed on the train once it enters a terminal from the line of road. This includes the switching operations (i.e., picking up and setting out rail cars) that are referred to as train *work*, inspections, refueling, and crew changes that are scheduled to occur and may incur delay in getting track space in the terminal. The planned work schedule, as well as adherence to the schedule, is reported in this data. Changes in

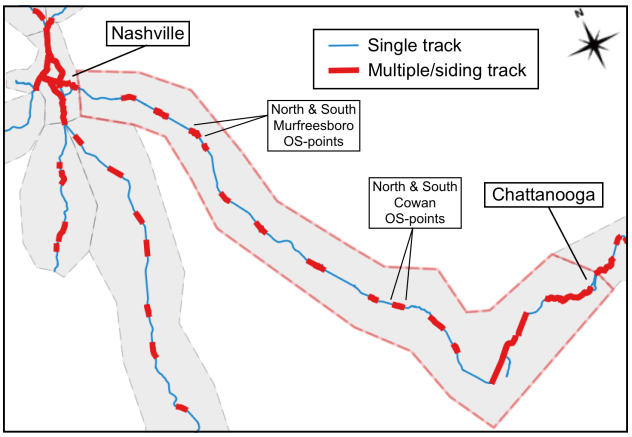


Figure 3.1: GIS network view depicting a portion of the Nashville division, with multi-track segments shown in bold red lines and single-track segments in thin blue lines. The primary study route is bounded by the red dashed line. OS-points at Murfreesboro and Cowan are also shown on the map, each of which corresponds to a point at which the ETA to Chattanooga is updated.

physical train characteristics (e.g., total number of cars, length, tonnage) are inferred based on the work reported on the train.

Crew data contains information about the crew assigned to the train, the originating location, the time at which they were called on duty, and the time at which the crew must legally go off duty (i.e., 12 hours after going on duty). The time between a crew going on duty and the departure of the train is non-negligible and is referred to as *on duty time to departure* (ODTOD). Crew information is important because the maximum crew on-duty requirement must always be satisfied, even at the large expense of stopping a train and transporting a replacement crew to finish the trip.

Locomotive assignment data indicates the equipment and total locomotive power available on each train, which can be important for predicting delays in regions with high grades.

This work also uses GIS data describing the physical infrastructure of the network, which includes individual tracks, switches, mileposts, and terminals. All locations referenced in the movement, work, crew, and locomotive data map to physical infrastructure locations, such as track mileposts and control points. Reconciling these GIS data sources is necessary to gather data on the number of tracks and siding locations and lengths on each route and build the network graph described in Section 2.2. Figure 3.1 depicts this data, along with the distinction between single track sections (shown by the thin blue lines) and multiple track sections or sidings (shown by the bold red lines) for a portion of the rail network. The primary study area, from Nashville, TN, to Chattanooga, TN, is bounded by the red dashed line. There are 14 distinct sections of multiple track in the study area. Four OS-points at North and South Murfreesboro and North and South Cowan are also shown on the map, each of which corresponds to a point at which the ETA to the rail yard in Chattanooga is updated.

Grade crossing raw data is in the form of GIS polygons, which correspond to the track in the coordinate reference system (CRS). See Figure 3.2 for an example of one of these grade crossing data points, where the crossing is

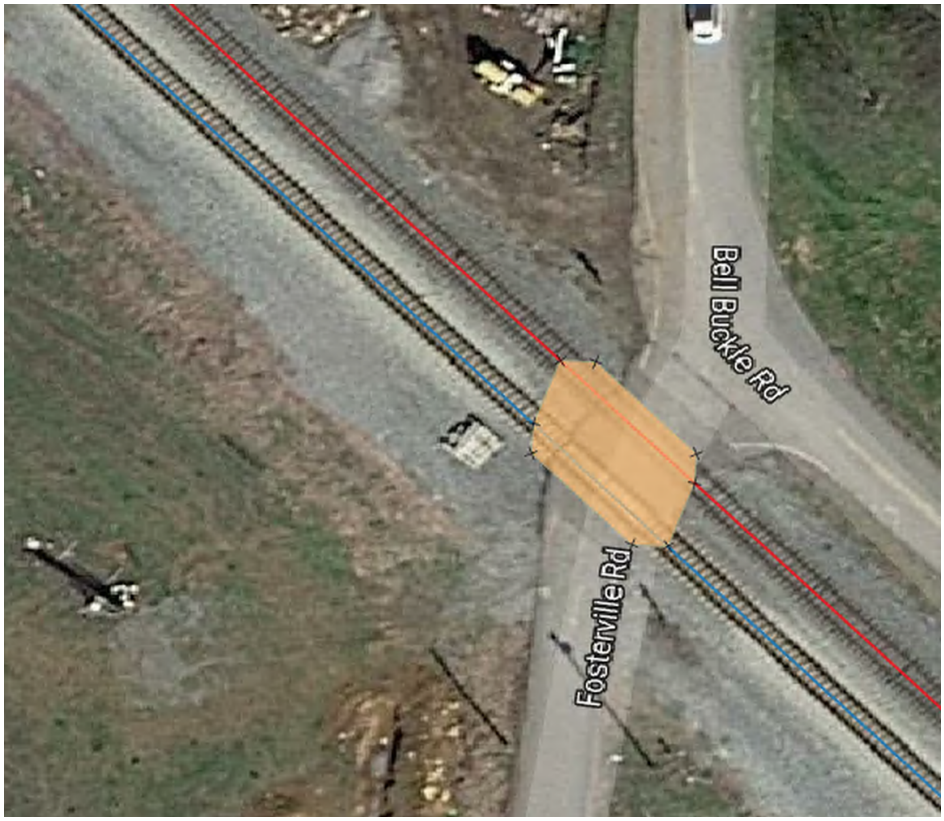


Figure 3.2: A single grade crossing boundary delineated by the orange shaded area and overlaid on a satellite imagery basemap. The tracks at this location are shown by the blue line (main line track) and red line (siding track).

delineated by the orange boundary.

3.2 Data cleaning and standardization tasks

A variety of data cleaning and data transformation tasks are necessary to organize the input for any prediction algorithm. With over 10,000 trips on the study route in a two year period, we decide to neglect trips with data completeness issues or data errors instead of devising a scheme to impute values; this resulted in the discarding of approximately 10% of trains. Errors consist of fields that contain missing data, or fields that contain illogical values. Examples include non-physical train lengths, or an arrival time prior to the train departure time.

The GIS data is examined to ensure proper connectivity and accuracy before being transformed into the network graph. Common errors encountered include duplicate geometries, disconnected track components, and minor mislabeling of infrastructure components. Many errors are automatically identified and resolved, while some errors require manual correction.

The detection and resolution methods for each of these data fields are summarized in Table 3.1. Each is implemented at the time of data mining and feature construction, so that an origin-destination dataset is clean at the time of

Table 3.1: Data cleaning and standardization steps

Data field	Data error	Data correction
Train arrival time	arrival time $\leq$ departure time	discard train
Train length	length ≤ 0	discard train
Train tonnage	tonnage ≤ 0	discard train
Train horsepower	horsepower ≤ 0	discard train
Crew assignment	no crew assigned to train	discard train
Crew on duty time	on duty time $\geq$ train departure time	discard train
Track segments	polylines connect spatially, but not by endpoint ID	detect spatial connection, resolve endpoint ID mismatch
Duplicate GIS elements	identical geometry encoded strings	check connectivity of each, keep most connected element

model training.

3.3 Handling of recrewed trains

In the process of early prediction efforts and data exploration efforts, a dominant source of runtime variability has been discovered. Specifically, we find that *recrewed* trains (i.e., a train that did not reach its destination before the crew reached its maximum on-duty time and needed a relief crew) define the dominant source of runtime variability on the study route.

To further investigate the impact of recrews on train variability, all trains are ex post facto labeled as either recrewed or non-recrewed. Less than 10% of the trains on the route were recrewed. The two classes (recrewed and non-recrewed trains) are separated and descriptive statistics are calculated for each class at each of the 35 OS-points, which are spread at irregular intervals across the route depicted in Figure 3.1. The standard deviation of runtimes is used to quantify the runtime variability of trains in each class as well as the variability of all trains in the dataset (not separated on recrew). As shown in Figure 3.3, the runtime variability of the recrewed trains is several times larger than that of the non-recrewed trains across all OS-points, which are ordered from Nashville to Chattanooga; runtime variability is expressed as a relative value to protect proprietary operational properties in the data. Despite recrewed trains representing less than 10% of the trips, they represent 53% of the variability within the dataset of all trains, when averaged across the full route.

Recrewed trains introduce high variability in runtime and their runtimes are not predictable by features inside the scope of the train and network state features (e.g., status of crew pools from which to reassign and the locations and availability of taxis to transport the crew to the train are significant factors and are not in our dataset). It is not a good

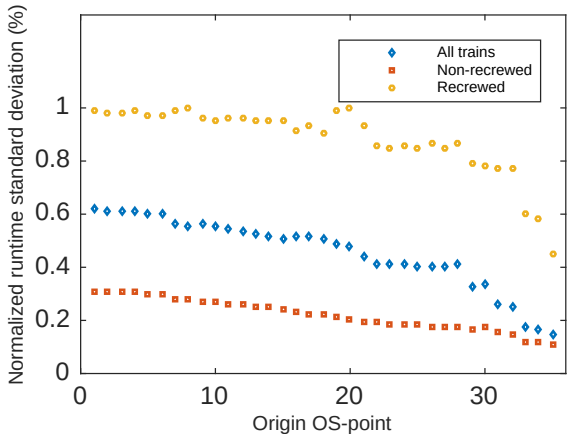


Figure 3.3: Comparison of variability in runtime, between recrewed trains, non-recrewed trains, and all trains, for each origin OS point, ordered from Nashville to Chattanooga.
Standard deviation is normalized by the largest variance OS-point, OS-point #21. OS-point IDs increase from Nashville (1) to Chattanooga (42).

idea to include the recrewed trains in the dataset because they have extreme delays, and predicting the duration of the delays is not possible without additional features (e.g., location of the replacement crew). If the recrewed trains are included in the training data, the model will be harder to train because the large error caused by the few but extreme outliers cannot be reduced under any parameter set of the model. Of course, if an outlier robust machine learning algorithm were used in place of the SVR approach, it would be possible to leave the recrewed trains in the training dataset, where they would be effectively ignored. It is likely, however, that the circumstances leading to a recrew will enable its preemptive classification and could be captured with the available data. It should be noted that features calculated from crew information contain the implicit knowledge that the train was not recrewed, given that the training data was cleaned of recrewed trains; this fact further motivates the need for preemptive classification of recrew events.

3.4 Grade crossings data and locations

Grade crossings are frequent elements of the rail network. They are spaced at uneven intervals on virtually every segment that is operated; for example, see a sample of grade crossings outside of Nashville, TN, in Figure 3.4. For each grade crossing, data exists on its precise location and extents, as shown in Figure 3.2. However, train timing is not reported at precise grade crossing locations, only at nearby OS-points through the dispatching system. Therefore, we associate each grade crossing by minimum distance with an OS-point, at which we have train timing data to train and test models. The number of grade crossings associated with each OS-point between Nashville, TN, and Chattanooga, TN, are shown in Figure 3.5, which shows a very uneven distribution.

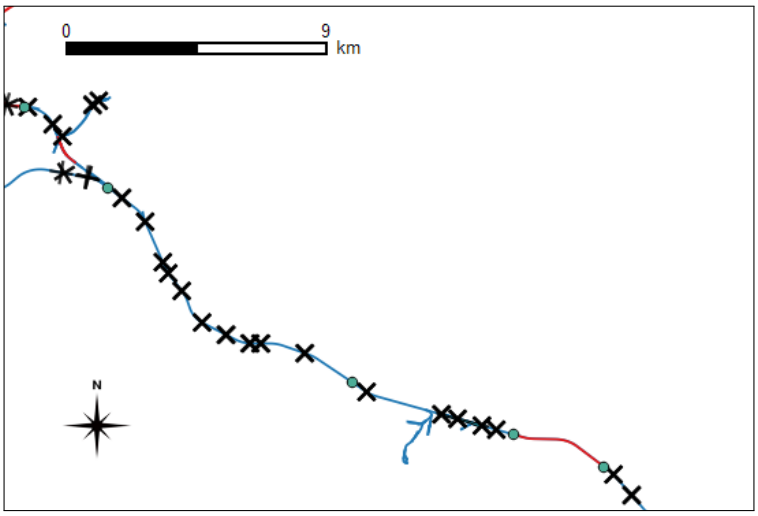


Figure 3.4: Sample of grade crossings, denoted by 'x' symbols, outside of Nashville, TN.

