Enhancing Insights into Traffic Flows and Activities: Evaluating and Exploiting Machine Learning Algorithms in Real-World Scenarios

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Diyi Liu
August 2024

Copyright © 2024 by Diyi Liu
All rights reserved.

DEDICATION

This dissertation is dedicated to my parents, Yuanyuan Zhou and Shouyuan Liu, and my friend Yuening Li for their long-time love and support

ACKNOWLEDGEMENTS

I would like to express my gratitude to everyone who helped me during my student career. I would like to thank Dr. Lee D Han for his support during my PhD program. He gives me the freedom to explore whatever I want to explore. Being a student under his guidance is one of the best things I can imagine for my research career, although I doubt whether I have fully taken advantage of that. His thinking approach to science and research, similar to that of Leonardo Da Vinci or Richard Feynman, helps me further consolidate my thinking approach from many perspectives.

I also appreciate Dr. Hyeonsup Lim at Oak Ridge National Laboratory for his extended support. Overall, I am lazy and try to avoid tedious research ideas. He often described ideas that looked either tedious or were not likely to work immediately. Still, in the end, the method becomes sound and proper after I implement and experiment with it. Without him discussing those ideas, I would not have been willing to try them from the beginning. I would like to thank Dr. Asad Khattak and Dr. Hairong Qi, the other two committee members, for spending precious time helping with my dissertation. I would like to thank all my teammates, including Yangsong Gu, Hairuilong Zhang, Ruqing Huang, Lanmin Liu, Yuandong Liu, and Mohammad Khojastehpour, for working with me during my Ph.D. career.

I would like to express my gratitude to everyone who helped me during my career. I have met lots of scholars, and many of them are lovely people. They taught me skills and shared their ideas, but, likely, a student like me may never have time to direct feedback and support. There are many people I feel I own a lot. I would like to thank Dr. Joan Walker, Dr. Alexander Skabardonis, and Dr. Raja Sengupta. These professors have hugely supported me during my stay at UC Berkeley. I would also like to thank Dr. Mark Hansen and Dr. Michael Cassidy for the bits of knowledge they shared with me about transportation and economics. I would like to thank Dr. Ronald Neath from Columbia University for sharing his knowledge and supporting me. I would like to thank Dr. Randall Guensler for his support during my stay at Georgia Institute of Technology. Finally, I would like to thank Dr. Santanu Dey and several other professors who took me on the journey of Linear Programming and Combinatorics. Overall, I am a greedy person trying to absorb ideas from all over the place, and these great professors have all shaped my view of the world and research more or less during my student career.

Finally, I would like to thank those people from whom I get help. These include all the contributors from the internet (e.g., GitHub, StackOverflow), many book authors, hardware and software builders, and all other people who supported my study and research in this world. The only way I can give back is to pass my knowledge and insights to more people. Another way to give back is to help other people you know in your daily life. Indeed, seeking knowledge is not the only purpose of life.

ABSTRACT

Understanding truck activities has become increasingly crucial in traffic research, considering the increase in electric vehicles, the potential failure of critical infrastructure, etc. There are many different data sources to monitor the traffic flow. In this study, four different data sets generated from different approaches are used to extract traffic information. An innovative approach is devised and implemented for each data set to get valuable insights. Chapter I improves a recent Linear Programming method to tackle the truck identification problem based on the results of the radar detector or its equivalents (e.g., single loop detector). Tested under different contexts, the method is simple yet stable. It outperforms the two classical methods in many cases. After discussing identifying trucks for the most equipped radar detectors, Chapter II focuses on extracting traffic information from CCTV cameras, another popular traffic detector. The problem is challenging because CCTV video's resolution can be very low. To solve the problem, a framework with four modules is proposed to tackle four different subtasks, including vehicle detection, classification, tracking, and information aggregation. Chapter III tries to impute traffic flow by vehicle types over the national highway network. A new method is proposed by interactively imputing traffic information over the network and generating traffic volume data for each vehicle class under the prior assumption of Annual Average Daily Traffic (AADT) flow is known. Chapter IV uses survey data to understand the shipper's decision. The survey data, namely the Commodity Flow Survey (CFS), is collected from real freight choice of shippers. Besides applying machine learning methods to predict the freight mode choice, ensemble techniques are used to achieve better prediction performance. SHAP values are computed to interpret the machine learning model. Four chapters tackle the problem of understanding truck traffic from different perspectives. The first two chapters focus on improving local estimates, which mainly benefit the truck estimated at corridor or metropolitan levels. The last two chapters generate the whole picture of truck flows at a state or national level, a critical ability for infrastructure planning, sustainability analysis, emission modeling, etc.

