PREDICTING TRANSIT TIMES FOR OUTBOUND LOGISTICS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Brooke R. Cochenour

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

August 2020

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Zina Ben Miled, Chair

Department of Electrical and Computer Engineering

Dr. Brian King

Department of Electrical and Computer Engineering

Dr. Dongsoo Stephen Kim

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

Head of Graduate Program

I dedicate this thesis to my family. First of all, to my parents Kathie and Mark Cochenour for teaching me that a disability is not delimiting and providing endless support as I attempted to push through my limits. My grandparents Walt Childs, Linda Flanagan, and Mike Flanagan for being there for me and always checking in and offering advice when things got tough. I would like to give a special thanks to Laura Foust and Paul Foust for welcoming me into their home and supporting me throughout the program. Finally, to my fiancé John Foust who has been a pillar of support for me throughout the entire Master's program.

ACKNOWLEDGMENTS

I would like to thank all of my committee members for sharing their expertise and time.

I would like to acknowledge Niels Da Vitoria Lobo, an instructor during my Freshman year of college who saw potential in me and was the first person to encourage me to pursue a graduate degree. I would like to also acknowledge Lynn Krieger who first introduced me to programming in an information technology class and was very supportive and encouraging.

Special thanks go to Jeremy Archbold and Emily Jerger for the opportunity to participate in this project and for their valuable feedback. I would like to also thank Ted Trepanier and INRIX for providing access to the data needed to support this project.

Special thanks to Connor Horne for assisting in packaging the solution and to Alvaro Esperanca for providing the application used to develop the models in this thesis.

This project was completed under the Cooperative Agreement W31P4Q-14-2-0001, between Army Contracting Command – Redstone and MxD USA. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CV	Coefficient of Variation
CI	Confidence Interval
ERD	Entity Relationship Diagram
GPS	Global Positioning System
MAE	Mean Average Error
MAPE	Mean Absolute Percentage Error
NOAA	National Oceanic and Atmospheric Administration
RMSE	Root Mean Square Error
SVM	Support Vector Machine

ABSTRACT

Cochenour, Brooke R. M.S.E.C.E., Purdue University, August 2020. Predicting Transit Times For Outbound Logistics. Major Professor: Zina Ben Miled.

On-time delivery of supplies to industry is essential because delays can disrupt production schedules. The aim of the proposed application is to predict transit times for outbound logistics thereby allowing suppliers to plan for timely mitigation of risks during shipment planning. The predictive model consists of a classifier that is trained for each specific source-destination pair using historical shipment, weather, and social media data. The model estimates the transit times for future shipments using Support Vector Machine (SVM). These estimates were validated using four case study routes of varying distances in the United States. A predictive model is trained for each route. The results show that the contribution of each input feature to the predictive ability of the model varies for each route. The mean average error (MAE) values of the model vary for each route due to the availability of testing and training historical shipment data as well as the availability of weather and social media data. In addition, it was found that the inclusion of the historical traffic data provided by INRIX^{TM} improves the accuracy of the model. Sample INRIX^{TM} data was available for one of the routes. One of the main limitations of the proposed approach is the availability of historical shipment data and the quality of social media data. However, if the data is available, the proposed methodology can be applied to any supplier with high volume shipments in order to develop a predictive model for outbound transit time delays over any land route.

1. INTRODUCTION

In the shipping industry, on-time delivery is essential since the slightest delay can disrupt the customer's supply chain and production schedule [1]. Disruption can lead to a ripple effect, causing brand name erosion and potential loss of market share. This problem is compounded in mass production manufacturing since even a single component delay can lead to the halt of the entire production chain [1]. Typically, different customers address delay mitigation from an inbound perspective either through increased inventory or the ability to source the product from different suppliers. This thesis investigates supply chain delays from an outbound perspective where the supplier proactively attempts to predict potential delays allowing for preventive rather than reactive mitigation of risks associated with these delays. In addition to being preventive, this upstream perspective is more efficient since it also offers more flexible options for remediation.

Outbound logistics delays can be due to multiple reasons, including manufacturing delays and physical distribution delays. The focus of this thesis is to develop a model for predicting physical distribution delays in the specific case of land transportation. Predicting transit time delays is being addressed by several commercial solutions. However, these solutions have three main limitations when predicting delays for supply chain physical distribution. The first is that the available solutions attempt to provide delay estimates based on travel times that encompass both commercial vehicles and heavy-duty vehicles rather than being specific to freight delays. The second limitation is that these solutions primarily apply to near future transit times. Therefore, they are only applicable to shipments that are en-route and have limited usefulness to shipment planning, especially for lead times that extend to multiple days before the actual shipment date. The third limitation is that these solutions do not take into account the specific history and experience of the supplier and tend to target transit times across all manufacturing sectors. Predicting physical distribution delays several days in advance of the actual shipment date for specific suppliers presents challenges and opportunities.

This thesis proposes a model that predicts delays for future shipments. Application administrators train the model for each specific supplier, customer pairs defining a shipment route. For each model, the prediction horizon can vary from one day to seven days ahead of the actual shipment date. Factors that can impact physical distribution delays are taken into consideration to create the proposed machine learning model. These factors include the shipment history from the same source and destination pair, the weather forecast for the planned shipment date, and social media reports of traffic and social events.

2. RELATED WORK

Previous research work related to physical distribution delay prediction is at the intersection of two main areas: transit experience and machine learning. Transit experience ranges from freight cost and transit time estimation to shipment tracking. Several commercial solutions focus on addressing these issues. Both commercial solutions and research work have also successfully used machine learning techniques to develop various classification and predictive models for predicting the transit time of vehicles. This chapter includes a review of related commercial solutions and research applications.

2.1 Commercial Solutions

Several existing applications provide added value to the shipment experience in general and to the physical distribution experience in particular. For example, Kuebix$^{\mathrm{TM}}$ [2] is an end-to-end transit solution that manages all product information from shipping and transit information to temperature monitoring and expense allocation. While shipping routes and delays are not the primary focus of this application, the underlying methodology relies on weather, season, fuel cost, and regulations to estimate freight costs for different shipping options.

PCMiler$^{\mathrm{TM}}$ [3] is an example of a commercial solution that focuses on estimating transit times derived from real-time traffic and weather updates for different routes. PCMiler $^{\mathrm{TM}}$ extracts estimates for traffic and transit times from INRIX$^{\mathrm{TM}}$ [4]. These estimates are derived from current and historical traffic data for each route by aggregating travel times of the segments that make up the route.

FourKites$^{\mathrm{TM}}$ [5] is a near real-time tracking solution for mixed-mode shipments by land, sea, air, and rail. It relies on vehicles equipped with electronic logging devices, which are pinged for their status every 15 minutes. Notifications can be customized to alert different stakeholders about the status of each shipment. In addition to GPS and temperature tracking, the solution also provides functionality for trend analysis over historical data.

Emicen$^{\text{TM}}$ [6] uses Bayesian networks, a machine learning model, to predict delays based on the customer's historical data. This solution targets inbound logistics rather than outbound logistics. Customer historical data is transformed into a knowledge map that captures the conditional probability of delays for each product. This information allows customers to adjust order and inventory margins in order to mitigate risks for products associated with high delay probabilities.

The model proposed in this thesis is different from the above mentioned commercial solutions because the objective is to predict shipment delay during the planning phase rather than during the shipment execution phase. Most commercial solutions, except for Emicen$^{\text{TM}}$, focus on real-time predictions. Emicen$^{\text{TM}}$ predicts delays in future inbound shipments based on the distribution of the historical data. Compared to Emicen$^{\text{TM}}$, the focus of this thesis is on predicting delays in future outbound shipments. Moreover, while Emicen$^{\text{TM}}$ relies only on shipment data, the proposed approach in this thesis uses both internal shipment data as well as external data such as weather and social media data.

2.2 Research Applications

Several transportation machine learning models have been proposed in the literature. Most of these models focus on general (public or individual) as opposed to freight transportation.

In 2002, an artificial neural network (ANN) model for predicting bus arrival time was proposed in [7]. The model uses two different types of data: link-based and stop-based data. The ANN is first trained on link-based data. Adding travel times on links between pairs of stops provides the arrival time at specific stops. A second ANN handles stop-based data between a pair of stops. An enhanced model combines the two methods: link-based and stop-based. The results show that the enhanced model can accommodate single and multiple stops. The stop-based model has improved performance when stops have multiple intersections between them, and the link-based model is more suitable for pairs of stops with limited number of intersections.

More than a decade ago, a survey [8] of different general transit time predictive models highlighted the impact of data quality and availability on the predictive accuracy of the models. This survey indicated that the performance of transit time models is poor when there is traffic congestion. Transit time models for both private and public transportation evolved considerably since this survey, and several are currently widely used.

We review two examples of earlier commuter transit models. The industry commonly uses these earlier models today. A Kalman filter model for short-term travel time prediction on freeways was introduced in [9]. The parameters of the model are updated in real-time based on the speed of probe vehicles in a two-step process. The first step collects new observation data and updates the traffic estimates. The second step uses traffic estimates to predict travel time.

A second commuter transit model was introduced in [10]. This model uses historical and real-time GPS vehicle location data to estimate the time of arrival. It takes into consideration average speed and stops. The model also relies on both historical and real-time data. A first-order-linear model is used to represent historical data. The output of the historical data model is then adjusted according to variations in the observed data and real-time position data collected from an advanced vehicle location (AVL) system. The AVL system determines and transmits the absolute position coordinates of a vehicle using a GPS. This position is used to estimate the average speed and calculate the arrival time of the vehicle.

For public transportation, research effort is ongoing. A model that predicts the next bus arrival time based on the distance between the vehicle of the commuter and the bus stop, and the average vehicle speed is proposed in [11]. In this study, travel time is defined as the running time on route sections and does not take into consideration additional delays due to other sources such as traffic signals, time for passengers getting on and off each bus stop, and stop time. This information has to be accounted for separately by keeping track of the total delay time. The bus arrival time results from summing all delays on the way to the designated bus stop and the travel time.