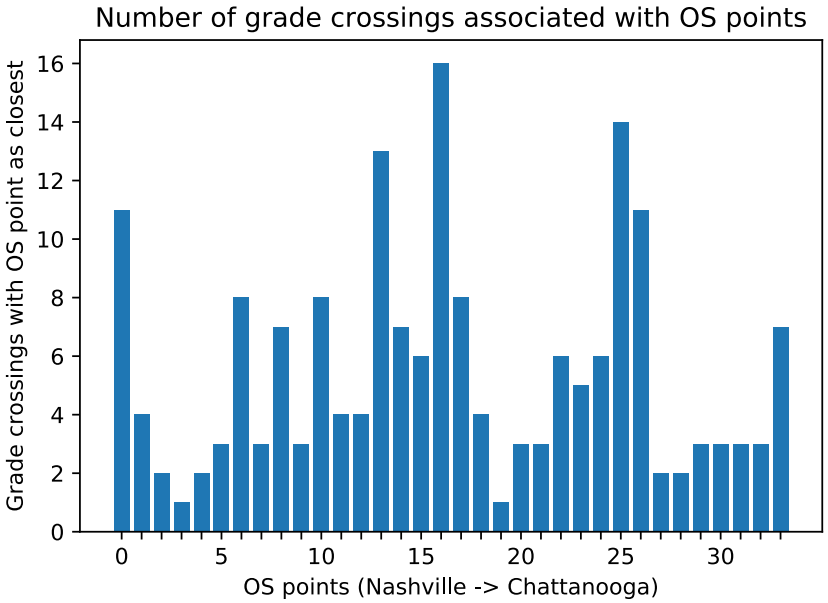


Figure 3.5: Number of grade crossings associated by minimum distance with OS-points between Nashville, TN, and Chattanooga, TN.

Table 3.2: Summary of implemented scalar features.

Feature	Notation	Description
Train length	$\lambda(i)$	The total length of locomotives and cars of train i.
Train tonnage	$\mu(i)$	The total mass of locomotives and cars.
Train horsepower per ton	$\eta(i)$	Total horsepower of locomotives divided by train tonnage, $\mu(i)$.
Train priority (high-resolution)	$\rho_{20}(i)$	Priority ranking on a 1-20 scale.
Train priority (medium-resolution)	$\rho_5(i)$	Five priority classes are constructed by aggregating the high-resolution priority ranking.
Train priority (low-resolution)	$\rho_3(i)$	Three priority classes are constructed by aggregating the high-resolution priority ranking.
Crew time remaining	$\gamma(i)$	Amount of time remaining that the current train crew can legally work.
On duty time to departure	$\theta(i)$	Time between crew on duty time and train departure.
Full traffic count	$\tau(i)$	Count of trains on the remaining line of road.
Directional traffic count	$\tau_\omega(i), \tau_\psi(i)$	Count of trains on the remaining line of road, categorized by direction of travel (i.e., in the same direction, ω, or in the opposite direction of travel, ψ).
Prioritized directional traffic count	$\tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i)$, $\tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i)$	Count of train on the remaining line of road, categorized by both direction of travel and priority relative to that of the train being predicted (i.e., lower or equal priority, β, or higher priority, α).
Available sidings	$\pi(i)$	Count of sidings on route with length greater than that of train i.

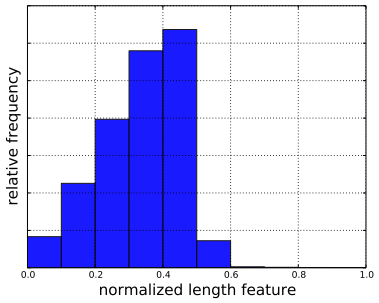
3.5 Calculated features

This section lists and discusses the features that are generated from the raw data in Section 3.1 and used to train machine learning algorithms in ETA prediction. A summary of the implemented scalar features appears in Table 3.2. These features include six train characteristics, two features that capture the state of the crew on each train, and multiple scalar features quantifying the network characteristics and traffic. We also propose features to describe the traffic state of the network in the geographic vicinity of prediction as vector quantities. Each element of the traffic feature corresponds to the traffic at a track segment between adjacent OS-points. The network traffic state is quantified in terms of occupancy, direction, and priority; these are summarized in Table 3.3. All of the features chosen for exploration are based on extensive discussions with operations research personnel from CSX Transportation.

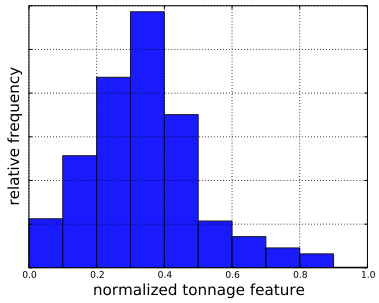
The priority of a train is determined by its cargo (e.g., bulk, merchandise, automotive, intermodal), its type of service (e.g., local, yard, road), and its load status (e.g., loaded, empty). The priority ranking is fully defined for all

Table 3.3: List of calculated and implemented track segment feature series, which correspond in dimension to the segmentation of the remaining route, and details for each.

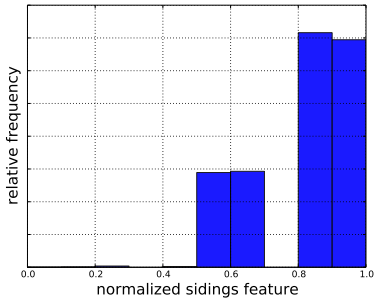
Feature	**Notation**	**Description**
Track segment occupancy	$O(i) = [O_1(i), \cdots, O_l(i)]$	Vector of elements denoting whether a segment on the origin-destination route, indexed 1 to l, is occupied by another train; non-zero when occupied.
Occupying train direction	$D(i) = [D_1(i), \cdots, D_l(i)]$	Denotes the direction (same or opposite) of a train occupying a track segment on the origin-destination route, indexed 1 to l; zero if segment is unoccupied.
Occupying train priority	$P(i) = [P_1(i), \cdots, P_l(i)]$	Assigns high-resolution train priority values to a train occupying a track segment on the origin-destination route, indexed 1 to l. Higher value for high priority train; zero if no train.
Relative priority of occupying train	$R(i) = [R_1(i), \cdots, R_l(i)]$	Non-zero when a train occupying a track segment on the origin-destination route, indexed 1 to l, has higher priority than the train for which the ETA is being predicted. Reflects likelihood of meet/pass delay.
Track segment occupancy around origin point	$G(i) = [G_{-1}(i), \cdots, G_{-h}(i)]$	Indicates track segment occupancy for segments -1 to $-h$ around the origin point, but not included in the primary route (segments 0 to l); captures trains that may enter the primary route and pass/overtake.
Track segment occupancy around destination point	$E(i) = [E_{l+1}(i), \cdots, E_{l+p}(i)]$	Indicates track segment occupancy for segments $l+1$ to $l+p$ around the destination point, but not included in the primary route (segments 0 to l); captures trains that may enter the primary route and conflict during the trip.



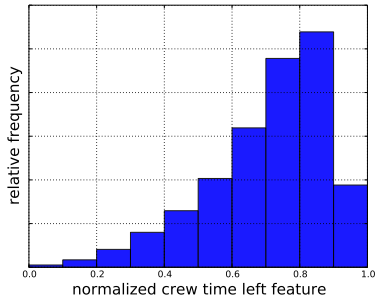
(a) Distribution of train lengths.



(b) Distribution of train tonnage values.



(c) Distribution of number of remaining sidings in which each train will fit.



(d) Distribution of crew time remaining for active train crews.

Figure 3.6: Frequency distributions of various scalar features in the training dataset taken at OS-point #1, outside of Nashville.

trains that run on the network and includes exceptions and specialty trains that run infrequently. The priority ranks are aggregated to different levels of granularity. At the highest resolution, all trains are placed into one of twenty priority classes. The relative priority ranking is understood to be non-linear based on its construction. The high resolution ranking is aggregated to a medium-resolution ranking using five priority classes and a low-resolution ranking of three priority classes. For example, scheduled merchandise trains have significantly higher priority than bulk/unit trains (e.g., loaded coal train), but in medium- and low-resolution classifications, the two types will get the same priority designation. The priority ranking and aggregations were provided by CSX Transportation.

The physical train characteristics such as train length and train tonnage are calculated by examining the work data, which contains the train dimensions after the most recent work was completed. The distributions of these parameters (shown in Figures 3.6a and 3.6b) demonstrate that there is a preferred maximum threshold for train length, but that train tonnage is subject to a significant tail of very heavy trains. Train length (together with the track geometry) factors into train runtime, in part because it determines the number of sidings in which a given train may fit. While the majority of sidings are sufficiently long for any train, some sidings are too short to accommodate the longer trains. This disparity is reflected in Figure 3.6c, where it is shown that the majority of trains are able to use 80-100% of sidings, while some trains fit into only 50-70% of the sidings.

The most recent crew change can be identified in the crew data, which is then used to calculate the *crew time remaining* (i.e., the maximum time the current crew may continue to operate the train). The distribution of the crew time remaining is shown in Figure 3.6d. Ideally, this parameter should be maximal when starting a route to give the best chance of completing the trip without needing a replacement crew. Note also in Figure 3.6d that all trains used to construct this distribution are not recrewed. When combined with the expected train runtime, it is possible to compute a crew *slack time* (i.e., the difference between the crew time remaining and the expected runtime). If the slack time is negative, but is not recrewed, it means the train ran the route faster than the average train. As the slack time becomes more negative, it is increasingly likely that the train will need to be recrewed. In terms of feature construction, the crew time remaining feature and the slack time feature are equivalent under min-max normalization (which is applied to all features in the dataset). Consequently, only the crew time remaining feature is used in the models presented in this work.

It is expected that the traffic along the route influences the runtime, and consequently we propose several methods to quantify the traffic. First, we construct six scalar measures of traffic, each of which consider only the track along the route between the current location of the train and the destination. The six measures differ in the degree of granularity. For example, in the first two measures, we count all other trains (including local trains), which are categorized based on their direction of travel (e.g., same direction, subscript ω, or opposite direction, subscript ψ, relative to the train being predicted). We also consider the fact that the priority of the traffic may also influence the prediction. Consequently we propose four traffic features that enumerate the directional and prioritized traffic counts ($\{$same, opposite$\}$ and higher priority, subscript α, or $\{$same, opposite$\}$ and lower/equal priority, subscript β, where the priority is relative to the train being predicted). It should be noted that the traffic considered in these counts, and in the more granular segmented traffic features, is based on all trains on the network (e.g., including local trains and recrewed trains).

We further examine the potential that the precise location of the traffic may improve the prediction accuracy of the models by considering a higher dimensional representation of the traffic. In the most basic treatment, we can treat each of l track segments between the train and the destination as an element in the feature vector, which is zero if no train is present on the segment or 1 if a train is present. Consequently the dimension of the track occupancy feature is equal to the number of track segments that are considered. For example if no trains are present between the present train and the destination, a vector of zeros of length l (one dimension per segment) would capture the traffic state.

In addition to the track segments between a train origin point and the destination, we consider the track segments in the area around the origin (h track segments) and around the destination (p track segments) that are not part of the origin-destination route segments (l track segments, as stated earlier). This results in the segments from origin to destination being indexed 1 through l, segments around the origin indexed -1 to $-h$, and segments around the destination indexed $l+1$ to $l+p$. The area considered around the origin and around the destination is limited to a distance of 50% of the origin-destination route length, which determines the quantities h and p. On our study route

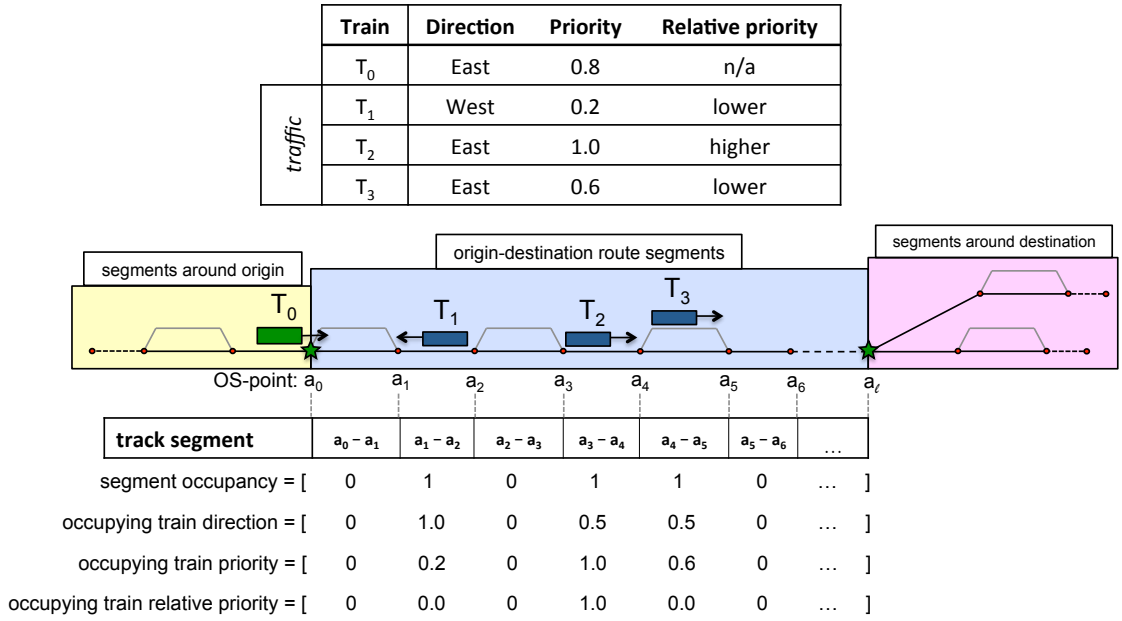


Figure 3.7: Segment-wise features are calculated for the area around an origin point, on the origin-destination route, and around the destination. Each is segmented by OS-points, a_0 through a_l on the origin-destination route in this case. The occupancy feature is first constructed, followed by a vector corresponding to the properties of the occupying train when it is present.

of approximately 140 miles, we consider track segments beyond the destination within 70 miles and track segments around the origin within 70 miles, each excluding the track segments on the 140-mile origin-destination route.