Keywords: *Truck Flow Estimation, Vehicle identification, Traffic Flow Imputation, Machine Learning, Freight Mode Choice.*

TABLE OF CONTENTS

INTRODUCTION 1
Motivations 2
Data 3
Single Loop Detector and Radar Detector 3
CCTV cameras 3
TMAS data set 3
Commodity Flow Survey 4
Miscellaneous datasets 4
Dissertation Structure 4
CHAPTER I 4
CHAPTER II 5
CHAPTER III 5
CHAPTER IV 5
Relationships between different chapters 5
CHAPTER I. ESTIMATING TRUCK VOLUME FROM RADAR DETECTORS CONSIDERING TRAFFIC HEURISTICS: A LINEAR PROGRAMMING APPROACH 8
Abstract 9
Introduction 9
Literature Review 10
Methodology 12
Empirical knowledge to exploit in the problem 12
Formulating The Linear Programming Problem 15
Solving the LP Problem at scale 16
Evaluation Metrics 18
Experiments 18
A quick summary of the benefits of the precompute strategy 20
Evaluate LTs estimation by comparing against WIM ground truth 20
Evaluate the LP algorithm by comparing against dual loop detector 21
Evaluate the LP algorithm with I-24 Motion data in Nashville, Tennessee 27
Conclusion 27
References 32
Appendix 33
A: A probability analysis of corner cases 33
CHAPTER II. DETECTING TRAFFIC INFORMATION FROM LOW-RESOLUTION VIDEOS COMBINING MULTIPLE DEEP LEARNING ALGORITHMS 35
Abstract 36
Introduction 37
Literature Review 37
Methods 40
Overall Framework 40

Training YOLO model over low-resolution data given samples generated by the pretrained YOLO (teacher-student model) 40
Learning vehicle image representations using self-supervised model 42
Training prediction classifiers as prediction heads 42
From Raw observation to transportation insights 44
Experiments 44
Experiment 1. Comparing self-supervised learning with finetuned model for classification tasks 46
Experiment 2. A qualitative study between proposed framework and pretrained YOLO 49
Experiment 3. Comparing extracted traffic information with radar detectors and NPMRDS data set 49
Conclusion 56
References 59
CHAPTER III. ANALYZING TRUCK'S ACTIVITY PATTERNS BY IMPUTING POINT DETECTORS TO TRAFFIC NETWORKS USING TMAS DATASET 61
Abstract 62
Introduction 62
Literature Review 65
Methodology 66
An overview of the data analysis framework 66
Estimating AADT values by matching HPMS links to FAF links 68
Imputing values over the network with prior weights 70
Cross validation and evaluation metrics 73
Results 75
Data preparation: data selection and data quality check 75
Impute traffic volume by each traffic class for each TMAS station 76
Breakdown traffic class distribution by different states 76
Model evaluation by visualizing imputation results 80
Analysis of truck travel pattern along highway: a preliminary case study of Savannah, Georgia 80
A further visualization of specific stations: analyzing outliers 82
Conclusion 82
References 89
CHAPTER IV. IMPROVING THE ACCURACY OF FREIGHT MODE CHOICE MODELS: A CASE STUDY USING THE 2017 CFS PUF DATA SET AND ENSEMBLE LEARNING TECHNIQUES 92
Abstract 93
Introduction 93
Literature Review 94
Explorative Data Analysis 98
Mode Classification and Variable Definitions 98
Analyzing Freight Mode Choice Patterns 99
Methodology 104