Two predictive models for bus transit time that enhance the above model are proposed in [12, 13]. As in the case of this thesis, the first model uses a SVM. This application creates a multi-index evaluation model and uses GPS coverage to predict arrival time when the GPS location cannot be determined. GPS coverage is a triangulation based on the position of the buses equipped with GPS. Each bus stop has a different traffic pattern, and error tolerance is subject to the actual time spent at the stop. GPS coverage and release rate accuracy are used as evaluation indices in the model. The release rate is the percentage of the number of buses that have released prediction information to the number of buses that have GPS onboard. The accuracy rate is the main performance evaluation index for the model. It represents the relative error between the actual time spent traveling and the predicted travel time. The SVM machine learning technique is used in this application to develop the bus prediction models. The features of the model are the GPS coverage, release rate, and accuracy rate. Evaluation results are classified into five grades: excellent, good, average, poor, and failing. Grades are in descending order, with excellent being the best grade and failing being the worst grade. When coverage is at 100%, GPS coverage is considered excellent. Every 10% decrease indicates a drop in grade. A failing grade is assigned when the release and accuracy rates are 60% or below. Similarly, every 10% increase indicates a grade upgrade. Once a GPS based prediction model for bus arrival time model has been trained and evaluated using the three performance indexes; each of the three indexes are assigned a grade. This model then uses these grades to rate the data used during training.

The second model [13] is a linear model that relies on both GPS location and real-time traffic flow. It uses traffic conditions specific to each driving segment on the bus route. An impact factor that results from a linear equation is used to represent the traffic condition. The traffic condition enhances the GPS location, and the enhanced calibration position information becomes an input to the model.

More recently, a real-time model for tracking bus locations was introduced in [14]. The model uses traffic information, weather condition, real-time tracking, and bus riders' experiences. Each route is divided into links. Links represent segments between bus stops on different routes. The input to the model includes the transit time for the link from the previous bus stop to the current stop and the link from the current stop to the next stop. In addition to the link information, the model input also includes weather conditions such as cloudy, clear, or rainy. In [14], two machine learning techniques are used: Simple Moving Average Model (SMA) and Artificial neural networks (ANN).

The results show that different stops result in different ANN models with varying weights because of the difference in the distribution of the data associated with different bus stops.

The MAE and RMSE for a hybrid model based on SMA and ANN were reported to be less than one minute. As in the case of the model proposed in this thesis, 1) weather was determined to be a primary factor that affects bus arrival times, and 2) the predictive contribution of each input feature differs for each model-specific route.

Research into freight transit time prediction is emerging. This effort increased with the availability of new data collected from electronic logging devices onboard of trucks (e.g., FourKites$^{\mathrm{TM}}$ [5]) as well as shipment data from different stakeholders in the supply chain. The added complexity in developing models for freight transit time is due to the limited amount of data as well as to the heterogeneity of the modes of transportation, which can include sea, land, and rail. Few recent representative studies that explore the use of machine learning for freight management are described next. These studies fall under three main categories: freight volume prediction, freight type classification, and freight transit time.

Several current studies [15–17] focus on modeling freight volume. In [15], the authors investigate road freight volume including Winter's seasonal method, harmonic analysis, and artificial immune system aiding the harmonic analysis.

Winter's seasonal method [15] is a set of formulas used for forecasting. In [15], five formulas are used to mimic a multiplicative version of Winter's seasonal method. The equations are obtained from [15]. Equation 2.1 forecasts for future periods when $t > n$ or $t < n$ where t is the time period, and n is the number of observations.

$$y_t^* = \begin{cases} (F_n + (t - n) \cdot S_n) \cdot C_{t-r} & \text{if } t > n \\ (F_{t-1} + S_{t-1}) \cdot C_{t-r} & \text{if } t < n \end{cases} \tag{2.1}$$

In the above equations, F_t represents the smoothed evaluation for t's average value level, S_t represents a smoothed trend growth value for t, and C_t is the seasonality index for t evaluated. These two equations together forecast future periods for freight volume.

Harmonic analysis [15] is done by creating a sum of harmonics as illustrated by Equation 2.2 where $f(t)$ is the trend function, n is the month, i is the count of harmonics, t is the time period, and α_i and β_i are coefficients

$$y_t = f(t) + \sum_{i=1}^{\frac{n}{2}} (\alpha_i \cdot \sin(\frac{2 \cdot \pi}{n} \cdot i \cdot t) + \beta_i \cdot \cos(\frac{2 \cdot \pi}{n} \cdot i \cdot t)). \tag{2.2}$$

Harmonic analysis aided by the artificial immune system [15] is an altered version of the harmonic function which is shown in Equation 2.3 where α_{i} is a coefficient, n is the month, i is the number of harmonic, t is the time period, and m is number of elements in the time series

$$y_t = \alpha_0 + \alpha_1 + t + \sum_{i=1}^{m} (\alpha_{2i} \cdot \sin(\frac{2 \cdot \pi \cdot i}{n} \cdot t) + \beta_i \cdot \cos(\frac{2 \cdot \pi \cdot i}{n} \cdot t)). \tag{2.3}$$

Root mean square error (RMSE) and mean absolute percentage error (MAPE) are used to evaluate the effectiveness of the above prediction methods. The harmonic analysis aided by the artificial immune system reported the smallest error values with a MAPE value of 4.14%. More recently, the same authors published a more accurate model [17]. The new model is a variation of the Holt-Winters method [15] and is defined by Equations 2.4, 2.5, 2.6, and 2.7. In these equations, t represents the index of the time series, F_t is the variable forecasted at time t smoothed, S_t is the smoothed value of growth of the trend at a specific moment t, n is the number of the month, y_t represents the freight volume for a given year t, α is a data smoothing factor between zero and one, and β is a trend smoothing factor between zero and one. The optimized values for α and β were determined using the average square error.

$$F_1 = y_1 \tag{2.4}$$

$$s_1 = y_2 - y_1 \tag{2.5}$$

$$F_{t-1} = \alpha \cdot y_{t-1} + (1 - \alpha) \cdot (F_{t-2} + S_{t-2}) \tag{2.6}$$

$$S_{t-1} = \beta \cdot (F_{t-1} - F_{t-2}) + (1 - \beta) \cdot S_{t-2} \tag{2.7}$$

Equation 2.8 determines the forecasted freight volume where t represents the time period and n is the number of observations. The variable y_t^* represents the forecasted freight volume for a given time t.

$$y_t^* = \begin{cases} F_{t-1} + S_{t-1} & \text{if } t \leq n \\ F_n + (t - n) \cdot S_n & \text{otherwise} \end{cases} \tag{2.8}$$

When the immune system variant is used, the model assumes that both F_1 and S_1 are independent variables. The MAPE for the model [17] was 2.5% compared to 4.14% for the earlier model [15].

The model proposed in [17] was compared to a Bayesian network. The Bayesian network took into account the historical freight volume data as well as other metrics indicative of the overall state of the national economy. The forecasting quality was significantly improved when the national economy metrics were included in the Bayesian network.

The focus of the study in [16] is freight type classification. That is the freight type moving between a source and destination location. Several classification techniques were evaluated and those that have higher accuracy were reported. These techniques include k-nearest neighbors and LogitBoost. The k-nearest neighbors classifier gathers predictions from neighbors and weights them according to their distance to a test record [16]. The dataset used to develop and test the model [16] consists of four attributes: source, destination, freight weight, and freight type. The waybill typically contains these attributes. An imputation algorithm was used to replace missing values by observing values around the missing instance. The ten folds cross-validation provided a performance evaluation of the sensitivity of the models to the training data. The ten folds cross-validation partitions a dataset into ten sets of equal sizes and then trains on nine datasets and tests on a single dataset. This validation method repeats ten times, and then the mean accuracy of these ten tests is calculated and used as a performance evaluation index for the models [16]. The results of the comparative study showed that the k-nearest neighbors algorithm performed the best with an accuracy rate of 82.72%.

The focus of the next three studies is on estimating freight transit time. In [18], a bimodal Gaussian mixture model (GMM) and a long short-term memory (LTSM) model were used to characterize driving speed. The input variables to the models were the horizontal curvature, vertical curvature, horizontal curve length, vertical curve length, cross slope, cross-section width, and 3D available sight distance from a 3D road map. Multiple Gaussian probability functions are applied by the GMM to estimate speed distribution. The model output variables were used in the GMM to show the speed distribution. The study shows that the LTSTM model had the best performance.

In [19], a Gradient Boosting Regression Tree (GBRT) model for the prediction of the travel time of freight vehicles is proposed. The model uses a training set consisting of a D-dimensional input vector x and a one-dimensional target vector y. The goal of GBRT is to find a latent function f that maps x to y while minimizing the expected value of a loss function over a joint distribution between x and y. It does this through gradient boosting. Boosting algorithms are iterative approaches to finding a simple regression function while minimizing error [16]. The study defines base learners as simple function and expansion coefficients [19]. The functions expand iteratively to higher levels up to a specified value. The model considers base learners solved when they fit with pseudo-residuals that meet a pre-defined loss criteria. Training and testing data cover three routes. The basic features of the model consist of the departure time, day of the week, month, day in the month, day in the year, weekday, workday, and public holiday in addition to historical travel times and mean speed sequence where each mean speed in the sequence is estimated using trajectory data. A fifteen-minute observation window constructs a trip. Trips are defined as a tuple. The tuple contains three groups of features: basic, historical travel time, and mean speed sequence. The study examines feature importance for the GBRT models by measuring the frequency of occurrences of each feature in all splits of the decision trees. The feature analysis showed that historical interval mean travel time and departure time were critical features in the model. The study presented a reduced feature model. This reduced model consisted of all features except for weekday, month, and workday.

The model proposed in the above-mentioned study is a real-time model that uses real-time data to supplement historical data. This model differs from the model presented in this thesis, which allows for a prediction horizon of seven days.

In [20], containership arrival for seaborne trade is predicted using a fuzzy rule-based Bayesian network. Several factors were determined to impact the prediction accuracy, including port-channel conditions, terminal conditions, port administration process, and inland corridors. Results show prediction errors range between 4.2% and 6.6%, a margin that was considered adequate.

The model proposed in this thesis differs from the above-mentioned applications as it aims to estimate transit time for future road shipments during the shipment planning phase. Moreover, the proposed model relies on an expanded feature set that uses the carrier and loading point of the shipment. It also extends to external factors that could affect the shipment travel time, such as weather and social media information. These factors provide a more comprehensive set of inputs for predicting the transit time of a shipment.