Segment occupancy can be encoded as a vector, each element of which is non-zero when another train is present in a track segment at the time that a prediction is made for a train at the origin point. Likewise, trains that are present on these segments can be described with respect to their relevant properties, namely direction of travel, priority, and relative priority. Each track segment vectors composed of elements that are strictly zero for unoccupied segments and positive for segments occupied by other trains. This process of describing the traffic state via network segments and traffic properties is illustrated in Figure 3.7, for origin-destination route, only. Predictions are made at the origin OS-point a_0 and OS-points delineating segments are labeled a_0 through a_l. An example traffic scenario for the moment at which a prediction is made for train T_0 at the origin is shown with trains T_1, T_2, T_3. The relevant features (i.e., direction and priority) of each train are listed in the figure table. The features for trains T_1, T_2, T_3 are mapped to the track segments corresponding to the location of each train resulting in four feature vectors for segment occupancy, occupying train direction, occupying train priority, and occupying train relative priority.

Chapter 4

Model implementation and evaluation

This chapter describes a set of experiments and a metric to assess the machine learning framework described above. The feature sets used in the models are composed of the features described previously in Section 3.5.

4.1 Description of single origin-destination models

Numerical experiments are performed with concentration on a single route, shown by the dashed area in Figure 3.1, in the Nashville division of the CSX Transportation network. The network contains a mix of single and double track segments, highly heterogeneous traffic, and high volume relative to capacity. Over 50 distinct trains can be seen on the route in a day and more than 20 of these will typically traverse the full route. The route represents one of the most challenging segments on which to estimate ETAs within the CSX Transportation network. Without loss of generality of the methods, the present analysis is restricted to common train types with sufficient trips in the two year dataset and includes the automotive, merchandise, and intermodal trains. These train types have differing priorities, and consequently have distinct behaviors in meet/pass movements and when delays occur. The dataset for trains running the full study route in the correct direction of travel initially contains over 10,000 trips. When the dataset is filtered by train type, recrewed trains are removed, local trains and trains with intermediate work are eliminated, and data errors and incomplete records are removed, there are still approximately 4,200 trips.

The selected route is composed of 35 points along the 140 mile route for which an ETA to the destination must be produced. For each of the 35 ETA problems, a total of five models are implemented and compared. The models include the baseline median predictor algorithm as well as four SVR-based algorithms. Many combinations of algorithm type and feature set have been explored, and the presented models are representative of the model type and performance. For example, the various priority features have each been evaluated for predictive performance by performing single-feature model experiments and $\rho_5(i)$ is found to be the most informative.

The exact model configurations are as follows:

- Model 0: baseline median predictor where $f(x(i)) = \text{median}\{y(i) \mid y(i) \in Y_{tr}\}$
- Model 1: linear SVR with all scalar features (length, tonnage, hp/ton, priority, crew time, ODTOD, traffic counts, and sidings fit); the feature vector is constructed as: $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i)]$, and where $x(i) \in \mathbb{R}^{14}$.
- Model 2: linear SVR with all scalar features plus track segment occupancy vector $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i), O_1(i), \cdots, O_l(i)]$, and where $x(i) \in \mathbb{R}^{14+l}$.
- Model 3: linear SVR with all scalar features and all track segment traffic vector quantities (occupancy, direction, priority, and relative priority on origin-destination route; occupancy around origin point; occupancy around destination point) $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i), O_1(i), \cdots, O_l(i), D_1(i), \cdots, D_l(i), Q_1(i), \cdots, Q_l(i), R_1(i), \cdots, R_l(i), G_{-1}(i), \cdots, G_{-h}(i), E_{l+1}(i), \cdots, E_{l+p}(i)]$, and where $x(i) \in \mathbb{R}^{14+4l+h+p}$.
- Model 4: RBF kernel SVR with all scalar features and all track segment traffic vector quantities $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i), O_1(i), \cdots, O_l(i), D_1(i), \cdots, D_l(i), Q_1(i), \cdots, Q_l(i), R_1(i), \cdots, R_l(i), G_{-1}(i), \cdots, G_{-h}(i), E_{l+1}(i), \cdots, E_{l+p}(i)]$, and where $x(i) \in \mathbb{R}^{14+4l+h+p}$.

4.2 Description of unified all-origin model

The unified all-origin model uses the same feature set as that used by the individual origin-destination models, described in Tables 3.2 and 3.3. The feature set is therefore a mix of categorical, binary, and continuous quantities. Data records for trains at all locations on the network segment with respect to a single destination are used in the unified model dataset. The selected prediction origin location is included as a one-hot encoding of the route locations.

The resulting feature space has 184 dimensions and the number of labeled data records is over 170,000, each of which represents a feature vector captured at a timing point for the train and the corresponding runtime label. The training and testing data is min-max normalized before being used in the models.

4.3 Model evaluation

The error metric used to evaluate a given model and to select model hyper-parameters is *mean absolute error* (MAE), defined as:

$$MAE = \frac{1}{m_{te}} \sum_{i=1}^{m_{te}} |f(x(i)) - y(i)|, \tag{4.1}$$

where $f(x(i))$ and $y(i)$ correspond to the predicted runtime and true runtime of train i, respectively, and m_{te} denotes the number of records in the testing dataset. It follows that numerically low MAE *scores* are better than high scores. Under MAE, all prediction errors are treated equally, regardless of the corresponding true runtime. Performance of each model is compared to that of the historical median predictor, Model 0. The improvement for each model is given as the reduction in the MAE relative to the historical median predictor.

The neural network models were implemented using Keras (Chollet et al., 2015) with TensorFlow (Abadi et al., 2015) backend. Support vector regression, random forest, and statistical models were built using Scikit-Learn (Pedregosa et al., 2011). All models were tested on a computer with 16-core 3.4 GHz processor, 64 GB of RAM, and Nvidia GTX 1080 GPU. Note that neural network models were run on the GPU, while other models were run on the CPU. Neither the preprocessing nor the model testing are parallelized, but could easily be implemented as such on a larger scale due to dataset and model independence.

The data processing steps are completed once for each origin-destination pair and the feature set is stored in a database, which takes approximately 5 minutes per dataset. Building the feature set is the most time consuming step in the process, due primarily to the size of the database of raw data. Adding more features or new data does not require reprocessing of previous data. Model experiments are performed by loading the feature set, selecting the desired data, normalizing features, and training the model and testing the performance via cross-validation. Model training is accomplished in approximately 1 minute for the 4,200 trips, with prediction on test data taking 0.005 seconds per prediction. If implemented network-wide, the computation requirements scale linearly with the number of origin-destination ETA predictions due to trivial parallelization, and could be handled on any modern distributed computing platform.

Chapter 5

Results for individual origin-destination models

This chapter first presents the process for choosing hyper-parameters C and ε in (2.3) for a single origin-destination model with scalar features. We then analyze the series of linear models trained with scalar features on the full 35-OS-point route and the differences between them, specifically with respect to feature weights. Performance results are then shown for each model in Section 4.1 across the route, which demonstrate the impact of feature richness and the nonlinear kernel.

5.1 Choosing hyper parameters for a single model

For each origin-destination model, the SVR parameters, C and ε are chosen as a result of the training and testing process. This is performed using a dataset containing approximately 4200 trips using a five-fold cross validation with an 80/20 training/testing data split. On each fold of the cross validation, the parameter space is explored using a grid search that explores all combinations of parameters within the bounds of each. The results are aggregated across the five folds and the optimal parameter combination is chosen based on the mean minimum testing error. The trained model must be checked for suitability such that it generalizes well to testing data. This check is done using validation curves for C and ε and a learning curve for the amount of data used to train the model.

The interpretation of these parameters as well as analysis of training and testing behavior kept the search space limited. The ε parameter is directly related to residual values between $f(x(i))$ and $y(i)$ and, therefore, can be limited to a search space proportional to the normalized spread of the true outputs $y(i)$. The C parameter penalizes the model training error, summed across all observations $x(i)$, relative to the model flatness, given by the two norm of feature

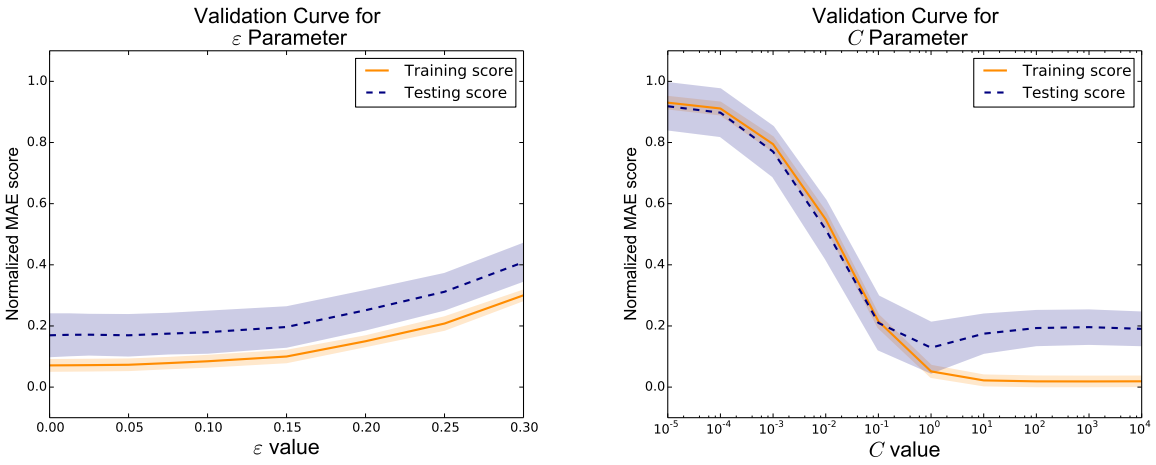
weights. We normalize C such that it is scaled by the number of features and inversely scaled by the number of observations in the training dataset. This maintains the impact of the parameter across models with different feature dimensions and data dimensions.

A validation curve explores training and testing scores across a range of a model parameter, with other parameters fixed. Using a fixed C value of 1, the validation curve for the ε parameter in Figure 5.1a shows relatively little effect of ε value on training and testing scores. Mean training scores are plotted in the solid orange line, with the narrow shading showing a single standard deviation above and below the mean of training scores in cross-validation. The mean testing scores and distribution of scores are plotted with the dashed purple line and shading. Within the acceptable parameter range, high values nor low values make an appreciable effect on testing MAE. Approaching $\varepsilon = 0$, the ε-insensitive loss disappears and the algorithm converges to linear regression. When ε is set too high, the minimization of prediction errors focuses only on observations with exceptionally high residual prediction values. In comparison, the value of C has significantly higher impact on model score. The validation curve is also shown in Figure 5.1b with ε fixed at 0.1 and plotted with common normalized MAE score for comparison. Small values of C emphasize model flatness, but result in poor training and testing scores because model complexity is low. Large values of C achieve low training MAE but generalize poorly to the testing data because of overfitting.

Validation curves show sensitivity for individual parameters. The optimal parameters are chosen simultaneously by evaluating the model on the grid space of all parameter combinations ($C \in [10^{-5}, 10^{4}], \varepsilon \in [0, 0.3]$). For the origin-destination model at grade crossings closest to OS-point #1, the optimal values that minimize MAE are found to be $C = 0.75$ and $\varepsilon = 0.05$. Under these parameters, the difference between the training and testing scores is less than 3% of the training error, which indicates that the model is not overfitting the training data.

The learning curve for a model shows the convergence of training and testing performance by increasing the amount of data available to build the model. The curve is analyzed after hyper-parameters have been chosen. The learning curve shows the MAE score against the number of training examples available to the model. With smaller amounts of data available, training scores will be improved, but at the expense of the model generalizing poorly to testing data. The learning curve in Figure 5.2 indicates that the model is trained on a sufficient amount of data because the curves converge before the full training dataset is used. The mean training and testing scores are denoted by the lines with the shaded areas showing one standard deviation in each direction between cross-validation folds. The model training and testing occur at five equally spaced divisions of available data between 10% and 100%, inclusive (i.e., 10%, 32.5%, 55%, 77.5% and 100%). The disparity in training and testing scores begins at over 40% of the training score when using only 10% of the available data, and decreases to less than 3% when 100% of the available data is used.

For any SVR model with a linear kernel, the feature weights can be interpreted meaningfully in both magnitude and sign. The feature weights are recorded at all cross validation folds and for each origin-destination model to



(a) (b)

Figure 5.1: The validation curves for the ε and C parameters on a single origin-destination model are plotted with a common MAE score axis, which is min-max normalized across both parameters. The sensitivity of the model to ε is relatively low compared to that of C.