An Overview of the Proposed Methodology 104
Developing Useful Features (Data Augmentation) 104
Building Base-learner (Level-1 Model) 105
Dynamically Building Ensemble Models (Level-2 model) 108
Experiment Results 110
Model Implementation 110
Experiment Design and Model Evaluation 110
Model Results 113
Conclusions 122
Statements and Acknowledgements 124
References 125
CONCLUSION 129
VITA 130

LIST OF TABLES

Table 1. The assumptions of three different versions of LP models 17
Table 2. A summary of all parameters involved in the Linear Programming problem 17
Table 3. A comparison of computational time for three scenarios: (1) precompute all combinations; (2) case study of WIM; (2) case study of dual loop detectors..... 22
Table 4. Evaluation of the prediction error by comparing between WIM measured results and the proposed method 22
Table 5. Evaluation of the moving average speed prediction error between dual loop detector measured results and the proposed method..... 25
Table 6. Evaluation results for estimating speed and LTs using the proposed LP algorithm for morning peak traffic in I-24, Nashville..... 30
Table 7. A preliminary analysis for the probability of greater than 5 LTs appeared considering probabilities and headways 34
Table 8. A confusion matrix by different vehicle class for the proposed method..... 48
Table 9. Comparison of running speed of different algorithms (unit: frames per second)50
Table 10 A comparison of the number of large trucks (i.e., semi-trailer trucks) by applying a naïve classification algorithm. 57
Table 11. The overall structure of the network..... 67
Table 12. Aggregation options involved in the study and provided by the analysis pipeline..... 69
Table 13. A comparison of different experiments segmented by time of the day (for weekdays) 81
Table 14. A summary of cross validation experiment results of the vehicle class shares of each state..... 83
Table 15. Summary of recent comparative study comparing between machine learning algorithms and traditional methods in the application of transportation mode choice. 97
Table 16. Five aggregated freight mode classifications and more specific CFS PUF data set mode classifications..... 100
Table 17. Description of all considered information from the CFS PUF data set..... 100
Table 18. All explanatory input variables and features employed in the model..... 107
Table 19. Experiment design aspects to measure the effectiveness of three countermeasures..... 111
Table 20. Performance of unified models, SCTG models, NAICS models, and the overall voting model on the testing data set..... 112

LIST OF FIGURES

Figure 1. Relationships between different Chapters in the dissertation..... 7
Figure 2. A histogram of vehicular axle-to-axle distance with respect to different vehicle types generated by results from WIM detectors 13
Figure 3. The artificial control curve between traveling pace and occupancy (left figure). The same relationship is converted to the right figure between traveling speed and occupancy 17
Figure 4. Results of the number of LTs for any combination of observed vehicles and occupancy with respect to different configurations 19
Figure 5. Comparing LP results from Loop detector against WIM ground truth data 22
Figure 6. A visual comparison of the results between occupancy and volume for each traffic lane in the WIM example. 23
Figure 7. A comparison between dual loop detector measured speed and observed speed. The transparent corresponds to raw inputs, whereas the untransparent lines are 10 minutes moving average 24
Figure 8. A visual comparison of the results between occupancy and volume 25
Figure 9. The overview of all average speed for each observed combination of occupancy and volume 26
Figure 10. Trajectory examples and lateral distributions of I-24 Motion vehicle trajectories 28
Figure 11. A comparison between RDS estimated results and I-24 Motion vehicle trajectories 29
Figure 12. A framework to fusion YOLO and self-supervised model for information detection. 41
Figure 13. Training YOLOv8s (small student) model with custom dataset generated by pretrained YOLOv8x (large teacher) model. 41
Figure 14. A review of the core SimCLR process to generate pairs of visual representations. 43
Figure 15. Steps of converting raw trajectories to projected trajectory. 45
Figure 16. Six pairs of augmented images for self-supervised training under SimCLR model. 47
Figure 17. New classification of small vehicle (“sv”), small trucks “st”, and large truck “lt”. 47
Figure 18. A Visualization of Embeddings Learned by SimCLR model for three types of vehicles. 48
Figure 19. Extract transportation information given trajectory outputs. 48
Figure 20. A qualitative comparison between YOLOv8x algorithm (above row) and the proposed framework (below row). 50
Figure 21. Six consecutive snapshots of a traffic fleet and identification results at the same site observing vehicle types of the traffic fleet. 50
Figure 22. The study site at mile marker 380.0 at I-40 highway near Knoxville, Tennessee. 51