3. METHODS

Predictive modeling can benefit supply chain in several phases from production to distribution. In this thesis, we focus on developing a predictive model for estimating transit time during physical distribution. Moreover, the model supports an extended prediction horizon because the goal is to use the estimates generated by the model during shipment planning, which can occur days before shipment execution.

This chapter describes the dataset, the data processing steps, and the predictive transit time model.

3.1 Dataset

Four routes of varying distances and geographical locations were chosen from an operational shipment database of a supplier. Shipments associated with these routes develop and validate the proposed model. The other sources of data are Twitter$^{\text{TM}}$ [21] and weather [22,23]. The data extracted from these sources collectively make-up the input feature space of the proposed model, as shown in Table 3.1. The next section describes each component.

3.1.1 Supplier Shipment Data

Table 3.2 shows an example shipment. Except for the shipment date, all the supplier features are categorical. These features include:

- Shipment Number (*ShipNum*) which consists of a unique key that identifies each shipment.
- Shipment type (*ShipType*) which indicates whether the shipment is a full truckload, partial truckload, tank truck, etc.

Table 3.1.: Model Input Features.

Feature	**Definition**
	Supplier
Sdate	Shipment date
ShipType	Shipment type (e.g., full truckload, partial truckload, ...)
DelivPriority	Delivery priority
DelivItem	Delivery item
Carrier	Carrier identification
DangGood	Dangerous good indicator (Yes/No)
LoadingPoint	Loading point for the shipment
	Twitter^{TM}
road, event, accident, traffic	Count of tweets for each keyword.
	Weather
Tmax, Tmin	Maximum and minimum temperature in tenth of Celsius
Rmax, Rmin	Maximum and minimum rain in Millimeters
Smax, Smin	Maximum and minimum snow in Millimeters

- Carrier (*Carrier*) which represents the carrier assigned to the shipment.
- Delivery Item (*DelivItem*) which represents the product being delivered.
- Delivery priority (*DelivPriority*) is the priority assigned to the shipment. The priority is included in the model because in can impact the transit time of the shipment.
- Dangerous good (*DangGood*) indicates whether or not the shipment consists of dangerous goods. This factor can affect transit time since some roads have hazardous material restrictions.
- Loading Point (*LoadingPoint*) at the source facility assigned to the shipment.

Table 3.2.: Example Shipment.

Field Name	**Description**	**Example**
ShipNum	Unique ID of the Shipment	1
Source	Five digits source zip code	46220
Dest	Five digits destination zip code	46143
Sdate	Date the shipment left the source facility	10/31/2019
ShipType	Shipment Type	1
Carrier	Carrier assigned to the shipment	Carrier_A
DelivItem	Type of item being delivered	100
DelivPriority	Delivery priority of the shipment	1
DangGood	Dangerous Goods Indicator	0
LoadingPoint	The loading station at the source zip code	1

3.1.2 Route Information

Route information is collected for each (source, destination) pair. The five-digit zip codes are used to represent the source and destination. Five-digit zip codes are used in this project, instead of the ten-digit zip codes, because several applications only support restricted zip codes in the United States. The route information consists of geofences along the route from the source to the destination. This information is used to collect relevant Twitter$^{\text{TM}}$ and weather data. The process consists of three steps, as illustrated in Figure 3.1.

The first step consists of converting the source and destination zip codes into geocodes. This process is known as forward geocoding and is performed using the Geocode.CA$^{\text{TM}}$API [24], as shown in Figure 3.2.

During the second step, the OSRM$^{\text{TM}}$ API [25] takes the geocoded source and destination and generates random nodes along the route between the source and destination zip codes, as shown in Figure 3.3. A temporary file stores the nodes returned by the OSRM$^{\text{TM}}$.

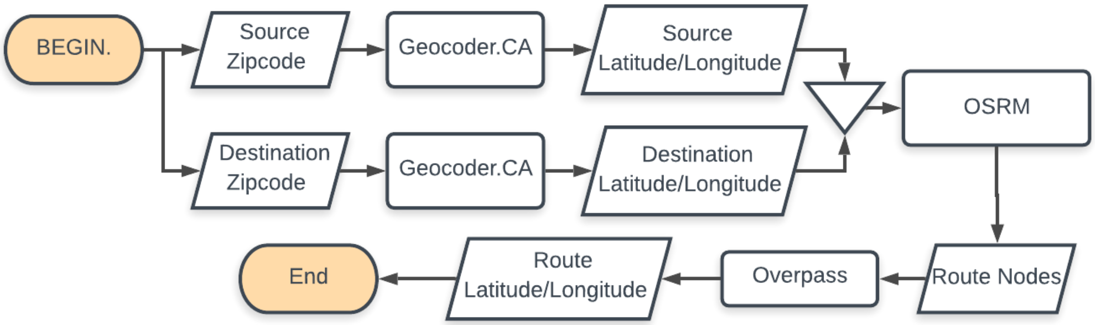


Fig. 3.1.: Route Information Collection Process.

```
# key = authentication key for API
# address = zip code for forward geocode
URL=”https://geocoder.ca/?locate=”+address+”&json=1&auth=”+key
```

Fig. 3.2.: API Request URL Format for geocode.ca.

```
# lats = latitude source
# longts = longitude source
# latd = latitude destination
# longtd = longitude destination
URL = ”http://router.project-osrm.org/route/v1/driving/” +
    longts + ”,” + lats + ”;” + longtd + ”,” + latd + ”?
    alternatives=false&annotations=nodes”
```

Fig. 3.3.: Example OSRM^{TM} API [25] URL Request.

In the third step, the nodes along the route are converted to geocodes using the Overpass^{TM} API [26]. They are retrieved from the temporary file created in the previous step and submitted to the Overpass^{TM} API, as shown in Figure 3.4. The API returns a JSON object that consists of a sequence of (latitude, longitude) pairs.

```
# node = a node from OSRM API
URL = ”https://www.overpass−api.de/api/interpreter?data=[out:
    json];node(” + node + ”);(._;%3E;);out;”
```

Fig. 3.4.: Example Overpass$^{\text{TM}}$ API [26] URL Request.

3.1.3 Twitter$^{\text{TM}}$ Information

Twitter$^{\text{TM}}$ data is extracted using a set of query keywords, the shipment date, and geofences created from the route nodes (longitude, latitude) described above.

A five-mile radius surrounding every nth (latitude, longitude) pair defines the geofence, where n is determined using the length of the route. The ratio of route length to the number of nodes varies per route. For example, route A is 69 miles long and has 1030 nodes, while route D is 288 miles and has 3021 nodes. Route A has a mile to node ratio of 14.9, while route D has a mile to node ratio of 10.5. Table 3.3 illustrates the variations in the node ratios. The number of nodes for each route can be extensive and a method is needed to sub-sample the nodes along each route. Initially, a node was selected every 5 miles. However, this approach was not efficient because it did not differentiate between long and short routes. It also led to gaps that ignored relevant tweets.

Table 3.3.: Route Mile To Node Ratios.

Route	**Mile To Node Ratio**
A	14.9
B	20.9
C	13.2
D	10.5

In order to reduce the number of nodes, Equations 3.1 and 3.2 are used to determine the number of nodes needed per route to extract the Twitter$^{\text{TM}}$ data. In Equation 3.1, d represents the length of the route in miles. Five is used as the divisor because the radius selected was five miles. This equation gives an estimate of the number of nodes that should be in the route fence for each route.

$$est = \frac{d_{mi}}{5} \tag{3.1}$$

Nodes, when gathered, are non-uniform and can vary. N determines the number of nodes collected. Every nth node is skipped when parsing the route fence in an attempt to limit the number of nodes to one node every five miles. Equation 3.2 is used to calculate the value of N.

$$N = \frac{node_{total}}{est} \tag{3.2}$$

In this equation, $node_{total}$ represents the total number of nodes that comprise a given route. Once N has been calculated using Equation 3.2, when making API calls, the geofence disregards every nth node while constructing the route fence. Table 3.4 shows the before and after node sub-sampling counts.

Table 3.4.: Route Lengths and Nodes

Route	**Distance (km)**	**Distance (mi)**	**nodes prior**	**nodes after**
A	111	69	1029	15
B	193	120	2508	26
D	463	288	3021	93
G	309	192	2535	63

The Twitter ^{TM} API call [27] searches each geofence along the route for all tweets containing the keywords: *event*, *accident*, *road*, and *traffic* on a given date. These query keywords were selected based on an examination of several tweets across the routes. For instance, contexts such as “road congestion” and “road construction” are aggregated under the keyword “road” because of the co-occurrence of the underlying terms.

3.1.4 Weather Information

A weather station list is constructed for each route. Weather stations cannot be identified from NOAA$^{\text{TM}}$ [22] using latitude and longitude. They are identified manually along each route based on their proximity to the centers of the geofences using the NOAA$^{\text{TM}}$ find station tool. The source and destination zip codes filter the search, and weather stations within the proximity of the route nodes are selected and added to the station list.

There are two types of weather data: 1) historical data, used to train the predictive transit model, and 2) forecast data, used to generate transit time estimates for future shipments. The NOAA$^{\text{TM}}$ yearly archive files provide the historical weather data, and the date and station list filter the data. For any given day, the application calculates the maximum and minimum temperature, rain, and snow across all of the weather stations in the station list for a given route. These values are used as an input to the model as shown in Table 3.1.

The weather.gov$^{\text{TM}}$ forecasting API [23] is used to retrieve weather forecasting data. The API returns raw data in JSON format. This API supports a seven-day forecasting window. Figure 3.5 shows the corresponding API call using the future shipment date for each (latitude, longitude) node in the route. As in the case of the historical weather data, the return values are aggregated into maximum and minimum temperature, rain, and snow along the route.

```
"https://graphical.weather.gov/xml/sample_products/browser_interface/
    ndfdXMLclient.php?lat"+latitude+"&lon="+longitude+"&product=time-
    series\&begin="+dayStart+"&end="+day +"&maxt=maxt&mint=mint&qpf=qpf&
    snow=snow"
```

Fig. 3.5.: Weather.gov$^{\text{TM}}$ API [23] Call Format.