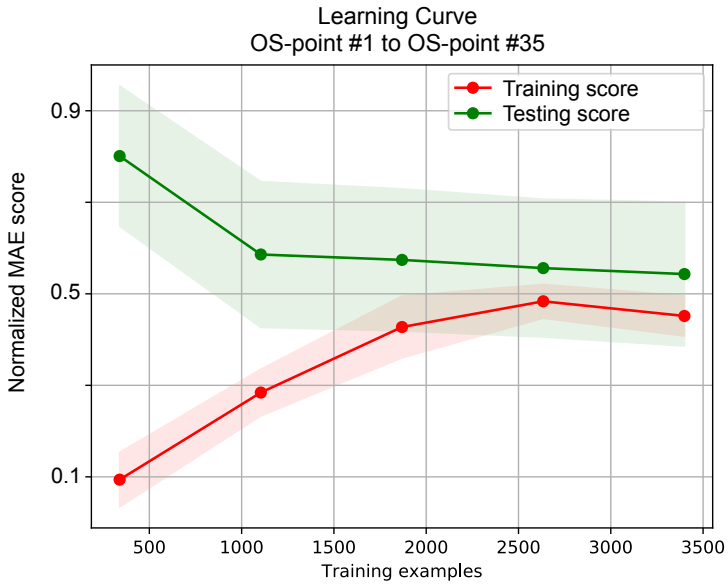


Figure 5.2: The learning curve for a single origin-destination model shows convergence of the training and testing error scores given increasing availability of observations in the full dataset. The MAE score values are min-max normalized.

Table 5.1: Average feature weights for 5-fold cross validation on origin-destination prediction and Pearson correlation coefficient between feature and runtime, calculated at OS-point #1 (Nashville). Exact model output weights are normalized by absolute sum. Features are ordered by highest mean absolute feature weight.

Feature	**Mean absolute feature weight**	**Pearson correlation coefficient**
Priority, ρ_5	0.346	0.294
Crew time remaining, γ	0.137	-0.124
Tonnage, μ	0.110	0.290
Traffic opposite direction lower/equal priority, $\tau_{\psi,\beta}$	0.089	0.226
Available sidings, π	0.067	0.007
Total traffic, τ	0.055	0.135
Traffic opposite direction, τ_{ψ}	0.050	0.085
Length, λ	0.047	0.014
Traffic same direction, τ_{ω}	0.040	0.121
Horsepower per ton, η	0.019	-0.148
Traffic opposite direction higher priority, $\tau_{\psi,\alpha}$	0.011	-0.114
Traffic same direction higher priority, $\tau_{\omega,\alpha}$	0.010	0.140
Traffic same direction lower/equal priority, $\tau_{\omega,\beta}$	0.009	0.275
On duty time to departure, θ	0.008	0.066

assess the relative impact of each feature. The feature weights are normalized by absolute sum within each model, such that $\sum_{j=1}^{n} |w_j| = 1$, to allow comparison between models (recall that the dimension of some traffic features depends on the number of segments between the origin and destination) and are reported in absolute terms for ranking in terms of absolute impact. The feature weight rankings for the origin-destination model at grade crossings closest to OS-point #1 (the beginning of the route in Nashville) are shown in Table 5.1. For this model, the effect of train priority is the dominant feature; this is supported by the distinct runtimes between priority classes at this distance from the destination. Crew time remaining has a large impact because it can affect the runtime of a train at this distance if the train experienced significant delay leaving its last terminal. Tonnage also play a large role due to the lower overall performance of the train during acceleration and deceleration. Features with particularly low impact include traffic counts separated by direction and priority, which is an overly simplistic view of the traffic state on long routes. Horsepower per ton also has less of an impact because the train power is typically sized in this region (which contains significant grades) to avoid delays due to under powered trains.

5.2 Model training across route

The hyper-parameter selection process and model evaluation is replicated for origin-destination predictions made at each of the 35 OS-points on the full route. Because the C parameter is normalized by the training data size and

feature dimension, and the ε parameter demonstrated little sensitivity, the optimal set of hyper-parameters found by the selection process varied little across the route.

The main finding is that feature weights show significant variability across the route. This supports the idea that dispatching techniques have fundamental differences based on relative location of the train to a terminal point and that unique origin-destination models capture some of this nuance. Noting the important result that the feature weights w learned from SVR are globally optimal and unique (Burges and Crisp, 2000), any change in feature weights from the optimal origin-destination specific weights will result in a loss of accuracy of the predictor.

All scalar feature weights are shown at each prediction point in Figure 5.3. The mean feature weight resulting from cross-validation at each location is represented by the lines and min-max ranges for each are shown by the shaded area around the lines. In prediction of the full route, at OS-point #1, the factor most heavily impacting train runtime is priority. Other factors certainly play into the dispatching decisions made for the route, but do not appear to have strong relationships far from the destination. Closer to the destination, traffic counts and train tonnage are driving factors due to decisions around the yard and a natural choke point in the network. The changing importance of train characteristics supports the domain expert knowledge that many factors contribute to train ETA and the impact of these factors is not constant. Along the route, some feature weights experience sharp changes which are due to distinct characteristics of the route. For example, one can observe a sharp dip at OS-point #27 in the weights corresponding to priority and to crew time remaining, along with the sharp increase in the weight corresponding to traffic in the same direction. This OS-point is located on the most significant hill on the route. At this location, a separate helper locomotive attaches to and assists some trains in climbing the hill. Availability of this locomotive is a driving factor in runtime from this location and its presence is captured indirectly in the feature set by the number of trains in the same direction without being explicitly defined as its own feature. The separation of the predictors by each origin-destination allows the estimator to implicitly encode unique attributes of the network which would otherwise require extensive feature engineering.

5.3 Performance comparison of SVR models

In this section, the performance of each model detailed in Section 4.1 is evaluated across all OS-points on the route. The four non-baseline models demonstrate increasing levels of model complexity based on the richness of features used in each model.

The model results across the full route are shown in Figure 5.4 in terms of relative reduction in MAE over the historical median predictor (Model 0). A features set with only scalar features (Model 1), as explored in the choice of hyper-parameters and examination of feature weights in Sections 5.1 and 5.2, represents the largest incremental performance gain for every origin-destination predictor. Inclusion of the track segment feature series (Model 2) and

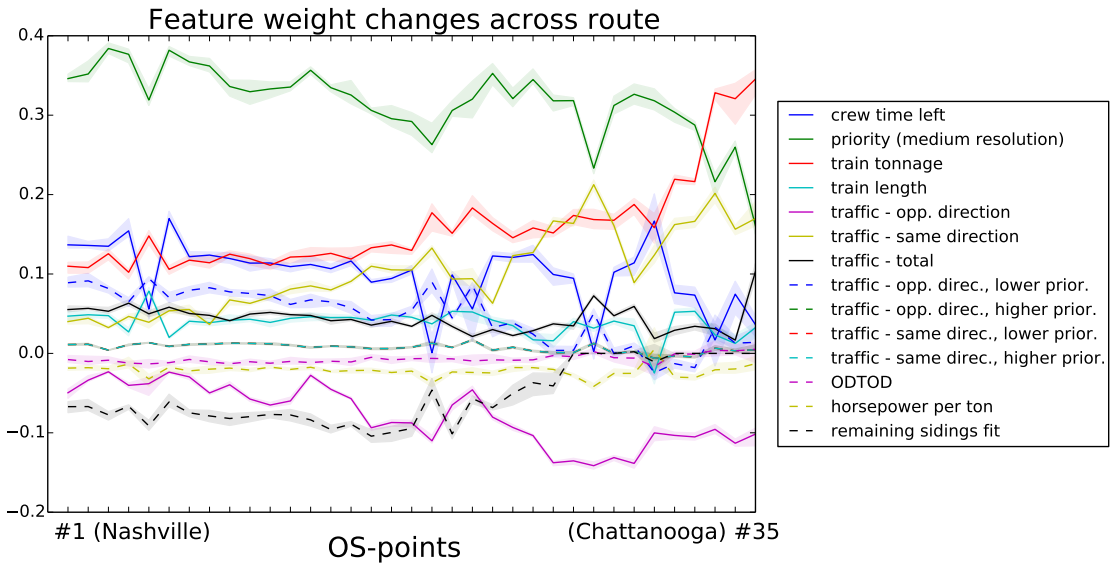


Figure 5.3: Feature weight change of scalar features in origin-destination SVR models across the route, Nashville (1) to Chattanooga (35). Feature importance is measured by absolute magnitude.

inclusion of all track segment feature series (Model 3) each attain small MAE improvements in addition to improvements gained by the scalar features. The RBF kernel, however, does not provide any substantive performance gain over the fully featured linear model. The γ parameter in the RBF kernel is chosen by exploring the range $\gamma \in [10^{-4}, 10^{3}]$ in a grid search alongside the ε and C parameters.

The scalar feature set (Model 1) contains information representing basic relationships and understandings about how the rail network functions. Unsurprisingly, groups of trains such as those that are heavy and low priority generally run slower than the lighter high priority trains. A few counts indicating the amount of traffic on the route will be roughly indicative of the total amount of delay due to meets and passes. These types of relationships can be determined by a linear model given the information in the scalar feature set. This information would be crucial to any ETA prediction, but it does not provide a complete picture.

Based on the relative performance improvements of the SVR models over the baseline, it is evident that the addition of the network traffic state features (Models 2 and 3) show clear advantages by providing high-resolution information compared to the simple counts of network traffic (Model 1). The larger gains are achieved by the track segment occupancy feature series, but the additional information on the direction and priority of the traffic improve the model performance by better informing on the likely meets and passes that will occur. For instance, the mere presence of another train on the route will be somewhat likely to increase runtime, but if this train is traveling in the direction opposing trains on the origin-destination route and has high priority then it may be highly likely to increase runtime. Moreover, the predictive capability of each type of traffic scenario varies. Generally, ETA error is lower for trains with few potential conflicts than it is for trains with higher numbers of potential conflicts. Additionally, ETA error is lower

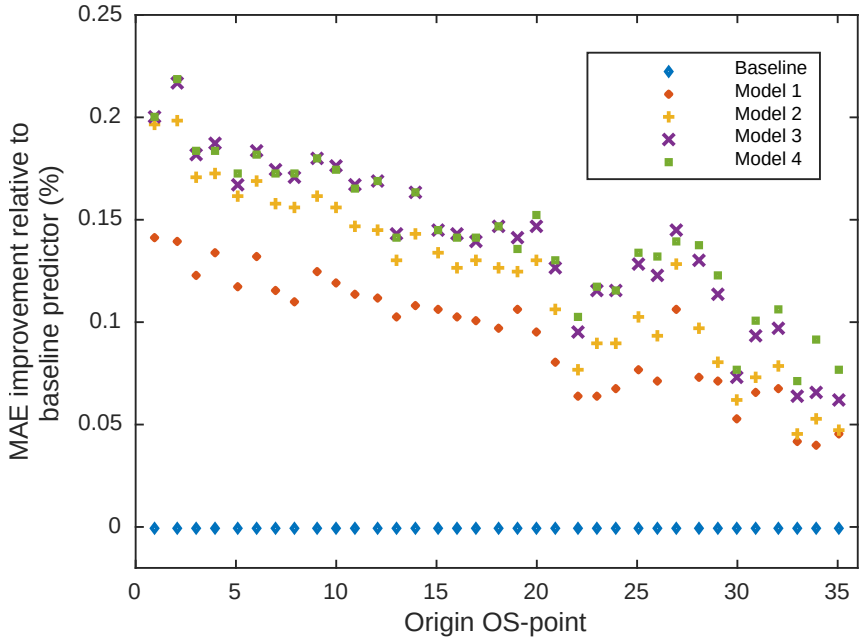


Figure 5.4: Improvement in MAE over baseline historical median predictor for each model at all OS-points between Nashville (1) and Chattanooga (35).

for trains with large number of conflicting *low* priority trains than it is for trains with large numbers of conflicting *high* priority trains.

Model performance varies somewhat across the origin OS-points due to the distribution of route delay, which is not uniform due to the locations of sidings and the likelihood of each to be used. Overall, the relative performance of the models with respect to each other is consistent across the route. Prediction performance relative to the baseline decreases closer to the destination. We expect this is due to the more unpredictable nature of operations close to rail yards. The factors that affect the exact arrival of a train when it gets close are not necessarily present in the data (e.g., ability of the yard to accept more trains, availability of the next train crew). The magnitude of mean average error for the SVR models follows the same decreasing trend.

These route results are summarized in Table 5.2 in terms of mean, maximum, and minimum percent improvement over the baseline. The minimum improvement values are consistently observed for models close to the destination point and the maximum improvement is consistently observed near the beginning of the route.

Table 5.2: Comparison of SVR model performance to baseline predictor, summarized for the 35 OS-points on the full route.

Predictor	**Mean % improvement**	**Maximum %**	**Minimum %**
Model 0 (Baseline)	–	–	–
Model 1	9.4%	14.2%	4.0%
Model 2	12.2%	19.9%	4.6%
Model 3	14.0%	21.6%	6.2%
Model 4	14.3%	21.8%	7.0%

Chapter 6

Results for unified all-origin model

In this chapter we explain the tuning process and results for the unified all-origin model and discuss performance.