Figure 23. Speed comparison between proposed method, RDS detector, and NPMRDS records for eastbound and westbound at the studied site. 53
Figure 24. Comparison of Identified vehicles for east bound and west bound directions at the studied site. 54
Figure 25. A quality check for the vertical length of identified trajectories within the detection zone. 55
Figure 26. Locations of TMAS detectors for both (a) volume observation and (b) vehicle class observation. 64
Figure 27. A simple example of imputing missing traffic volume in a grid network traveling from bottom left to top right. 64
Figure 28. The proposed analysis framework for estimating vehicle volume and class distributions along traffic networks based on AADT values and TMAS datasets. .. 67
Figure 29. A comparison of two networks: HPMS and FAF. FAF networks are more comprehensive and tailored for freight traffic analysis. 67
Figure 30. A typical local representation of an interstate segment with an overpass arterial using FAF and TMAS networks. 69
Figure 31. The procedure of estimating AADT values from HPMS networks. 71
Figure 32. A typical local representation of a highway segment with an overpass arterial network. A TMAS station (the blue circle) is located along the interstate highway. 71
Figure 33. Graph representing a local road segment to demonstrate the neighboring links (in solid red) of a link of interest (in solid black). 72
Figure 34. The pseudocode of the graph algorithm running the imputation process. 75
Figure 35. Two special kinds of traffic links with special rules for imputation. 74
Figure 36. A visualization of algorithm evaluation using cross validation. 74
Figure 37. Quality check of detector availability, average truck volume (normalized between 0 and 1), and truck activity level for each month for each state. 77
Figure 38. A visualization of directional AADT of trucks for (a) Texas and (b) Georgia. The AADT values are matched to FAF links from HPMS links. 78
Figure 39. Analysis of the R^2 of detectors of traffic volume from Vehicle Classes 5–13. 78
Figure 40. Breakdown analysis for the performance of truck volume prediction for different states. 79
Figure 41. A bubble plot comparing the Pearson R^2 performance for traffic volume with the density of TMAS stations by state. 85
Figure 42. A comparison between imputed truck value per hour and HPMS AADT truck volume (normalized to hourly volume) for Georgia. 85
Figure 43. A visualization of the ratio of Classes 5 and 9 for different times of the day for weekdays and weekends in Savannah, Georgia. 86
Figure 44. An interactive map of TMAS stations visualizing truck volume and vehicle class distributions near (left) Savannah, Georgia, and (right) near Tampa, Florida. 87
Figure 45. Share of different modes with respect to the number of shipments, total weight, total value, and total ton-miles traveled 101
Figure 46. Percentage value share (in US dollars) of different freight modes by origin state 101