3.2 Database

The data retrieved from all the sources is consolidated into a database consisting of five groups of tables:

- Dictionary tables which are used to encode the shipment data,
- Route tables which include the route information,
- Stat table which maintains the statistics corresponding to each transit model,
- Historical shipment table which includes the shipment information used to train the model, and
- Future shipment data table.

Figure 3.6 shows the ERD of the entire database, and the five groups of tables are described below.

The dictionary tables convert different string attributes into an encoded integer value. This group includes the followings tables: *dangerousGoods*, *DeliveryPriority*, *carrier*, *lateDelv* and *LoadingPoint*. Each table maintains the string-to-integer mapping for a given attribute. The *lateDelv* table is used to code the *lateDelv* parameter. The dictionary tables must be reviewed regularly since the supplier may add a new carrier or a loading point.

The *route* table includes a record for each route where a record consists of the route number, the source and destination zip codes of the route. The *routefence* table stores the sequence of latitude and longitude pairs for each route. Similarly, the *stationlist* table holds the list of NOAA^{TM} weather stations along each route. These tables are generated once for each route and are used by the model to extract weather and Twitter^{TM} data.

The history tables are organized by source zip code. That is, all routes that start from the same source are associated with a single history source table. A single table is shown in Figure 3.6 to represent the history tables. The attributes in this table consist of a combination of the parameters in Tables 3.2 and 3.5

Table 3.5.: Attributes of History Source Tables in the Database.

Attribute	**Description**
ShipNum	Unique ID of the Shipment
ShipType	Represents whether shipment is a full truck-load, partial truck-load, tank truck, etc...
Dest	Five digit destination zip code
Sdate	Date the shipment left the source facility
Day	Day of month shipment leaves source facility
Month	Month shipment leaves source facility
Year	Year shipment leaves source facility
Carrier	Carrier assigned to the shipment
DelivItem	Type of item being delivered
DelivPriority	Delivery priority of the shipment
DangGood	Dangerous Goods Indicator
LoadingPoint	The loading station at the source zip code
Tmax	Maximum temperature on the date the shipment left source facility
Tmin	Minimum temperature on the date the shipment left source facility
Rmax	Maximum rain on the date the shipment left source facility
Rmin	Minimum rain on the date the shipment left source facility
Smax	Maximum snow on the date the shipment left source facility
Smin	Minimum snow on the date the shipment left source facility
road	Count of Tweets with keyword road the date the shipment left source facility
event	Count of Tweets with keyword event on the date the shipment left source facility
accident	Count of Tweets with keyword accident on the date the shipment left source facility
traffic	Count of Tweets with keyword traffic on the date the shipment left source facility
transit time	transit time of the shipment

The *newshipments* table stores the shipments that are currently being planned. The table has similar attributes to those of the history tables. This table is used by the predictive transit model to estimate the transit time for future shipments. Most of the Table 3.5 attributes are present in the *newshipments* table. The *newshipments* table does not have the *lateDelv* and *transit time* attributes. Instead, it has an attribute called *transit time prediction*, which holds the transit time value predicted by the model for each future shipment. The *road, event, accident,* and *traffic* fields contain the most up-to-date Twitter$^{\text{TM}}$ data for the route. The *Tmax, Tmin, RMax, Rmin, Smax,* and *Smin* fields contain forecasted weather information. This information is updated when the Twitter$^{\text{TM}}$ data is updated. It is recommended to update this information on a daily basis to get the most accurate predictions possible.

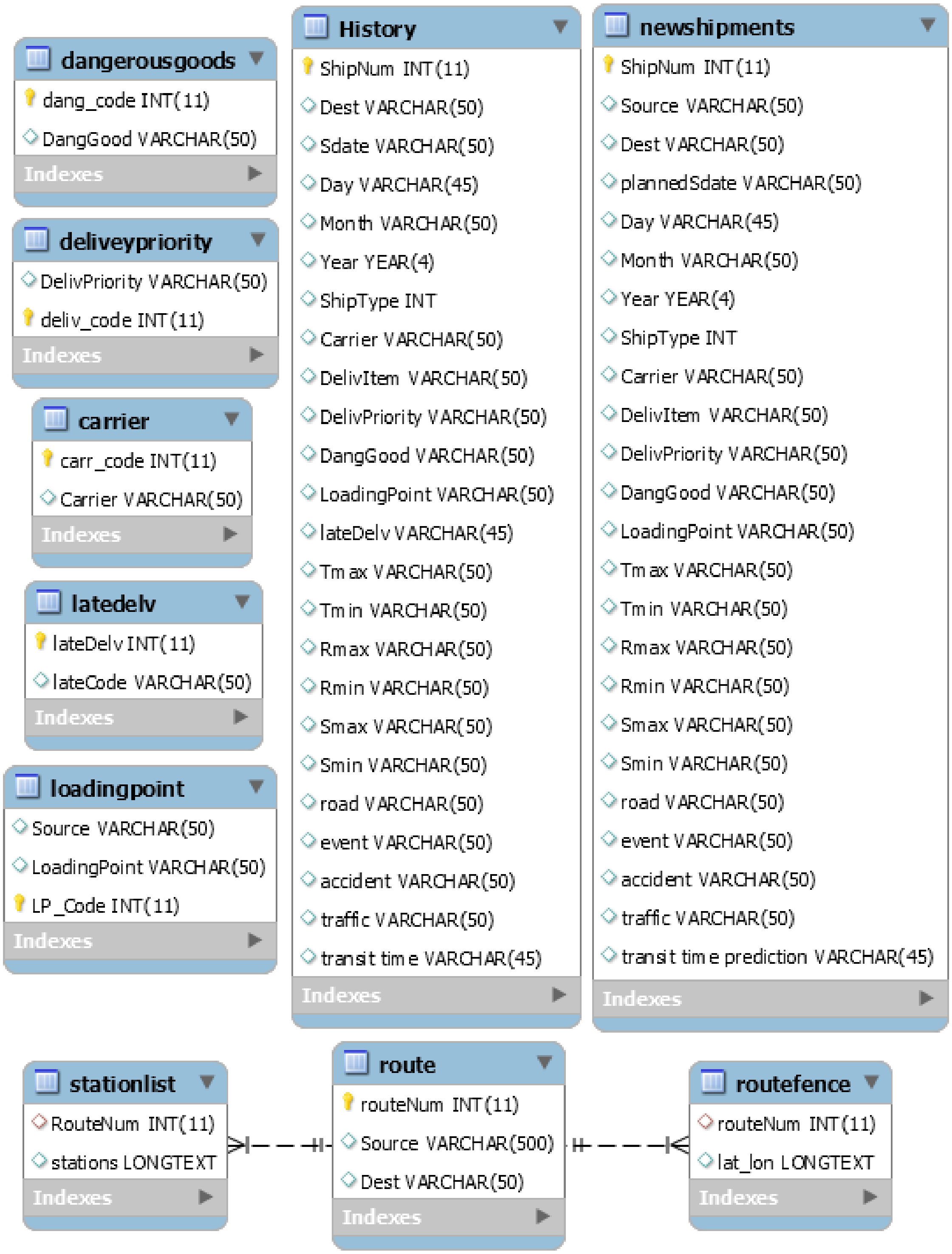


Fig. 3.6.: Database ERD.

3.3 Transit Time Models

A machine learning model is developed for each route. Each model is an ensemble of multiple classifiers that are trained using a set of shipments and tested using a different set of shipments. An SVM with a radial basis function kernel (RBF) is the basis of each classifier. The shipment data ranges from 01/01/2017 to 6/30/2019. The training shipment range covers two-years (i.e., from 01/01/2017 to 12/31/2018). The model uses the remaining shipments (i.e., from 01/01/2019 to 6/31/2019) for testing. Using this data range split corresponds to the recommended 80/20 split between training and testing as well as aligns with the aim of the model of using historical shipment data to estimate the transit time for future shipments.

Transit time for historical shipment varies for each route. The transit time range determines the number of classifiers used in the model for each route. Transit time measurements are in the number of days. For example, when the transit time equals 0 days, the shipment is delivered on the same day. Similarly, a transit time of 1 day indicates a next day delivery.

The minimum transit time of all the shipments on the target route represents the minimum value of transit time. The average transit time plus two standard deviations defines the maximum transit time for a route. Data entry errors often create outliers in the dataset. For example, a route with an average transit time of one day can include a historical shipment with a transit time equal to 30 days because of an error in an entry in the month of the shipment. The threshold imposed on the maximum transit time reduces these outliers and limits the number of classifiers needed for each route.

The transit time range for each route as defined above will have multiple days. For each of these days a classifier is developed. For example, a range of 1-3 days has a classifier that trains using one day as the target, a classifier that uses two days as a target, and a classifier that uses three days as its target. Each classifier trains using the one-versus-all approach [28]. For example, the one-day classifier considers shipments with a transit time of one day as positive and all other shipments as negative.

The prediction from all classifiers is combined to generate an estimate for the transit time of the shipment. Each classifier is a binary classifier that returns either a true or a false depending on the test shipment. This indicates whether the test shipment is estimated to have a transit time equal to the classifier or not. For example, let a given model route model have three classifiers corresponding to transit times of 1 day, 2 days, and 3 days. Each test shipment on this route submits to all three classifiers. If the results indicate true for a single classifier, then that classifier delay is assigned as an estimate of the test shipment transit time. If more than one classifier returns true, then the transit time of the shipment is estimated as the average of the delays of these classifiers. If all three classifiers return false, then the estimate transit time is set to the mean of the maximum and minimum transit times for the model.