6.1 Choosing hyper parameters for a single model

The SVR model was tuned using an exponential grid space for the hyperparameters C and ε. Kernel hyperparameters (γ for RBF kernel and degree for polynomial kernel) were also tuned in the same grid space. Optimal values were selected from all combinations of hyperparameter values in the grid space and found to be $C = 10.$ and $\varepsilon = 0.075$ for all SVR models. The random forest regression model was tuned by exploring a grid space that included the hyperparameters: number of estimators, maximum features considered in split, and minimum samples required for node split. Values explored in the grid space were chosen based on the dimensionality and characteristics of the data and hyperparameter values were chosen to achieve high predictive performance and minimize overfitting.

The deep neural network model architecture is the results of extensive tuning both in the configuration of hidden layers (from three to ten hidden layers were tested), activation function (ReLU and tanh were tested) and optimization function (Adam and SGD were tested) used in the model. Ultimately, eight hidden layers are used with 200, 200, 150, 100, 70, 40, 20, and 10 nodes in each, respectively. The rectified linear unit (Nair and Hinton, 2010) is chosen for the activation function of neurons; the Adam optimizer (Kingma and Ba, 2014) is found to perform best for training the neural network. Early stopping criteria are employed to avoid overfitting. The learning curve for the resulting Adam-ReLU model is shown in Figure 6.1 and compared to the same architecture using stochastic gradient descent optimization. The Adam optimizer not only converges faster (in 32 epochs), but also shows far less variability in validation loss on the way to convergence.

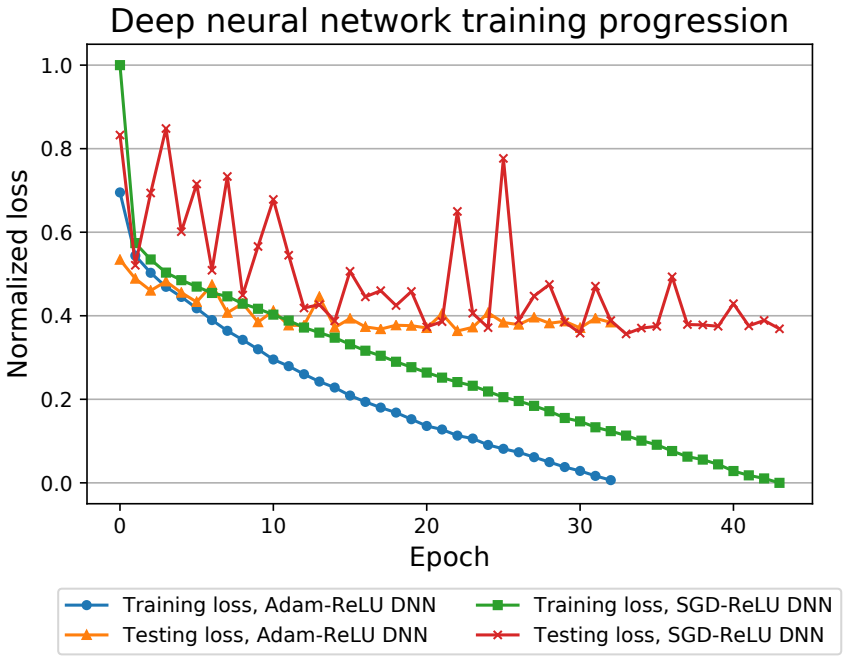


Figure 6.1: Learning curve for deep neural network showing normalized training and testing loss values.

6.2 Prediction results across route

The predictions resulting from the testing data samples are grouped by the station to which they correspond. Each of these samples serves as an ETA prediction made for a particular train at that station, traveling towards the destination following station 35. The results for all six models are shown in Figure 6.2, in terms of percent improvement in MAE compared to the baseline statistical predictor. The results across the route are averaged and shown in Table 6.1. Model training times were also monitored and are shown in Table 6.2. SVR models are constrained to single-threaded computation in this implementation. Conversely, random forest model can be trained in parallel across CPU cores and the DNN model can use the GPU for computation.

6.3 Performance discussion of unified model

There is a noticeable grouping of the SVR models and DNN that maintains over 20% improvement over baseline at stations far from the destination and decreases in relative performance approaching the destination. Linear SVR conspicuously drops below the baseline at the three stations closest to the destination. Meanwhile, the random forest model outperforms all other models at nearly every station. Its performance varies more widely across the route, but achieves improvements exceeding 60% relative to the baseline at stations far from the destination. Predictions at this point are of particular interest for the railroad because of the difficulty of prediction and the increased decision-making

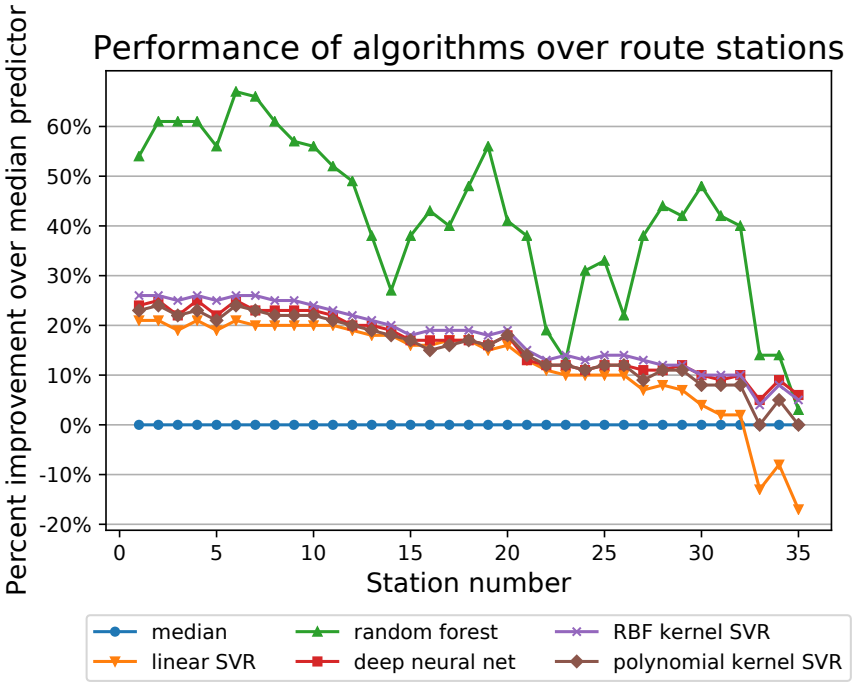


Figure 6.2: Relative improvement of arrival time estimates at each station.

Table 6.1: Summary of model performance over full route

Model	**Mean % Improvement**	**Maximum % Improvement**
Median	0.0%	0.0%
Linear SVR	12.2%	21.0%
Polynomial kernel SVR	15.3%	23.6%
RBF kernel SVR	17.6%	26.4%
Random Forest	42.1%	67.1%
Deep Neural Net	16.3%	24.9%

Table 6.2: Mean model training time.

Model	Mean training time (seconds)
Median	0.1
Linear SVR	20
Polynomial kernel SVR	11360
RBF kernel SVR	5560
Random Forest[1]	25
Deep Neural Net[2]	250

potential for hours in the future. The random forest model also see frequent prediction improvements over 50% and average 42% improvement over baseline across the route.

As predictions are made closer to the destination, the mean runtime and expected mean average error (in absolute terms) decrease. But mean average error relative to baseline also decreases as predictions were made closer to the destination. This is likely due to the fact that runtimes are also less variable close to a train's destination and the factors that drive the residual variability are difficult to quantify with available data. For example trains can be held outside of the yard due to personnel constraints or space constraints such as lack of availability of a specific track needed (e.g., for refueling or classification). We hope to construct features that quantify this destination yard state in future work.

Fluctuating performance of all models, but particularly the random forest model is notable. This can be explained in part by the nonlinear dynamics of the route. Train and route features differ in their predictive impact by location (Barbour et al., 2018). For example, route topography plays a role in the predictive impact of train length and tonnage. At locations with a significant hill on the route, long and heavy trains will have a statistically higher runtime than others; but after the hill is traversed, the statistical difference in remaining runtime will diminish. We see a performance variation at approximately the route midpoint that is likely due to a mountain that must be traversed producing an effect of this sort. However, the dramatic performance variability of the random forest model is likely caused by the nuanced relationships that tree-based regressors can extract from categorical and binary data such as the network state used in this work. It is possible that predictions made by the random forest model at some of the low-performing stations depend highly on additional variables not present in the feature space, such as availability of helper locomotives that supplement train power when traversing hills that are present on the route. The training error of the random forest model is up to 10% lower than the testing error (in absolute terms), but the testing results are consistent through cross validation.

Chapter 7

Prediction of arrival times at grade crossings

In this chapter, we explore the problem of ETA prediction to grade crossings. Specifically, we discuss the limitations and difficulties, modeling choice, and results.

7.1 Prediction limitations

One of the principal difficulties of making arrival predictions at grade crossings is their frequency relative to the sparsely-distributed timing points on the network (OS-points). This raises the notable limitation that we do not have ground truth arrival times at these locations. Interpolating expected arrival times at grade crossings for use in model evaluation would only increase error and uncertainty. Consequently, we can only judge the validity of ETA's made at the upstream and downstream OS-points. The best arrival time prediction at the nearest OS-point would need to be made and correcting based on the precise location of the grade crossing relative to the OS-point. Additional factors could be relevant in calculating the correction factor, such as the last time the train encountered a meet or pass maneuver, which could alter its speed and acceleration state. In the scope of this work, we assess predictions made only to a grade crossing's nearest OS-point because of the absence of ground truth arrival data at grade crossings. Despite this limitation, model performance is still indicative of that expected precisely at grade crossings, due to the magnitude of distance between OS-point and grade crossing being on the order of a single mile.

This limitation can be addressed using additional data sources that railroads collects that are not available for research. We believe this is the correct scientific approach that provides the community realistic estimates of the achievable accuracy the first study of its kind in the United States.

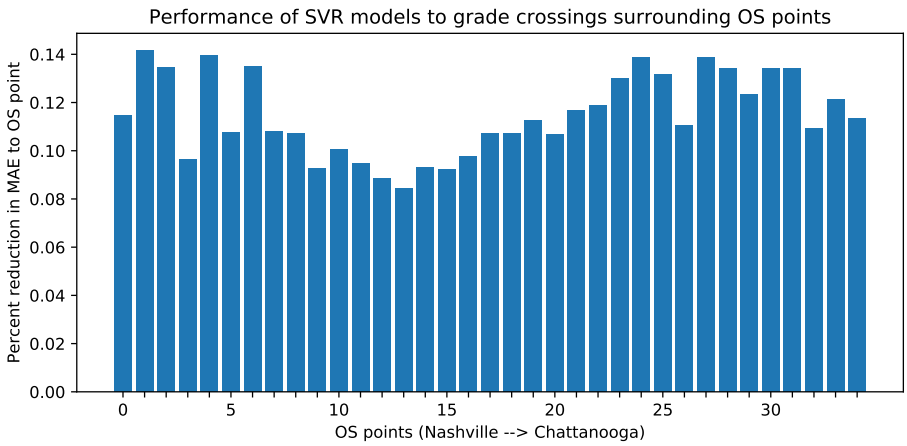


Figure 7.1: Mean improvement percentage over baseline statistical model for predictions made to grade crossings associated with each OS-point.

7.2 Model choice and construction

In order to predict arrival times at grade crossings, we use the individual origin-destination modelling scheme. The origin-destination modelling results supported the notion of location-specific models with respect to the origin point. It follows that this location-specificity would be increasingly important when building models with varying origins and varying destinations (grade crossings). We construct models for each two-OS-point pair on the Nashville-Chattanooga subdivision as a predictor of the ETA to grade crossings associated with that OS-point. We consider trains traveling in only the direction from Nashville to Chattanooga and place a lower bound limit on the distance between OS-points due to known variability factors of train runtimes between nearby OS-points for which we can not quantify with data inside the scope of this work.

The mean improvement in MAE relative to the baseline predictor is shown in Figure 7.1 for predictions made to each OS-point aggregated across other OS-points on the network serving as origin points.

7.3 Performance discussion of grade crossing models

Model performance shows notable variations for predictions made between OS-points as a best proxy for nearby grade crossings. However, the values for improvement in MAE are very comparable to those observed in individual single-destination models across the same track territory. A maximum mean improvement in MAE of 14% was observed at multiple destination OS-points.

7.4 Practical model implementation

As previously discussed in Sections 1.3, 3.4, and 7.1, there are limitations in the present work inherent to the accessible data sources. These would need to be addressed as a necessary condition for implementing real-time grade crossing prediction models, but are entirely surmountable. Making simultaneous predictions for thousands of trains at thousands of grade crossings would require a large scale computing environment, but this is a problem well within the realm of technological feasibility for modern distributed computing. As noted in Section 4.3, model training required a significant amount of time up front and would need to be performed periodically to keep models up to date. But training may be performed offline so as to not interfere with real-time online prediction. Each prediction on test data in this work took on average 0.005 seconds. This is reasonable for a real-time ETA prediction system.

Though the analysis in this work focuses on predictions made on single-track network segments, the same prediction framework is immediately extensible to any network configuration. Models could be constructed of the same form because the nuance introduced by some geographic factors is captured inherently in historical data. We also expect that this approach is applicable for all freight railroads in the United States and, potentially, elsewhere in the world.