Figure 47. Percentage value share (in US dollars) of different freight modes by destination state 102
Figure 48. Distribution of different freight mode choices by different SCTG (commodity) codes and NAICS (industry) codes (measured in the number of shipments) 103
Figure 49. Proposed data processing pipelines and analysis framework 106
Figure 50. Collected derived distance and spatial information for each mode between each origin–destination pair of CFS areas 106
Figure 51. One possible pseudocode of steps applying a voting mechanism to obtain ensembled predictions of base learners for each record. 110
Figure 52. The basic structure of stacking different types of tree-based methods using a two-level hierarchy to obtain final ensembled predictions 109
Figure 53. Data flow of the analysis framework: data preparation and cross-validation for the two-level stacking method 109
Figure 54. Accuracy of the final model for each freight mode by SCTG code (the left column) and by NAICS code (the right column) 115
Figure 55. Evaluating performance by receiver operating characteristic's AUC for various models 116
Figure 56. Variable importance for the (left) global RF model and (right) XGBoost stacking model 117
Figure 57. A force diagram of SHAP values of one DT model over one testing data sample 118
Figure 58. (a) Summary plot of absolute SHAP values and (b–f) swarm plots of each mode choice using the SHAP game theory approach to explain feature influences on the overall RF model 120
Figure 59. SHAP dependence plot of the single weight ("sw") feature and some interaction features for each freight mode 121
Figure 60. (a) Summary plot and (b–f) swarm plots of the level-2 stacking model 123

INTRODUCTION

Motivations

Recently, truck and freight traffic has become a more and more heated topic in transportation studies for multiple reasons. First, many trucks will transfer from traditional vehicles to electric vehicles. Problems like planning traffic charging stations, evaluating pollutant decrease, and economic benefits are impossible without a better understanding of truck flow. Second, many critical transportation infrastructures are aging and becoming fragile as time goes on in the United States. For many sites, the impact of incidents on critical infrastructure will go on for months or even years. As an extreme example, the closure of Baltimore's Francis Scott key bridge in 2024 has stopped ground traffic for months. Besides losing a critical traffic link, the change of port activities caused by this event will also shift the freight or ground truck's activities. Thus, it is hard to understand the impacts well without a better understanding freight flow using a data-oriented approach. Thirdly, at a nationwide level, understanding traffic has become increasingly important in evaluating the impacts of policies related to the logistics chain. Thus, there is an immediate need to design better ways of assessing truck activities using existing data sources.

This study has four chapters that provide various perspectives on tackling the problem of understanding truck/freight activities. The first two chapters use the two most equipped traffic detectors (i.e., radar detectors and CCTV cameras) to monitor traffic at a metropolitan level. The objective is not only to provide some methods to identify truck vehicles and flows but also to provide a practical, practical application to bring results for traffic management and understanding truck traffic. While the first two chapters zoom into metropolitan-level traffic, the remaining two try to interpret truck activities at a nationwide level. The problem is challenging because the total observable information is sparse at a nationwide level. Moreover, in many cases, there is a lack of focus on understanding the state-wise or nationwide picture of traffic flows, especially considering time-varying factors. The impact of the incidents (e.g., the closure of key infrastructures) can usually affect a large portion of state-wise or nationwide activities. Thus, it is essential to design tools to evaluate traffic on state-wise or nationwide levels.

Besides designing tools to interpret traffic performance, this dissertation aims to demonstrate relatively mature methodologies that are deployable in the field to support traffic analysis. In other words, using current data sources could help understand freight or truck flows. For example, methods in Chapter I can be deployed to estimate truck flows along a chain of radar detectors separated by 0.5-1.5 miles. Chapter II tries to identify truck flows assuming a video stream with a resolution of 320×240 dimensions running at 15 frames per second, a low resolution. Experiments will show that most traffic information can be captured even at this low resolution. Compared with higher-resolution videos, there is much less of a requirement for computing resources. Chapter III provides imputation results over the national highway networks, making it possible to use them directly for emission modeling, planning of electric charging stations, etc. Finally, Chapter IV provides an accurate machine learning model for estimating freight choice for all scenarios using the open-sourced Commodity Flow Survey's Public Use File data set. Any interested person investigating the freight topic can use the trained model to predict the transportation mode for the type of products and origin-destination pair of interest. Similarly, for the methods of Chapters I-II, any interested people can easily use the proposed methods or models on their projects.