The accuracy of the ensemble model is evaluated using the MAE and RMSE as defined in Equations 3.3 and 3.4 respectively. In these equations, i represents the number of shipments in the test dataset.

$$MAE = \frac{1}{i}\sum_{1}^{i}|predicted \quad transit \quad time - actual \quad transit \quad time| \tag{3.3}$$

$$RMSE = \sqrt{\frac{1}{i}\sum_{1}^{i}(predicted \quad transit \quad time - actual \quad transit \quad time)^2} \tag{3.4}$$

Each classifier is built using SVM. Equations 3.5 [29], 3.6 [30], 3.7, 3.8 [31], and 3.9 [31] are associated with the RBF SVM used in this thesis. In Equation 3.5, $\overrightarrow{x}$ represents the input feature vector of the model and $\overrightarrow{y}$ represents the target label vector. Gamma is calculated using Equation 3.7 where j represents the number of inputs to the model. Multipliers are solutions to the quadratic optimization problem solved using a quadratic solver that minimizes Equation 3.8 and maximizes Equation 3.9 [31]. In these equations, P is the matrix of solver parameters, x and s are the primal variables, y and z are the dual variables, and matrices A and b contain equality constraints where A is a sparse matrix and $\overrightarrow{b}$ is a single-column dense matrix [31]. x and y represent a single element from $\overrightarrow{x}$ and $\overrightarrow{y}$ respectively. Solutions are represented in $\overrightarrow{b}$. The SVM algorithm searches for optimal solutions that will solve the hyperplane that follow the following constraints for Equations 3.8 and 3.9. Equation 3.8 has two constraints to abide by: (1) $Gx \leq h$ and (2) $Ax = b$. Equation 3.9 also has two constraints: (1) $q + G^T z + A^T y \in range(P)$ and (2) $z \geq 0$.

$$kernel = exp(-||X - Y||^2 * (\gamma)) \tag{3.5}$$

$$||X|| = \sqrt{(\sum_{i,j} |(x_{i,j}|^2)} \tag{3.6}$$

$$\gamma = \frac{1}{j} \tag{3.7}$$

There are several parameters associated with the model and these parameters need to be calibrated for each specific application. A tolerance level of 1×10^{-5} was used in this application. A range between 0 and 1×10^{-8} was tested. The tolerance level that produced the least amount of error was then selected.

The model did not use C, a constraint in the constrained optimization formulation of the SVM. C is used to calculate G and h when assigned a value.Tests were conducted with negative and positive C values ranging from 1×10^{-5} to 1×10^{1}. This parameter did not improve the performance of the model. Due to this reason, C is set to none. When the C value is set to none, the quadratic solver excludes C from the calculation of G and h. In this thesis, G is the diagonal of a 1xn matrix filled with negative ones, and h is a 1xn matrix of ones where n represents the number of training data. The quadratic optimization Equations (3.8 and 3.9) use the matrices G and h in the calculations.

$$(\frac{1}{2})x^T Px + Px + q^T \tag{3.8}$$

$$-(\frac{1}{2})(q + G^T z + A^T y) P^t (q + G^T z + A^T y) - h^T z - b^T y \tag{3.9}$$

According to [32], SVMs are considered the most robust and accurate classification techniques. SVMs have a single optimal solution for a problem [33, 34]. Learning problems are reduced into optimization problems by SVMs [34]. SVMs operate by maximizing margins and creating the largest possible distances between a separating hyperplane and the instances on either side [32]. The models in this thesis uses an RBF kernel. A kernel is used to transform the input space into a higher dimensional space where classification can be performed.

The SVM has several limitations; among these are feature selection and parameter selection [34]. Standard SVMs cannot select important features. To overcome this limitation, SVMs are typically complemented with feature selection strategies such as the wrapper-type method used in this thesis [34]. Other approaches for choosing an SVM parameter also exist. An effective approach, outlined in [34], consists of estimating the generalization error and then searching for parameters that minimize the estimator.

4. RESULTS

In this chapter we present the results of applying the predictive transit model to the four routes selected from the supplier database. We also show the application that was developed to facilitate the training and testing of the model in production.

4.1 Model Validation

Table 4.1 shows the four routes used to validate the accuracy of the models. These routes are labeled A through D. They have varying distances, headings, and average transit times. Some of these routes share the same source zip code, but all travel to different destination zip codes. Routes A and B have the same source. Routes C and D have a unique source each.

Table 4.1.: Characteristics of the Routes.

Route	**Distance (km)**	**Heading**	**Average transit time (days)**	**Transit time Standard Deviation**
A	111	NE-north	0.7	0.7
B	193	NE-south	0.8	0.7
C	309	SE-west	1.6	7.0
D	463	SE-west	2.0	15.2

Some of the routes cover a longer distance than other routes. These routes tend to have higher average transit times. Also, as illustrated in Table 4.1, the standard deviation of the transit time of the routes varies considerably.

For example, Route C has an average transit time of 1.6 days with a standard deviation of 7.0 days, while route A has an average transit time of 0.7 and a standard deviation of 0.7. The transit time for route D is high because of four outliers in the dataset, three of which mark the transit time to be 365 days and more. These outliers could be due to human error when entering the *Sdate* field information. With these four outliers excluded, the transit time standard deviation for route D is 1.0.

Table 4.2.: Number of Shipment Records, MAE, and RMSE for each Route Model.

Route	**Number of training shipments**	**Number of testing shipments**	**MAE**	**RMSE**
A	1923	346	0.56	0.75
B	989	142	0.47	0.66
C	1267	254	0.74	1.07
D	1403	364	0.85	1.08

Table 4.2 depicts the number of training and testing shipments used for each model, along with the MAE for each route. The models in this table use all of the input features shown in Table 3.1. The MAE, calculated using Equation 3.3, represents the difference between the predicted and actual transit times. Lower MAE values indicate more accurate models. Most of the models train with more than 1,000 shipments. Table 4.2 shows that the MAE values for each route are less than the standard deviation of the transit time for all the routes. The RMSE is calculated using Equation 3.4 and represents a higher weighted difference between the predicted and actual transit times. The RMSE is less than the standard deviation of the transit time for routes B and D, but greater than the deviation for routes A and C.

Moreover, the table shows that models for longer routes are more accurate than shorter routes. Table 4.3 shows the coefficient of variation (CV) values for each route. The CV values represent the relative standard deviation. These value show that the deviation in the predictions between shorter routes and longer routes cannot be directly compared. Due to this evaluation, T value confidence intervals were constructed with a significance level of 0.05 and a T value of 1.96 for each route. Table

4.4 shows the confidence intervals (CI) for each route. When taking into account the CV and CI for each route, longer routes appear to be more accurate than shorter routes. For example, the longest route has an MAE of 0.85 days and RMSE of 1.08 days, with a standard deviation of 15.2 and an average transit time of 2 days, while the shortest route has a MAE of 0.56 days and RMSE of 0.75 days with a standard deviation of 0.7 days and an average transit time of 0.7 days. Also, the longest route has a CI of 1.29 to 2.71 and the shortest route has a CI of 0.67 to 0.73. This variation is due to the number of training shipments and the distance of each route. The model obtains higher accuracy when training with more shipments. Longer routes also have an extended transit time range, and therefore their ensemble model includes more classifiers.

Table 4.3.: Coefficient of Variation for Each Route.

Route	**CV**
A	100.00%
B	87.50%
C	435.5%
D	760%

Table 4.4.: Confidence Intervals for Each Route.

Route	**Confidence Interval**
A	(0.67, 0.73)
B	(0.76, 0.84)
C	(1.25, 1.95)
D	(1.29, 2.71)

For further comparison, z-score values were calculated and p-values were calculated. These values are displayed in Table 4.5. One-tailed tests were conducted to test the confidence in the resulting MAE values. A significance level of 0.05 was used. The p values for routes A, B, and C are not significant, instilling confidence that the MAE values are equivalent to their current values. The p value for route D is significant. This significance indicates that the MAE for route D may be less than 0.85.

Table 4.5.: Z-Score and p Value for Each Route.

Route	**Z-Score**	**p Value**
A	-0.20	0.42
B	-0.47	0.32
C	-0.12	0.45
D	-7.18	< 0.00001

Each route has a different number of classifiers that train and test the data. Table 4.6 shows the number of classifiers created for each route. Routes A and B use three classifiers and routes C and D use five. Table 4.7 depicts the number of classifiers rows marked as true for each test case.

Table 4.6.: Number of Classifiers Created for Each Route.

Route	**Number of Classifiers**
A	3
B	3
C	5
D	5

Table 4.7.: Number of Classifiers Marked True for All Test Shipments on Each Route.

Route	0	1	2	3	4	5
A	35	172	94	20		
B	38	76	28	0		
C	51	120	70	11	2	0
D	11	102	167	71	13	0

4.1.1 Feature Engineering

The feature representation and selection are essential design characteristics of the model. For instance, we explored the different representations of the shipment date *Sdate*, namely, day of the week, day of the month, and day of the year. The day of the month representation resulted in the lowest MAE values as shown in Table 4.8.

Table 4.8.: MAE for the Different Date Tests.

Route	Day of Week	Day Of Month	Day Of Year
A	0.57	0.56	0.60
B	0.57	0.47	0.49
C	1.02	0.74	0.88
D	1.13	0.85	0.94

To analyze each feature's impact on the model's performance, a wrapper-based reduction approach, similar to the one presented in [35], was used. This method is an iterative approach that removes an input feature on each iteration. After eliminating the feature, the model is re-run with the remaining features. Next, the application compares the MAE from the previous iteration to the new MAE. The model retains a feature if the new MAE is higher than the previous MAE. Otherwise, the feature is eliminated. Table 4.9 illustrates this approach for route A. A one indicates that the parameter is present, and a zero indicates that the parameter is omitted from the model. Limitations include predictions converging to only the average transit time when the model has only a single input. When a single input is used, the model cannot accurately predict transit time. Due to this, more of the classifiers predict false for the test data. When there are no true entries for a classifier, the average transit time is used as the prediction. This prediction method causes the model to converge towards the average transit time when a single parameter is used.

Additional tests are conducted using the important features found using the wrapper-based reduction approach to overcome this limitation. These tests consist of the essential features with and without the *road* and *event* parameters because these appear at the end of the wrapper-based approach. When the MAE is higher without the feature, the feature is classified as important and included in the reduced model. Table 4.9 indicates that for route A, *DelivPriority, DelvItem, DangGood, Sdate, Tmax, Tmin,* and *event* are potential important features; However, after additional tests are conducted, the reduced model for A without the *road* feature results in an MAE of 0.39. When adding the *road* parameter to the model, the MAE increases to 0.41, indicating that *road* is not an important feature for the model. The reduced model of A with the *event* feature results in a MAE of 0.50, while excluding the *event* feature results in a MAE of 0.39. Therefore, the final reduced model of A does not include the features *event* and *road.*

The above approach helped identify a minimal model for each route. Table 4.10 shows the retained features for all the routes. The MAE values in Table 4.10 are lower than those of the full model shown in Table 4.2, with an exception for route G where the MAE is 0.3 higher than the MAE of the full model. The RMSE for routes A and D are less than the RMSE of the full model. Routes B and C have a RMSE greater than those of the full model by 0.07 and 0.18, respectively.