Implementation of this work fits into the IEEE Standard 1570-2002 for highway-rail intersection interface (discussed in Section 1.1) as a long-horizon train detection system that informs traffic management systems. Ouyang et al. (2018) addressed strategic incidence response planning in the presence of correlated disruptions. Specifically, where assets and resources should be positioned in anticipation of potential accidents when there is a risk of systematic disruption to the transportation network. They note, in particular, trains blocking railroad crossings as a common example. The joint consideration of our work on real-time prediction with strategic planning could enable robust emergency response operations under the inevitable disruptions.

Chapter 8

Conclusion

This work presents a data-driven approach to predict ETAs at grade crossings on freight rail networks on long time horizons necessary for early warning applications and emergency management systems. The ETA generation problem is posed as 1) a series of independent origin-destination ETA prediction problems to capture location specificity and avoid bias in the training data of a single general model due to time varying features and 2) a unified all-origin prediction model to utilize a greater amount of training data in a single model and leverage more advanced algorithms.

In terms of the origin-destination models, the problem is tractable for sparse rail networks in the United States due to relatively low network complexity that reduces the number of relevant origin-destination pairs. This approach is shown to demonstrate specificity with respect to distinct feature weights (i.e., relative importance) between origin-destination models. Compared to naive prediction based on historical median runtimes, an average improvement of 14% and maximum improvement of over 21% are achieved by the best performing SVR models.

In terms of the unified all-origin modeling scheme, six models including one statistical model, three SVR models, a deep neural network model, and a random forest regression model are implemented. Performance of the models is analyzed at locations across the study area and found to vary, particularly for the random forest model. The random forest model achieves the best performance yet realized on this dataset, with an average 42% improvement in MAE relative to the baseline statistical predictor. The average improvement of the random forest model and the maximum predictive improvements of over 60% are actionable for freight rail operational decision making and potentially useful in grade crossing applications.

Based on these findings, our future research steps include the following. Due to the large variance caused by recrews, we are interested in developing a data-driven classifier to preemptively classify trips that are likely to be recrewed. This step is necessary because ETA estimates produced by models trained on data from non-recrewed trains will not generalize to recrewed trains because the ETA of recrewed trains depends on factors outside the scope of this

work. For the trains that are not likely to be recrewed, further improvements in the ETA accuracy are possible with the construction of additional targeted traffic features constructed for route topography and meet-pass events. Considering externalities such as weather, as well as ancillary operations such as scheduled track work or slow orders, may also increase prediction accuracy. We are also interested in building models on the state of the origin or departure yard, which may create delays that cascade onto the line of road. Further studying the effects of each input feature to the prediction result may provide practical insight into delay causes and mitigation. The comparison of these results to optimization-based simulation as well as the exploration of more sophisticated deep learning approaches are also areas of future exploration.

This work showed improved results for ETA estimation compared to the state of practice, which can be valuable to emergency vehicle scheduling and management around highway-rail grade crossings. In implementation, the real time up-to-the-minute reporting of arrival times at grade crossings will also require the fusion of additional data sources such as real-time GPS data that is collected by the railroads but outside the scope of this work. But more improvements will require model enhancements that incorporate predictive rail traffic evolution and incorporate more of the railroad operational factors that contribute to primary and knock-on delay.

Appendix A

Estimation of individual passenger train delays for ETA prediction at grade crossings

A.1 Introduction

For a given train, the delay is defined as the difference between the true running time and the free running time. The variabilities associated with ancillary train operations (e.g., equipment maintenance, station dwell time, weather) may contribute to the travel time delay, which consequently impacts the overall variability and predictability of railroad operations (Dingler et al., 2010). In the United States, Amtrak passenger trains have priority over freight trains, and yet the average on–time rate of Amtrak is less than 75% (Bureau of Transport Statistics, 2017). In the presence of this variability, it is important to investigate train delays as constituent parts of larger efforts to predict train ETAs at grade crossings and enable proactive safety applications.

The objective of this appendix is to develop new data-driven methodologies to estimate passenger train delays and to assess their performance on a large dataset of more than 100,000 trips. In the past, many analytical models and simulation approaches have been proposed to estimate train delays. While these approaches have merit due to their elegance (analytical approaches) and realism (simulation based approaches), application of either approach constitutes a major model building or calibration task. For complex systems, analytical methods require some degree of abstraction to maintain tractability. Simulation based approaches can model the complexity of the realistic train operations, but require extensive effort to accurately calibrate the model.

With the recent advances in sensing and communication technology, train positioning data is now available to improve train delay estimation through data-driven methods. For example, regression models can be constructed to estimate delay, where the parameters associated with the regression models are calibrated by learning from historical

data. Compared to the analytical methods and simulation methods, data-driven approaches can be easily generalized and deployed to estimate train delays for any train, as long as training data is available. Note this necessarily prevents the applications of these methods for scenario planning, which analytical or simulation approaches are more appropriate.

The main challenge associated with data-driven approaches for passenger train delay estimation is data availability. First, accurate data may not be available, or it may be sparse or incomplete. For example, the Amtrak data considered in this work does not contain records between stations, and no information is publicly available about the freight traffic, which shares the same track. Moreover, the data is incomplete, and some delays are never recorded. In spite of these limitations, this work shows standard regression models can significantly improve passenger train delay estimation compared to the predictions based on the scheduled time table. While additional refinements are certainly warranted, data-driven approaches appear promising for delay estimation. The main contributions of this section are summarized as follows:

- This section proposes two data-driven approaches for passenger train delay estimation. A historical regression model is designed to predict train delays before the current trip starts, and online regression models are proposed to provide a more accurate train delay estimate after the trip begins, using the delay recorded at the upstream station on the current trip and the delay recorded by other nearby trains.
- Data from 282 Amtrak trains (over 100,000 trips), are used to illustrate and test the proposed algorithms. The estimation results show the proposed historical regression model improves the *route mean square error* RMSE by 12% and the online regression model improves the RMSE by 60%, compared to prediction based on the scheduled time table.

The reminder of this Appendix is organized as follows. In Section A.2, data-driven autoregressive approaches are proposed for train delay estimation. The proposed methods are implemented and tested with Amtrak data, and the estimation results of the proposed methods are shown in Section A.3. In Section A.4, we conclude the proposed methods significantly improve delay estimation compared to the scheduled time table, and note the need for further work on capturing knock-on delays and other delay related factors.

A.2 Methodology

In this Section, we develop two approaches to estimate train delays. The first method is a historical regression model developed by assuming delays from one trip to the next follow an vector autoregressive process. This model predicts train delays at each station before the current trip starts based on the delay recorded in the past trips. Next, two variations of an online regression model are developed, which aims at providing accurate train delay estimation by

using delay information of the train at earlier stations long the current trip, as well as delay information of other trains that share the same corridor.

A.2.1 Historical regression model

Passenger train delay can be assumed to follow a vector autoregressive process (Lütkepohl, 2007) , because passenger trains operate on a fixed frequency (e.g., daily) and schedule. As a result, prior delays on previous trips bring information to estimate the train delay at each station for the current trip. The vector autoregressive process predicts train delays at each station along the route simultaneously based on the prior delays on previous trips. The historical regression model constructed by a vector autoregressive process of order p is described as follows:

$$y_t^i = A_1 y_{t-1}^i + \cdots + A_p y_{t-p}^i + \nu + u_t, \tag{A.1}$$

where $y_t^i = \left(y_{1,t}^i, \cdots, y_{k,t}^i, \cdots, y_{K,t}^i\right)^T \in \mathcal{R}^K$ denotes the vector of train delays on trip t for train i. Here, $y_{k,t}^i$ is a scalar that denotes the train delay at station k on trip t for train i, with $k = 1, \cdots, K$, and K is the total number of stations on the train trip. The matrix $A_m \in \mathcal{R}^{K \times K}$, with $m = 1, \cdots, p$, denotes the relationships of delays among the current and past trips, and among stations. The variable $\nu = (\nu_1, \cdots, \nu_K)^T \in \mathcal{R}^K$ is an intercept term which allows constant delays. The variable $u_t = (u_1, \cdots, u_K)^T \in \mathcal{R}^K$ is the *white noise* which denotes the error between the predicted $\hat{y}_t$ and the true y_t, where $\hat{y}_t$ is given as:

$$\hat{y}_t = A_1 y_{t-1} + \cdots + A_p y_{t-p} + \nu. \tag{A.2}$$

Model (A.1) is also called a vector autoregressive process with lag p, since the vector y_t is computed using only the system state in the previous p trips. To apply the model, we first select p, and then train the parameters A_m and ν by using a least squares fit on the historical data. Then, the vector autoregressive process with the trained parameters can be used for prediction.

A.2.2 Online regression model

The historical regression model can predict train delays at each station before the current trip starts. After the trip begins, the accuracy of the train delay estimation at a station can be further improved if delays of the train at its upstream stations are known, and if the delays of another trains that may interact with the current train are known. In this section, an online regression model is proposed to incorporate such information for train delay estimation by

using an autoregressive process (Cook, 1985):

$$y^i_{k,t} = a_1 y^i_{k-1,t} + \cdots + a_p y^i_{k-p,t} + c_k + \Phi^i_{k,t} + u_{k,t}, \tag{A.3}$$

where $y^i_{k,t}$ is a scalar that denotes the delay of train i at station k during trip t. The parameters a_m and c_k denote the relationship of train delays among the current and past stations. The term $u_{k,t}$ in (A.3) is a scalar which denotes the error associated with the model. The predictor $\hat{y}^i_{k,t}$ is given as:

$$\hat{y}^i_{k,t} = a_1 y^i_{k-1,t} + \cdots + a_p y^i_{k-p,t} + c_k + \Phi^i_{k,t}. \tag{A.4}$$

The term $\Phi^i_{k,t}$ denotes the delays of another trains that may contribute to the delay of train i at station k. This term is modeled as:

$$\Phi^i_{k,t} = \Sigma_{\left(j,\tilde{k},\tilde{t}\right)\in\Omega_{i,k,t}} b_j y^j_{\tilde{k},\tilde{t}}, \tag{A.5}$$

where $\Omega_{i,k,t}$ denotes the set of train–station pairs that contribute to $y^i_{k,t}$. The term $y^j_{\tilde{k},\tilde{t}}$ is the delay of train j at station $\tilde{k}$ during trip $\tilde{t}$. Note that station $\tilde{k}$ is not necessarily the same station as k since the delay of train j at other stations $\tilde{k}$ may also influence the delay of train i at station k (e.g., if train i and train j share the same track, but move in opposite directions). Moreover, trip $\tilde{t}$ must be distinguished from t since it is a trip index for train j. The parameter b_j is the factor that indicates how the delay of train j at station $\tilde{k}$ on trip $\tilde{t}$ impacts $y^i_{k,t}$.

The existence of Φ can be interpreted as follows. If two trains are closely scheduled on a single track line and the front train is delayed at a station, then it is possible for the following train to experience knock–on delay. Note that because Amtrak shares track with freight trains, and freight train positioning data is not publicly available, the knock–on delay caused by freight traffic cannot be captured when this model is implemented with Amtrak data only.

We also consider two variations of the online regression model (A.3). The first one is a predictor which is constructed based on the assumption that the delay of train i at station k is simply equal to the delay of the same train at station $k-1$. In this case, the model (A.3) becomes:

$$y^i_{k,t} = y^i_{k-1,t} + u_{k,t}. \tag{A.6}$$

The second variation of regression model does not consider the delays caused by the interactions among trains. The simplified (interaction free) model is given as:

$$y^i_{k,t} = a_1 y^i_{k-1,t} + \cdots + a_p y^i_{k-p,t} + c_k + u_{k,t}, \tag{A.7}$$

As a result, model (A.6) can be viewed as a baseline approach where delay is assumed to propagate from the upstream station to the downstream station. Model (A.7) captures non–constant delay relationship between stations by training the parameters a_m and c_k, while model (A.3) adds another component Φ to incorporate the delay caused by interactions among trains.

A.3 Implementation and results

The proposed methods are tested with Amtrak passenger train data released by AmtrakStatusMaps (2015). The historical regression model (A.2), the online baseline model (A.6), the online regression non–interacting model (A.7), the online regression interacting model (A.3), and the scheduled time table are tested and compared using data from 282 Amtrak trains (or 120 trains in the case of the interacting model (A.3) since the other trains have an empty interaction set $\Omega_{i,k,t}$).

The general procedure to evaluate the models is as follows. First, structural parameters of each regression model must be selected (e.g., the lag p and the interaction set $\Omega_{i,k,t}$). Second, the available data is partitioned into a training dataset and a test dataset. Third, the parameters of each regression model are determined through least squares estimation on the training dataset. Finally, the model is evaluated on the test dataset to determine the accuracy of the predictor.

A.3.1 Data description and training data selection

Amtrak data is used as an example of passenger train data to illustrate and test the proposed method. The dataset contains all Amtrak passenger train arrival and departure data at each station from 2006 to 2013. The dataset is released by AmtrakStatusMaps (2015) and is publicly available. For each train and each trip, the following data are recorded: *station code*, *scheduled arrival day and time*, *scheduled departure day and time*, *actual arrival time*, *actual departure time* and *comments*.

After an exploration of the dataset, it is found that the Amtrak data is coarse and a number of data records are missing. In particular, most stations do not have records for actual train arrival times, and some stations do not have records for scheduled train arrival times. However, nearly all the stations have records for the scheduled departure time and the actual departure time. As a result, the time difference between scheduled departure time and actual departure time is used to denote travel time delay in the following experiments.