Data

The section aims to discuss the major data sets or its underneath technique involved in this dissertation.

Single Loop Detector and Radar Detector

Identifying truck flow using Single Loop detectors and Radar Detectors is the focus of Chapter I. For transportation agencies, single loop detectors are the most crucial detector. More recently, radar detectors have become popular as a substitute for the single detector. Both detectors are similar in generating traffic volume and traffic occupancy for each traffic lane every 30 seconds. These detectors are the most frequently equipped detectors in the United States for transportation agencies to monitor highway traffic. Besides the single loop detector, another type of detector, the double loop detector, can further measure average travel speed and total vehicle lengths at 30-second intervals for each traffic lane. This dissertation uses two major data sets: (1) California Caltran's PeMS data set. (2) the Tennessee DOT's Radar Detector System (RDS) data set.

CCTV cameras

Identifying traffic and truck information and truck flow using video data is the main topic of Chapter II. Besides loop or radar detectors, transportation agencies also widely adopted surveillance CCTV cameras. For many agencies, CCTV cameras can cover a large portion of the road segments along major highways or arterials. In most cases, using CCTV camera data is the only way for traffic operators to understand the situation when there is a traffic incident. The CCTV camera streaming data is also shared with the public using a low-resolution format so travelers can better understand the current traffic situation at various locations. Such information can be handy in building up an automated system that extracts traffic information and even fire alarms when incidents happen. Some Examples of such camera systems include (1) California Caltran's CCTV map (https://cwwp2.dot.ca.gov) and (2) Tennessee DOT's Smartway Traffic Maps (https://smartway.tn.gov).

TMAS data set

Traffic Monitoring and Analysis System (TMAS) data set is the main topic of Chapter III. TMAS data set is a historical data set containing thousands of detectors at fixed locations. Each year, each state is required to submit its data to TMAS to form a nationwide view of traffic. The TMAS can be decomposed into three data subsets, including (1) vehicle volume, (2) vehicle class, and (3) vehicle weights. The vehicle volume and class subsets provide traffic volume and class information, respectively. The volume data is measured by any detectors that can estimate traffic volume at a fixed location on the major highway. The vehicle class subset provides the traffic flow of the Federal Highway Administration (FHWA) Scheme F classification. The data is the total traffic flow aggregated by vehicle class and by each hour for each travel direction. Such information can be beneficial for understanding vehicle class distributions at various locations. Compared with the Annual Average Daily Traffic (AADT) value, the number is more detailed and is observed every hour of the year. The weights data subset provides information about truck weights using either the Weight In-Motion (WIM) system or traditional Weigh Stations. Chapter II focuses on the vehicle class data set with volume and flow values recorded over the year. Chapter II proposed a method to impute truck flows over the national highway

network, given that all information provided by TMAS is at fixed locations over the traffic network.

Commodity Flow Survey

Commodity Flow Survey is the main topic of Chapter IV. Commodity Flow Survey is an open-source dataset recording the choice of shippers at a disaggregate shipment level. It contains detailed information on the product's property, including a single product's weight, value, origin, destination, industry code (i.e., NAICS), commodity code (i.e., SCTG), etc. Miscellaneous information is also recorded, including whether it is temperature controlled/hazardous/exported, etc.). There are two rounds of data sets published: the 2012 and 2017 surveys. The 2017 survey has more than 6 million disaggregated shipment records. The big data nature of the data set provides a chance to understand the freight shipment choice using machine learning models by predicting the freight choice given inputs. As a reminder, the most recent survey results won't be published until 2025.