Table 4.9.: Wrapper-Based Reduction Approach on Route A.

MAE	Ship Type	Deliv Priority	Delv Item	carrier	Dang Good	Loading Point	Tmax	Tmin	Rmin	Rmax	SMax	SMin	Sdate	traffic	accident	event	road
0.56	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.56	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0.61	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.53	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.56	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0.55	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0.60	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0.57	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0.55	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0.57	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.59	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0.53	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0.56	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0.57	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0.40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 4.10.: Reduced Model Results.

Route	**Twitter**	**Weather**	**Supplier**	**Reduced MAE**	**Reduced RMSE**	**Full MAE**	**Full RMSE**
A		*Rmin, Rmax, Smin*	*Sdate, DelivPriority, Carrier, LoadingPoint*	0.39	0.64	0.56	0.75
B	*traffic, event*	*Tmin, Rmin, Smax, Smin*	*ShipType, DelivItem, Carrier, LoadingPoint*	0.43	0.73	0.47	0.66
C	*accident, event*	*Tmax, Rmin, Rmax, Smax*	*ShipType, Carrier, LoadingPoint, Sdate*	1.04	1.25	0.74	1.07
D	*accident, road*	*Tmax, Tmin, Rmin, Rmax*	*Sdate, DelivPriority, DelvItem, Carrier, DangGood*	0.78	1.05	0.85	1.08

The heuristic used to eliminate the feature is not optimal in all cases. For example, the eliminated features for routes A, B, and D were appropriated; However, the features for route C may not have been since the reduced model has a higher MAE than the full model. Table 4.10 shows that the Twitter$^{\text{TM}}$ features are not significant for route A and that the supplier and weather features are significant. For routes B, C, and D, features from the three different data categories are significant.

The three sources of features (Twitter^{TM}, weather, and supplier) suffer from missing and noisy data. Twitter^{TM} data is very sparse for shorter routes and some geographical regions. The majority of tweets were posted by a limited number of users that focus on monitoring traffic in their geographical location. Table 4.11 illustrates the number of tweets for each route on 10/16/2018. Route A and B both have a low number of Tweets. This similarity could be due to the fact that routes A and B start at the same source and follow similar paths. Route C has an even smaller number of Tweets. Route D, one of the longer routes, as expected, has the highest number of Tweets.

Table 4.11.: Twitter Counts on 10/16/2018.

Route	**Accident**	**Event**	**Traffic**	**Road**
A	4	13	15	3
B	37	21	26	23
C	35	17	0	0
D	279	79	358	41

Weather information for individual stations along routes can also be missing when stations do not report information for a given day. The distribution of missing data cannot be calculated because NOAA ^{TM} represents missing data as zeroes. Differentiation between zero values that are real zeroes and zero values that are missing data is not possible; Tables 4.12 AND 4.13 display the number of rows that contain 0 for training and testing files for each route.

Finally, supplier data can also include entry errors. Moreover, since a model is developed for each route, there may be little variation for some of the features. For example, *DelivItem* and *DangGood* are all constant for route A, and these features were not retained in the model (Table 4.10).

Table 4.12.: Training Data Rows that Contain 0 for at Least 1 Weather Parameter.

Route	Tmin	Tmax	Rmin	Rmax	Smin	Smax	Total Rows
A	0	0	648	648	1786	1786	1788
B	0	0	379	379	936	936	936
C	15	0	689	689	1259	1259	1259
D	0	0	158	158	1143	1143	1143

Table 4.13.: Testing Data Rows that Contain 0 for at Least 1 Weather Parameter.

Route	Tmin	Tmax	Rmin	Rmax	Smin	Smax	Total Rows
A	0	0	142	142	296	296	296
B	0	0	53	53	127	127	127
C	12	0	123	123	254	254	254
D	0	0	44	44	255	255	264

4.1.2 Prediction Horizon

The proposed model has a prediction horizon of 1 to 7 days as a result of the limitations on the weather forecasting horizon. That said, weather information can be forecasted up to fifteen days out from other weather sources such as AccuWeather [36]. Moreover, traffic data from Twitter^{TM} posts daily and thus may or may not be relevant for future dates. In fact, social media data may become less applicable for extended prediction horizons. Because of these reasons, transit time estimates should improve in accuracy as the shipment execution date approaches. Therefore, the model should be applied to future shipments during the planning phase daily until the shipment execution date. The weather forecasting information is not archived in NOAA^{TM} and actual weather data of 1 to 7 days prior to the shipment date is not an adequate replacement. Thus, the prediction horizon does not include changing weather data.

Table 4.14.: MAE of Models with Extended Prediction Horizon.

Horizon	**A**	**B**	**B reduced**
0	0.56	0.47	0.43
1	0.57	0.42	0.42
2	0.59	0.49	0.46
3	0.53	0.48	0.46
4	0.57	0.44	0.45
5	0.55	0.46	0.49
6	0.59	0.47	0.44
7	0.59	0.43	0.48

The full models for route A and route B and the reduced model for B ran using Twitter TM for data from 1 to 7 days prior to the shipment date was used to demonstrate the performance of the model over an extended horizon prediction. The reduced model for A was not included because it does not have any TwitterTM inputs as shown in Table 4.10.

The MAE values for the prediction horizon tests ranged between 0.53 to 0.59 for route A, and 0.42 to 0.49 for route B. Table 4.10 shows that for route A Twitter TM features are not relevant, while for route B *traffic* and *event* are essential features. The MAE for the prediction models do not consistently follow a trend. Therefore, the results are inconclusive and require future investigation.

The reduced model of route B has an increase MAE for all prediction horizon days with an exception for day one. The values range between 0.42 and 0.49. This model performs as expected. The performance degradation of routes A and B is likely due to the extended prediction horizon and availability of some of the relevant Twitter TM data.

4.1.3 Enhanced Model

An enhanced model for route D using historical traffic data from INRIX^{TM} [4] was developed. Historical INRIX^{TM} data was available for 2018 and 2019 and includes the fields shown in Table 4.15.

Table 4.15.: INRIX^{TM} Parameters.

Field	**Description**	**Example**
Severity	Incident Severity	4
Incident_obstruct	Total number of incidents involving an obstructed roadway	9
Incident_construct	Total number of incidents involving construction	5
Incident_accident	Total number of accidents along the route	7
Speed	Average historical speed for a route on a given day	64

Table 4.16.: Shipment Record Counts for Models with INRIX^{TM} Features.

	Number of training shipments	**Number of testing shipments**
With Speed	165	38
Without Speed	781	364

Average historical speed was not available for a portion of the date range. Table 4.16 shows the number of training and testing cases with and without speed. Due to the lack of training and testing speed data, the speed model was not attempted. In order to compare the INRIX enhanced models with the baseline models, the range of training data for all models was limited to 2018 and the testing data was limited to 2019.

Table 4.17 shows the MAE values for the models over this reduced date range. This table shows that the enhanced INRIX^{TM} models have higher performance. For the baseline full feature model the MAE decreases by 0.07 and the RMSE decreases by 0.03. For the reduced feature model MAE decreases by 0.14 and the RMSE decreases by 0.07.

Feature reduction was performed on the new enhanced model after the addition of the INRIX data using the same methodology described in Section 4.1.1.Table 4.10 shows that the features of the reduced models with and without INRIX data are different. The MAE and RMSE for the new INRIX reduced feature model are less than the MAE and RMSE for the original reduced feature model for route D. The new reduced feature model has the same MAE as the baseline reduced feature model, but the RMSE is 0.11 less than the the RMSE of the original reduced feature model. This indicates that the new reduced feature model outperforms both the original reduced feature models with and without INRIX data.

Table 4.17.: MAE of the Models With and Without INRIX^{TM} Data.

Model	**MAE**	**RMSE**
Baseline Feature Model	0.88	1.15
Baseline + INRIX^{TM}	0.81	1.12
Baseline Reduced Feature Model	0.94	1.14
Baseline Reduced Feature Model + INRIX^{TM}	0.80	1.07

Table 4.18.: Reduced Feature Models With and Without INRIX^{TM} Data.

Route	**Twitter**	**Weather**	**Supplier**	**INRIX**	**MAE**	**RMSE**
Baseline Reduced Feature Model	*accident,* *road*	*Tmax,* *Tmin,* *Rmin,* *Rmax*	*Sdate,* *DelivPriority,* *DelvItem,* *Carrier,* *DangGood*		0.94	1.14
Baseline Reduced Feature Model + INRIX^{TM}	*event*	*Rmin,* *Smax*	*Sdate,* *DelvItem,* *Carrier,* *LoadingPoint*	*Incident_construct,* *Incident_accident*	0.80	0.96