A year of delay data of a typical train (train 68 in 2013) is used as an example to visualize the delay (Figure A.1). The train travels daily from Montreal (MTR) to New York, and stops at 18 stations. In Figure A.1, two data patterns can be observed from the recorded delays. First, some stations are more likely to experience delay compared to the

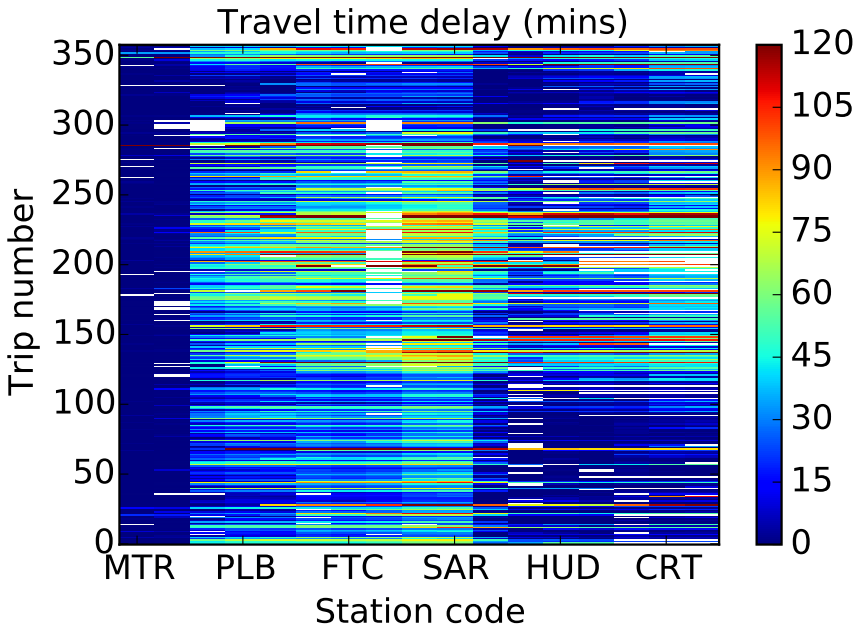


Figure A.1: Delays for Amtrak train 68 in 2013. Six of the 17 stations along the route are labeled. The color in the figure denotes departure delay at each station for each trip. Missing data are shown in white.

others (e.g., Ticonderoga, NY (FTC) compared to Hudson, NY (HUD)). Second, once delay occurs on a trip, the delay is likely to last for several stations. Such data patterns are also commonly observed on other Amtrak trains.

Data from 282 Amtrak trains from 2011 to 2013 are used to train and test the proposed algorithms, which consists of more than 100,000 train trips. For each train, data from 2011 and 2012 are used as training data, while the first 30 trips of 2013 for each train are used as test data. Note that the first 30 trips may occur over one to several months depending on the frequency at which the train operates (e.g., daily, weekly). We also note that *AmtrakStatusMaps* contains data for more than 450 Amtrak trains from 2011 to 2013, however a regression model cannot be constructed for all trains. The vast majority of excluded trains were subject to a route re-configuration (e.g., adding a station) during the three year period, meaning that a complete set of training or test data is not available. A small subset of trains without schedule reconfigurations were also excluded due to a large amount of missing data. These are practical issues that must be addressed before data-driven methods can be widely deployed.

When models (A.2), (A.6) and (A.7) are tested, data from all 282 Amtrak trains are used as training data. The online regression interacting model (A.3) is evaluated on a smaller subset consisting of 120 trains, where the interacting set $\Omega_{i,k,t}$ is non-empty.

A.3.2 Cross validation

In order to test if the results of the proposed methods are sensitive to the training data, a k–fold cross validation (Kohavi, 1995) is used. The training data is partitioned to five sets. Each model is run five times, and for each run, a district set composed of four of the five sets are used to construct the training dataset. Different from the standard k–fold cross validation where the test dataset is also changed during each fold, the algorithm is tested with the data in 2013, to avoid the scenario that the model is trained with data from the future and tested on the past.

A.3.3 Selection of structural regression parameters

We briefly describe how the order p for models (A.3), (A.2), and (A.7) is selected, and how the set $\Omega_{i,k,t}$ is determined when model (A.3) is deployed.

When the historical model (A.2) is implemented, multiple p values have been tested, and it is found that the historical model has the overall best performance when the order p is set as one. Practically, the order p associated with the historical model for each train can also be determined individually by minimizing the final prediction error following the criteria in Lütkepohl (2007). When the online regression models (A.3) and (A.7) are implemented, the order p is also chosen as one. Because once the train delay at the upstream station is known, the delays of the train from the stations further upstream do not contribute to the estimation accuracy. This assumption was also tested by evaluating larger orders p for the model, which caused slight decreases in the predictive accuracy.

When the online regression interacting model (A.3) is implemented, the set $\Omega_{i,k,t}$ for each station is constructed according to the following assumption. If train j is scheduled at a the same or neighboring station $\tilde{k}$ within an hour of train i at station k, the then the delay of train j at station $\tilde{k}$ is considered as part of the regression. As a final step we prune any trains that are scheduled at the same station but do not share the same track, which is common at major terminals such as Chicago's Union Station.

A.3.4 Regression results without interactions among trains

In this section, the historical model (A.2) and the online models (A.6) and (A.7) are trained and tested with the data from the 282 Amtrak trains.

The average RMSE e_i of the proposed models for train i is computed as follows:

$$e_i = \sqrt{\left(\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{K}\sum_{k=1}^{K}u_{n,k,t}^2\right)\right)\right)}, \tag{A.8}$$

where N denotes the total number of cross validations and T denotes the total number of trips to be estimated. The

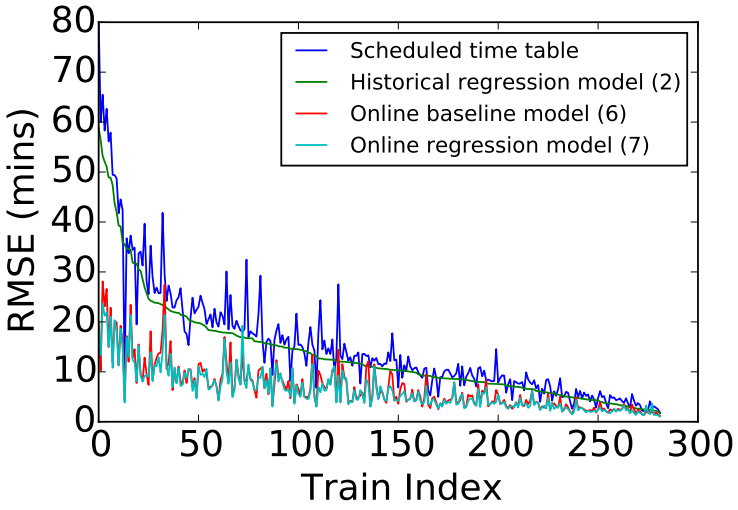


Figure A.2: Ranked average MSE associated with scheduled time table, models (A.2), (A.6), and (A.7) for each train.

Table A.1: Average RMSE of the proposed methods. The improvement shown in the table are percentage improvements of proposed methods compared to the scheduled time table with respect to RMSE

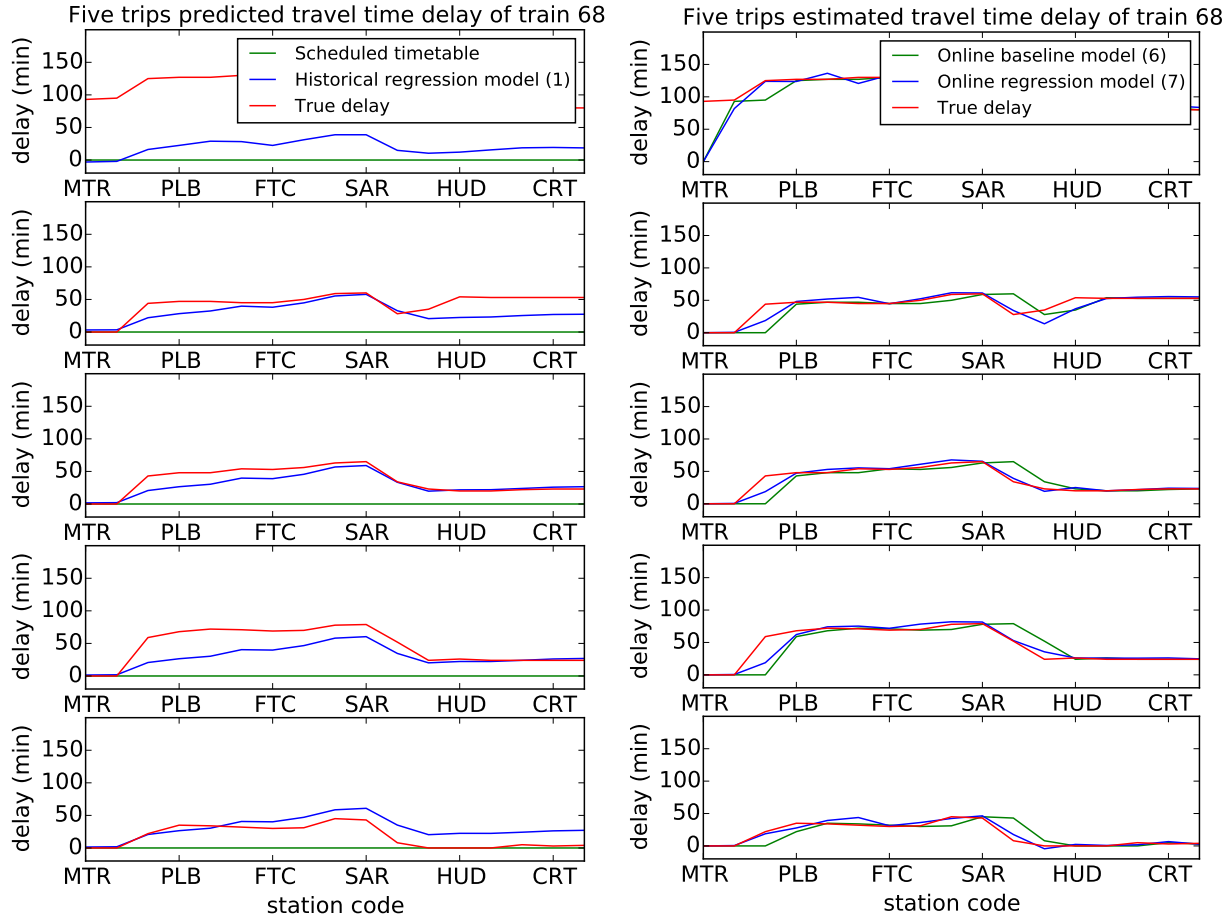
Method	RMSE (min)	Improvement
Scheduled time table	19.4	N/A
Historical regression model	17.0	12%
Online baseline model	8.4	57%
Online regression model	7.7	60%

term $u_{n,k,t}$ is the model error of the t^{th} trip for the n^{th} cross validation for train i. The mean square error is computed and averaged over the T estimated trips, and then averaged over the N cross validations. In this simulation, $T = 30$ and $N = 5$.

The ranked average RMSE of the proposed methods and the scheduled time table for each train are shown in Figure A.2. The average RMSE over all trains for each predictor is summarized in Table A.1. The historical regression model has better estimation accuracy compared to the scheduled time table, since delays from the past trips are incorporated in the model. Both online algorithms perform significantly better than the historical model, because online delay information from the upstream station are used to estimate the delay for the downstream station. Moreover, the online regression model (A.7) performs better than the online baseline model (A.6), because it is able to incorporate the potential delay that may occur between the current station and the next station, by training the parameters a_m and c_k. In summary, compared to the scheduled time table, the historical regression model (A.2) improves the RMSE by 12%, and the online regression model (A.7) improved the RMSE by 60%.

Note it is not possible to compactly display the calibrated model parameters for each train and for each model within the space of this report. In order to provide more details of how the regression models perform, we again use

train 68 as an example. The prediction results by the scheduled time table and historical model (A.2) of the first 5 of the 30 trips for train 68 are shown in Figure A.3a, and the estimation results by online models (A.6) and (A.7) are shown in Figure A.3b. Again, we can conclude the historical model performs better than the scheduled time table, and the online models can further improve the delay estimation accuracy compared to the historical model.



(a) Historical model and time table

(b) Online models

Figure A.3: Five trips delay estimation results of train 68 (top to bottom). The left figure shows the results for the scheduled time table and the historical regression model. The right figure shows the results for the online regression models.

A.3.5 Regression results with interactions among trains

Next, the online regression interacting model (A.3) is tested. The online interacting model (A.3) is compared with the online non–interacting model (A.7) to investigate if modeling the delay caused by interaction among trains may help to improve the estimation accuracy. It is found that the RMSE difference between the two models for most of the trains are less than 2%, and the average RMSE over all trains of the two models are computed as 7.42 and 7.40 min, respectively.

The online regression interacting model (A.3) tends to capture the knock–on delay effect by including the term Φ. However, the performance of these two models are very close. After an investigation on the trained parameters b_j, it is found the values of b_j are nonzero and they do influence the final estimation, however, it does not outperform the online regression model (A.7).