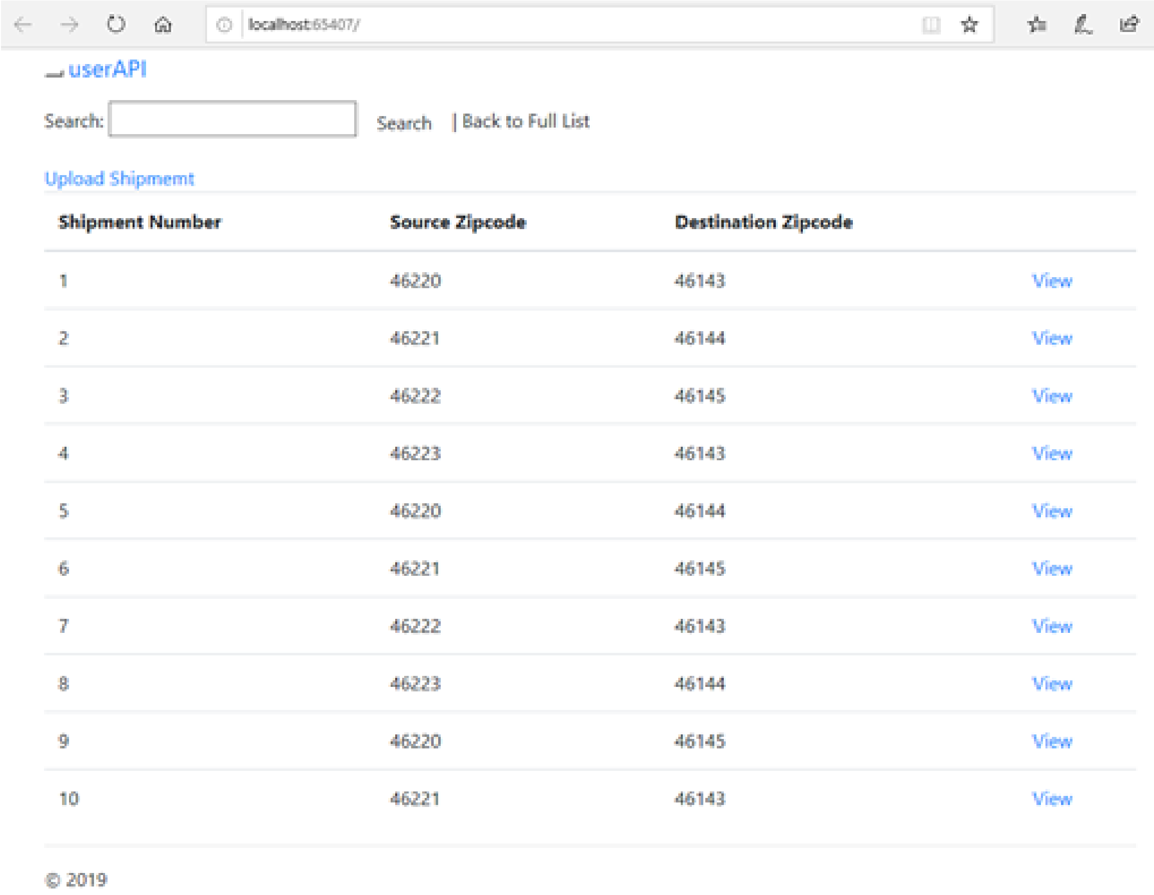
5. APPLICATION

An application that facilitate the development and use of the predictive transit model in production was developed. The application consists of two components: User Application and Administrator Application.

5.1 User Application

The user application provides the shipment planner with easy access to estimated transit times. The estimate is specific to each shipment and includes additional information such as source, destination, day of the month, product type, as shown in Figure 5.1. The accuracy of the information entered by the shipment planner has a significant impact on the accuracy of the estimated transit time. Moreover, the accuracy of the estimate improves as the shipment date approaches.

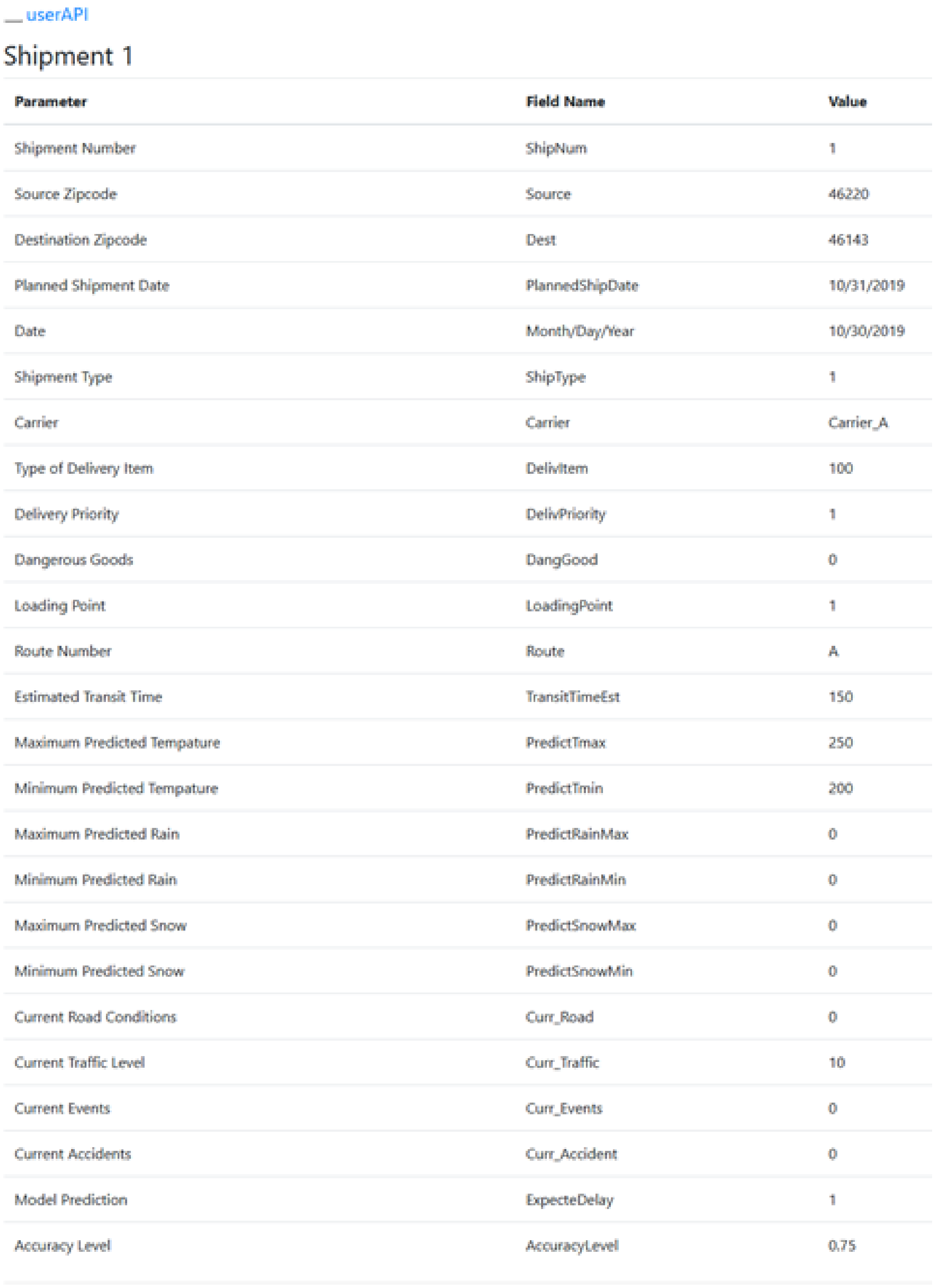
The user application view shows all of the shipments that are currently being scheduled. This application has two main functionalities. The first displays the current transit time estimates for planned shipments and those that were recently uploaded. The second allows the user to upload a new set of shipments and calculates the corresponding transit time estimates. The interface in Figure 5.1 shows the unique identification number of the shipment and the 5-digit zip codes of the source and destination of the shipment. Other parameters for a given shipment can be viewed by expanding the view option, as shown in the example in Figure 5.2.



Shipment Number	Source Zipcode	Destination Zipcode	
1	46220	46143	View
2	46221	46144	View
3	46222	46145	View
4	46223	46143	View
5	46220	46144	View
6	46221	46145	View
7	46222	46143	View
8	46223	46144	View
9	46220	46145	View
10	46221	46143	View

© 2019

Fig. 5.1.: User Application Interface.



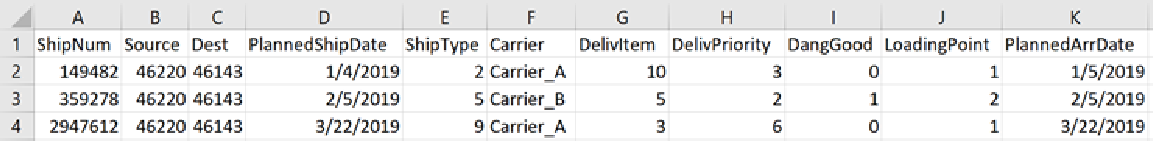
__ userAPI

Shipment 1

Parameter	Field Name	Value
Shipment Number	ShipNum	1
Source Zipcode	Source	46220
Destination Zipcode	Dest	46143
Planned Shipment Date	PlannedShipDate	10/31/2019
Date	Month/Day/Year	10/30/2019
Shipment Type	ShipType	1
Carrier	Carrier	Carrier_A
Type of Delivery Item	DelivItem	100
Delivery Priority	DelivPriority	1
Dangerous Goods	DangGood	0
Loading Point	LoadingPoint	1
Route Number	Route	A
Estimated Transit Time	TransitTimeEst	150
Maximum Predicted Tempature	PredictTmax	250
Minimum Predicted Tempature	PredictTmin	200
Maximum Predicted Rain	PredictRainMax	0
Minimum Predicted Rain	PredictRainMin	0
Maximum Predicted Snow	PredictSnowMax	0
Minimum Predicted Snow	PredictSnowMin	0
Current Road Conditions	Curr_Road	0
Current Traffic Level	Curr_Traffic	10
Current Events	Curr_Events	0
Current Accidents	Curr_Accident	0
Model Prediction	ExpecteDelay	1
Accuracy Level	AccuracyLevel	0.75

Fig. 5.2.: Detail Shipment View in User Application.

To upload new shipments, the shipment planner prepares a shipment file, as shown in Figure 5.3. The attributes of the shipment file are *shipNum, Source, Dest, Sdate, ShipType, Carrier, DelivItem, DelivPriority, DangGood,* and *LoadingPoint*. Table 3.2 includes the definitions of these parameters. Once uploaded, the model corresponding to the route associated with each shipment is invoked, and the model calculates the estimated transit time.



	A	B	C	D	E	F	G	H	I	J	K
1	ShipNum	Source	Dest	PlannedShipDate	ShipType	Carrier	DelivItem	DelivPriority	DangGood	LoadingPoint	PlannedArrDate
2	149482	46220	46143	1/4/2019	2	Carrier_A	10	3	0	1	1/5/2019
3	359278	46220	46143	2/5/2019	5	Carrier_B	5	2	1	2	2/5/2019
4	2947612	46220	46143	3/22/2019	9	Carrier_A	3	6	0	1	3/22/2019

Fig. 5.3.: Example New Shipments.

5.2 Administrator Application

The administrator application allows for the training/testing of models, historical data upload, and the addition of a new route. For the user application to generate transit time estimates for new shipments, an administrator must train the corresponding route model. Figure 5.4 shows the user interface of the administrator applications, which facilitates this process.

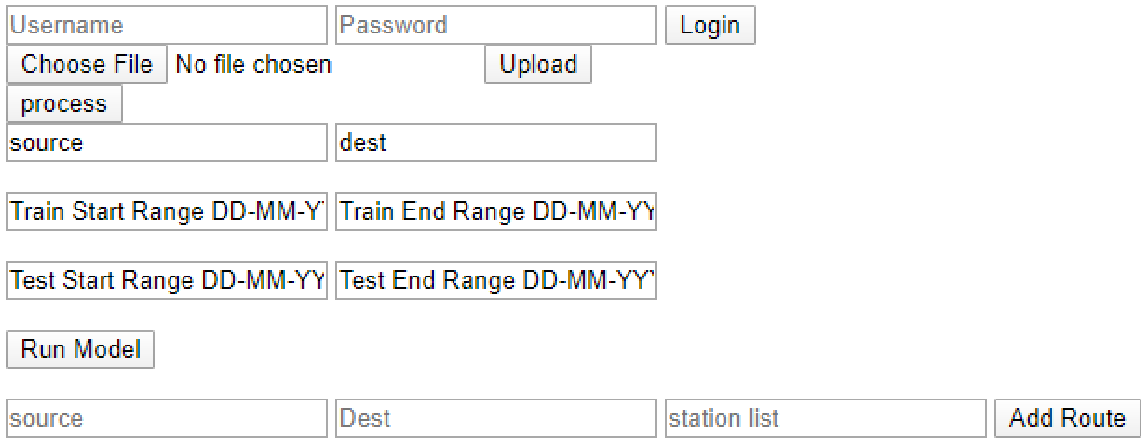


Fig. 5.4.: Administrator Application User Interface.

To add a route to the database, the administrator enters the source and destination zip codes in the first two fields and the weather station list into the third field and then selects the add route button.

Similarly, to develop a model for a target route, the administrator enters the source and destination of the route into the first two fields in Figure 5.4. The administrator must provide the training and testing ranges. Clicking on the "Run Model" invokes the SVM classifier to train the model using the shipments in the specified training range. Once training completes, the model is then applied to the shipments in the specified test range, as illustrated in Figure 5.4.

Historical data can be uploaded into the database by having an administrator choose a file and selecting upload (Figure 5.4). A list of the routes associated with the shipments in the uploaded file that exist in the route table is displayed, enabling the administrator to select the target routes. Historical shipments over the selected routes are then uploaded into the database.

6. CONCLUSION

Many commercial solutions exist for tracking shipments. However, this thesis expands supply chain capabilities by proposing a method for planning and predicting shipping delay with a one to seven-day prediction horizon. The existing commercial solutions, except for Emicen [6], rely on real-time data. Emicen [6] predicts delays in future inbound shipments based on the distribution of historical data. This thesis focuses in on predicting transit delays for future outbound shipments.

The proposed approach was applied to four different routes located in the United States. A database holds the historical information related to the four selected routes, including Twitter, weather, and shipment data. A model was developed and ran for each route. Each route is associated with a varying number of classifiers where each classifier predicts a specific transit time in number of days. Longer routes were more accurate than shorter routes when taking into consideration the standard deviation, average transit time, CV, and CI.

Reduced feature models were also developed in order to gain a better understanding of the contribution of each feature towards transit time prediction. Feature selection is iterative. A single feature is removed, and a new model is developed after each iteration. If the MAE increases, then the feature is important in the model. If it decreases, the feature is excluded from the reduced model. The impact of features on different models varies.

The prediction horizon for the models was varied from 1 to 7 days. The results were inconclusive and require additional investigation.

An enhanced model was developed for route D using historical traffic data from INRIX^{TM}. The MAE and RMSE of the the baseline INRIX^{TM} model outperformed the baseline feature model by 0.7 and 0.03, respectively. The MAE of the baseline reduced feature INRIX^{TM} model outperformed the baseline reduced feature model by 0.14. The baseline reduced feature INRIX model RMSE outperformed the baseline reduced feature model by 0.07.

User and administrator applications were developed to provide an interface for shipment planners to use the model and analyze the resulting data. The user application is for future shipment planning and provides estimated transit times. The administrator application allows users to add additional routes to the database. It also provides a way to upload historical data. The administrator application is used to train and test new models.

There are several directions for future work. First, transforming data from social media platforms into more accurate traffic predictors can help improve the accuracy of the proposed transit model. An enhanced extraction approach that relies on a semantic classifier that can more accurately evaluate the relevance of each tweet to the transit time for a given route is needed. Including data from multiple social media platforms can help improve the predictors for short routes.

REFERENCES

REFERENCES

[1] T. Erkan, E. Sancak, E. Yildirimand, and F. S. Salman, “Manufacturing parts sourcing with delayed transportation policy,” in *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, 2007, pp. 950–954.

[2] Kuebix, “Kuebix transportation management system,” https://www.kuebix.com, 2019, Last Date Accessed: 04/04/2020.

[3] PCMiler, “Commercial truck routing and mileage software,” https://www.pcmiler.com, 2019, Last Date Accessed: 04/04/2020.

[4] Inrix(n.d.), “Intelligence that moves the world,” https://inrix.com/, 2020, Last Date Accessed: 04/04/2020.

[5] FourKites, https://www.fourkites.com, 2020, Last Date Accessed: 04/04/2020.

[6] Emcien, “On-time deliveries - emcien,” Leading Perspective Analytics & Predictive Analytics Software: https://emcien.com/solutions/supply-chain/, 2017, Last Date Accessed: 04/04/2020.

[7] S. I.-J. Chien, Y. Ding, and C. Wei, “Dynamic bus arrival time prediction with artificial neural networks,” *Journal of Transportation Engineering*, vol. 128, no. 5, pp. 429–438, 2002.

[8] H.-E. LIN and R. Zito, “A review of travel-time prediction in transport and logistics,” *Proceedings of the Eastern Asia Society for Transportation Studies*, vol. 5, January 2005.

[9] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, “Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway,” *Transportation Research Record*, vol. 1855, January 2003.

[10] C. Tan, S. Park, H. Liu, Q. Xu, and P. Lau, “Prediction of transit vehicle arrival time for signal priority control: Algorithm and performance,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 4, pp. 688–696, December 2008.

[11] J. Zhang, L. Yan, Y. Han, and J. Zhang, “Study on the prediction model of bus arrival time,” in *2009 International Conference on Management and Service Science*, September 2009, pp. 1–3.

[12] Z. He, H. Yu, Y. Du, and J. Wang, “Svm based multi-index evaluation for bus arrival time prediction,” in *2013 International Conference on ICT Convergence (ICTC)*, October 2013, pp. 86–90.

[13] L. Jianmei, C. Dongmei, L. FengXi, H. Qingwen, C. Siru, Z. Lingqiu, and C. Min, "A bus arrival time prediction method based on gps position and real-time traffic flow," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, November 2017, pp. 178–184.

[14] C. Lam, B. Ng, and S. H. Leong, "Prediction of bus arrival time using real-time on-line bus locations," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, October 2019, pp. 473–478.

[15] B. Mrówczyńska, K. Łachacz, T. Haniszewski, and A. Sładkowski, "A comparison of forecasting the results of road transportation needs," *Transport*, vol. 27, no. 1, pp. 73–78, 2012.

[16] S. Bakhtyar and L. Henesey, "Freight transport prediction using electronic waybills and machine learning," in *Proceedings 2014 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS)*. IEEE, 2014, pp. 128–133.

[17] B. Mrowczynska, M. Ciesla, A. Krol, and A. Sladkowski, "Application of artificial intelligence in prediction of road freight transportation," *Promet-Traffic&Transportation*, vol. 29, no. 4, pp. 363–370, 2017.

[18] Y. Chen, Y. Chen, and B. Yu, "Speed distribution prediction of freight vehicles on mountainous freeway using deep learning methods," *Journal of Advanced Transportation*, vol. 2020, 2020.

[19] X. Li and R. Bai, "Freight vehicle travel time prediction using gradient boosting regression tree," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 1010–1015.

[20] N. H. M. Salleh, R. Riahi, Z. Yang, and J. Wang, "Predicting a containership's arrival punctuality in liner operations by using a fuzzy rule-based bayesian network (frbbn)," *The Asian Journal of Shipping and Logistics*, vol. 33, pp. 95–104, July 2017.

[21] Developers(n.d.), developer.twitter.com, Last Date Accessed: 04/04/2020.

[22] Developers(n.d.), National Oceanic and Atmospheric Administration, Last Date Accessed: 04/04/2020.

[23] Developers(n.d.), api.weather.gov, Last Date Accessed: 04/04/2020.

[24] Developers(n.d.), "Geocoding USA & Canada since 2005," geocode.ca, Last Date Accessed: 04/04/2020.

[25] Developers(n.d.), "open source routing machine." project-osrm.org, Last Date Accessed: 04/04/2020.

[26] Developers(n.d.), "Overpass turbo," https://overpass-turbo.eu/, Last Date Accessed: 04/04/2020.

[27] J. Henrique and D. Mottl, "Getoldtweets3," https://github.com/Mottl/GetOldTweets3, Last Date Accessed: 04/04/2020.

[28] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *Proceedings, First International Conference on Knowledge Discovery & Data Mining, Menlo Park.* AAAI Press, 1995, pp. 252–257.

[29] A. Kowalczyk, *Support Vector Machines Succinctly.* Syncfusion, October 2017.

[30] Developers(n.d.), "numpy.linalg.norm," numpy.org, Last Date Accessed: 05/31/2020.

[31] L. V. Martin Andersen, Joachim Dahl, "Cone programming," cvxopt.org, Last Date Accessed: 05/31/2020.

[32] S. S. Nikam, "A comparative study of classification techniques in data mining algorithms," *Oriental journal of computer science & technology*, vol. 8, no. 1, pp. 13–19, 2015.

[33] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," vol. 2049, 01 2001, pp. 249–257.

[34] Y. Tian, Y. Shi, and X. Liu, "Recent advances on support vector machines research," *Technological and Economic Development of Economy*, vol. 18, 03 2012.

[35] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," *2014 Science and Information Conference*, pp. 372–378, 2014.

[36] Developers(n.d.), https://developer.accuweather.com/, Last Date Accessed: 05/11/2020.