One possible explanation is as follows. Once a train is delayed at a station, it is observed that the delay will propagate for several stations. As a result, it is usually the case that both the front train and the following train are delayed for several consecutive stations on a trip. While it is true that the following train is delayed due to an interaction with the leading train, the online regression model (A.7) is able to capture this knock–on delay by modeling the delay propagation from its upstream station for all stations except the first one, where the delay is initiated. As a result, similar performance is found for the online regression model (A.7) and the online regression interacting model (A.3).

A.4 Summary of passenger train findings

This Appendix studies the passenger train travel time delay problem by using data-driven approaches. A historical regression model is proposed to predict train delays before the current trip starts, and an online regression model with two variations are developed to estimate train delays using delay information from current trips recorded at upstream stations and other related trains. The proposed methods are tested with Amtrak passenger train data. Compared to the prediction based on the scheduled time table, the historical regression model (A.2) improves the RMSE of the delay estimation by 12%, and the online regression model (A.7) improves the RMSE by 60%.

This is the first use of data-driven approaches to study passenger train delays in the United States with the goal of enabling proactive safety at grade crossings. It shows standard regression models can significantly improve the travel time delay estimates compared to the scheduled time table even though data are coarse and limited. The primary shortcoming of the data is the lack of information about freight trains that operate concurrently on the same tracks. This motivates the focus on freight train data to further enhance the accuracy of the estimates, which is found in the main body of this report.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org/.

Altinkaya, M. and Zontul, M. (2013). Urban bus arrival time prediction: A review of computational models. *International Journal of Recent Technology and Engineering (IJRTE)*, 2(4):164–169.

Amtrak (2016). Route on-time performance. Retrieved from https://www.amtrak.com/historical-on-time-performance.

AmtrakStatusMaps (2015). Amtrak status maps. http://dixielandsoftware.net/Amtrak/status/StatusMaps/.

Assad, A. A. (1980). Models for rail transportation. *Transportation Research Part A: General*, 14(3):205–220.

Association of American Railroads (2013). Class I railroad statistics. Retrieved from https://www.aar.org/StatisticsAndPublications/Documents/AAR-Stats-2013-02-07.pdf.

Barbour, W., Martiniz Mori, J. C., Kuppa, S., and Work, D. B. (2018). Prediction of arrival times of freight traffic on US railroads using support vector regression. *Transportation Research Part C: Emerging Technologies (expected)*.

Basak, D., Pal, S., and Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing-Letters and Reviews*, 11(10):203–224.

Bonsra, K. B. and Harbolovic, J. (2012). Estimation of run times in a freight rail transportation network. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, USA.

Brady, T. F. (2003). Public health: emergency management: capability analysis of critical incident response. In *Proceedings of the 35th conference on Winter simulation: driving innovation*, pages 1863–1867, New Orleans, LA, USA.

Breiman, L. (1984). *Classification and regression trees*. Routledge, 1st edition.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Bureau of Transport Statistics (2017). Amtrak on-time performance. Retrieved from https://www.bts.gov/archive/publications/multimodal_transportation_indicators/2015_08/system/amtrak_ontime.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Burges, C. J. and Crisp, D. J. (2000). Uniqueness of the SVM solution. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 223–229, Denver, CO, USA.

Cambridge Systematics (2007). National rail freight infrastructure capacity and investment study. Retrieved from https://codot.gov/programs/transitandrail/resource-materials-new/AARStudy.pdf.

Cats, O. and Loutos, G. (2016). Real-time bus arrival information system: an empirical evaluation. *Journal of Intelligent Transportation Systems*, 20(2):138–151.

Chadwick, S. G., Saat, M. R., and Barkan, C. P. (2012). Analysis of factors affecting train derailments at highway-rail grade crossings. In *Proceedings of the Transportation Research Board 91st Annual Meeting*, Washington, DC, USA.

Chadwick, S. G., Zhou, N., and Saat, M. R. (2014). Highway-rail grade crossing safety challenges for shared operations of high-speed passenger and heavy freight rail in the us. *Safety Science*, 68:128–137.

Chapuis, X. (2017). Arrival time prediction using neural networks. In *Proceedings of RailLille2017: 7th International Conference on Railway Operations Modelling and Analysis*, Lille, France. International Association of Railway Operations Research (IAROR).

Chen, B. and Harker, P. T. (1990). Two moments estimation of the delay on single-track rail lines with scheduled traffic. *Transportation Science*, 24(4):261–275.

Chien, S. I.-J., Ding, Y., and Wei, C. (2002). Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering*, 128(5):429–438.

Chollet, F. et al. (2015). Keras. Retrieved from https://keras.io.

Cook, E. R. (1985). *A time series analysis approach to tree ring standardization*. The University of Arizona, Tucson, AZ, USA.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

D'Ariano, A. and Pranzo, M. (2009). An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9(1):63–84.

D'Ariano, A., Pranzo, M., and Hansen, I. A. (2007). Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222.

Dingler, M., Koenig, A., Sogin, S., and Barkan, C. P. (2010). Determining the causes of train delay. In *AREMA Annual Conference Proceedings*, Orlando, FL, USA.

Dingler, M., Lai, Y.-C., and Barkan, C. P. (2009). Impact of train type heterogeneity on single-track railway capacity. *Transportation Research Record: Journal of the Transportation Research Board*, (2117):41–49.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161, Denver, CO, USA.

Estes, R. M. and Rilett, L. R. (2000). Advanced prediction of train arrival and crossing times at highway-railroad grade crossings. *Transportation Research Record: Journal of the Transportation Research Board*, (1708):68–76.

Federal Railroad Administration (2018). Highway-rail crossing inventory data. Retrieved from https://safetydata.fra.dot.gov/officeofsafety/publicsite/downloaddbf.aspx.

Federal Railroad Administration Office of Safety Analysis (2018). Safety statistics. Retrieved from https://safetydata.fra.dot.gov/OfficeofSafety/default.aspx.

Furtado, F. M. B. A. (2013). US and European freight railways: The differences that matter. *Journal of the Transportation Research Forum*, 52(2):65–84.

Gorman, M. F. (2009). Statistical estimation of railroad congestion delay. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):446–456.

Goverde, R. M. (2005). *Punctuality of railway operations and timetable stability analysis*. Netherlands TRAIL Research School, Delft, Netherlands.

Goverde, R. M. and Meng, L. (2011). Advanced monitoring and management information of railway operations. *Journal of Rail Transport Planning & Management*, 1(2):69–79.

Hallowell, S. F. and Harker, P. T. (1998). Predicting on-time performance in scheduled railroad operations: Methodology and application to train scheduling. *Transportation Research Part A: Policy and Practice*, 32(4):279–295.

Hertenstein, J. H. and Kaplan, R. S. (1991). *Burlington Northern: The ARES decision*. Harvard Business School, Cambridge, MA, USA.

Higgins, A., Ferreira, L., and Kozan, E. (1995). Modelling delay risks associated with train schedules. *Transportation Planning and Technology*, 19(2):89–108.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

IEEE Vehicular Technology Society (2002). Standard for the interface between the rail subsystem and the highway subsystem at a highway rail intersection. *IEEE Standards*, (1570-2002).

Kecman, P. and Goverde, R. M. (2013). An online railway traffic prediction model. In *RailCopenhagen2013: 5th International Conference on Railway Operations Modelling and Analysis*, Copenhagen, Denmark. International Association of Railway Operations Research (IAROR).

Kecman, P. and Goverde, R. M. (2015). Online data-driven adaptive prediction of train event times. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):465–474.

Khadilkar, H., Salsingikar, S., and Sinha, S. K. (2017). A machine learning approach for scheduling railway networks. In *Proceedings of RailLille2017: 7th International Conference on Railway Operations Modelling and Analysis*, Lille, France. International Association of Railway Operations Research (IAROR).

Khoshniyat, F. and Peterson, A. (2017). Improving train service reliability by applying an effective timetable robustness strategy. *Journal of Intelligent Transportation Systems*, 21(6):525–543.

Kim, Z. and Cohn, T. E. (2004). Pseudoreal-time activity detection for railroad grade-crossing safety. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):319–324.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1145.

Kraay, D. R. and Harker, P. T. (1995). Real-time scheduling of freight railroads. *Transportation Research Part B: Methodological*, 29(3):213–229.

Li, H., Parikh, D., He, Q., Qian, B., Li, Z., Fang, D., and Hampapur, A. (2014). Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Liu, Y., Tang, T., and Xun, J. (2017). Prediction algorithms for train arrival time in urban rail transit. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Lovett, A. H., Dick, C. T., and Barkan, C. P. (2015). Determining freight train delay costs on railroad lines in North America. In *Proceedings of RailTokyo2015: 5th International Conference on Railway Operations Modelling and Analysis*, Tokyo, Japan. International Association of Railway Operations Research (IAROR).

Lu, Q., Dessouky, M., and Leachman, R. C. (2004). Modeling train movements through complex rail networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 14(1):48–75.

Lütkepohl, H. (2007). *New introduction to multiple time series analysis*. Springer Science & Business Media, New York, NY, USA.

Marinov, M. and Viegas, J. (2011). A mesoscopic simulation modelling methodology for analyzing and evaluating freight train operations in a rail network. *Simulation Modelling Practice and Theory*, 19(1):516–539.

Marković, N., Milinković, S., Tikhonov, K. S., and Schonfeld, P. (2015). Analyzing passenger train arrival delays with support vector regression. *Transportation Research Part C: Emerging Technologies*, 56:251–262.

Medina, J. C. and Benekohal, R. F. (2015). Macroscopic models for accident prediction at railroad grade crossings: comparisons with us department of transportation accident prediction formula. *Transportation Research Record: Journal of the Transportation Research Board*, (2476):85–93.

Meeker, F., Fox, D., and Weber, C. (1997). A comparison of driver behavior at railroad grade crossings with two different protection systems. *Accident Analysis & Prevention*, 29(1):11–16.

Mori, U., Mendiburu, A., Álvarez, M., and Lozano, J. A. (2015). A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science*, 11(2):119–157.

Mu, S. and Dessouky, M. (2011). Scheduling freight trains traveling on complex networks. *Transportation Research Part B: Methodological*, 45(7):1103–1123.

Murali, P., Dessouky, M., Ordóñez, F., and Palmer, K. (2010). A delay estimation technique for single and double-track railroads. *Transportation Research Part E: Logistics and Transportation Review*, 46(4):483–495.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Nelson, E. and Bullock, D. (2000). Impact of emergency vehicle preemption on signalized corridor operation: An evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, (1727):1–11.

Oliver Wyman (2016). Assessment of European railways: Characteristics and crew-related safety. Retrieved from https://www.aar.org/Documents/.

Oruganti, A., Sun, F., Baroud, H., and Dubey, A. (2016). Delayradar: A multivariate predictive model for transit systems. In *Proceedings of the IEEE International Conference on Big Data*, pages 1799–1806.

Ouyang, Y., Xie, S., and An, K. (2018). Positioning, planning and operation of emergency response resources and coordination between jurisdictions. Center for Transportation Studies, University of Minnesota.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Petersen, E. and Taylor, A. (1982). A structured model for rail line simulation and optimization. *Transportation Science*, 16(2):192–206.

Pouryousef, H., Lautala, P., and White, T. (2015). Railroad capacity tools and methodologies in the US and Europe. *Journal of Modern Transportation*, 23(1):30–42.

Richards, S. H. and Heathington, K. W. (1988). Motorist understanding of railroad-highway grade crossing traffic control devices and associated traffic laws. *Transportation Research Record: Journal of the Transportation Research Board*, (1160).

Şahin, İ. (1999). Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research Part B: Methodological*, 33(7):511–534.

Saunders, C., Gammerman, A., and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521, Madison, WI, USA.

Savage, I. (2006). Does public education improve rail–highway crossing safety? *Accident Analysis & Prevention*, 38(2):310–316.

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

Sun, F., Dubey, A., Samal, C., Baroud, H., and Kulkarni, C. (2018). Short-term transit decision support system using multi-task deep neural networks. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 155–162, Tokyo, Japan.

Surface Transportation Board (2016). Decision 45126: CSX transportation, inc.–acquisition of operating easement–grand trunk western railroad company.

Tumulty, B. (2015). Google maps will highlight rail crossings. *USA Today*. 29 June 2015.

United States Department of Transportation and Federal Railroad Administration (2007). Intelligent transportation system/positive train control at highway-rail intersections. *Research Results*, (RR 07-20).

Vromans, M. J., Dekker, R., and Kroon, L. G. (2006). Reliability and heterogeneity of railway services. *European Journal of Operational Research*, 172(2):647–665.

Wang, R. and Work, D. B. (2015). Data driven approaches for passenger train delay estimation. In *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems*, pages 535–540, Washington, DC, USA.

Weatherford, B. A., Willis, H. H., Ortiz, D. S., Mariano, L. T., Froemel, J. E., and Daly, S. A. (2008). *The State of US Railroads: A Review of Capacity and Performance Data*. Rand Corporation.

Wronski, R. (2008). Time limits on trains at crossings barred. *Chicago Tribune*. 26 January 2008.

Zhang, Z., He, Q., Gou, J., and Li, X. (2016). Performance measure for reliable travel time of emergency vehicles. *Transportation Research Part C: Emerging Technologies*, 65:97–110.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.