Miscellaneous datasets

The four data sets, as mentioned earlier, correspond to the main topic of the four chapters. However, using other detectors or data sets to evaluate the results of proposed methods is necessary. For example, by collecting data from vehicles' GPS records, the National Performance Management Research Data set (NPMRDS) provides a reasonable estimate of average travel speed aggregated by 10 or 15 minutes for each short traffic segment for major highways and arterials. For Chapter I, to get the ground truth of Scheme F classification results, Weigh-in-Motion detector results are used. Furthermore, to evaluate travel speed estimation results, double loop detectors are used to get the measured travel speed.

Dissertation Structure

The dissertation contains four chapters. The following section discusses each chapter separately before summarizing their connections and relationships.

CHAPTER I

Chapter I aims to find a reasonable yet simple method to estimate trucks using the single loop detector or its equivalent: the radar detector. The radar detector or single loop detector is one of the most widely equipped detectors in the United States for traffic agencies to monitor traffic. For decades, previous studies have discussed the problem of identifying trucks with various proposed algorithms. However, the previous methods are usually not accurate enough. The assumptions of previous methods cannot stand for all scenarios. In this study, we proposed a Linear Programming approach to better estimate truck flow given radar detector outputs. Instead of solving statistics fitting or optimization problems in real time, the results for all combinations are precomputed. Thus, a big look-up table is generated to query the number of trucks given occupancy and volume pair. Three experiments are designed to evaluate the performance gained from this method. The algorithm is also tested along the I-40 traffic corridor to show the consistency of the truck estimations.

CHAPTER II

Chapter II focuses on designing a better framework to identify vehicles from low-resolution CCTV cameras. For low-resolution videos, the images of vehicles are only several pixels in width and height. Thus, it becomes a very challenging problem to identify, classify, and track vehicles consistently over time. To deal with the challenge, in this chapter, we divide the problem into four isolated subtasks or modules: vehicle identification, vehicle classification, vehicle tracking, and trajectory aggregation. A module solves the problem embedded with one deep learning algorithm for each of the first three tasks. A YOLO model is retrained for vehicle identification. For classification, the SimCLR contrastive learning framework is applied to learn vehicle representations. On top of vehicle representations, a small classifier model is fitted for classification. For vehicle tracking, the ByteTrack model tracks vehicles across frames to generate traffic trajectories. Finally, after all vehicle trajectories are identified, given the perspective view of the camera, the trajectory is converted into an aerial view to estimate traffic volume, speed, etc. The experiment shows that the speed estimate is accurate, and the volume estimates are also reasonable on the near side of the camera pole.

CHAPTER III

Chapter III targeted imputing traffic flows, giving detector results at fixed locations. Specifically, results are imputed from fixed detected locations to the national highway networks. This problem is also very challenging because the total information gained is sparse compared with the entire network. A limited number of TMAS stations are used as time-varying observations. Besides, for each road, the AADT volume is applied as a prior assumption of truck volume. In this study, we propose a method to iteratively update the traffic flows by taking the averages across neighboring links in the traffic network. Results show that the performance is better than expected in many states. This elegant method can give a reasonable method to impute traffic volume by vehicular types, making it possible to break down the black box of freight movements on the national level.

CHAPTER IV

The objective of Chapter IV is to predict the freight mode choice based on the Commodity Flow Survey (CFS) data set. Previous researchers have tested different machine learning models to predict freight mode, including random forests, XGBoost, and k-NNs. The last method's performance is not good enough for interested stakeholders directly. In this study, we further investigate the possibilities of improving the performance by applying the following tricks: (1) build different models for different commodity types or industry types; (2) use the stacking method to ensemble different models to generate one output; (3) add the feature of routed distance for available freight modes as input features. By applying those three countermeasures, the prediction accuracy reaches over 92%. Furthermore, using the SHAP package over the model can help explain the model's behavior on different features' impact on different freight modes.

Relationships between different chapters

Extracting truck information is vital for ground transportation. Ideally, one can estimate the 13-class FHWA Scheme F classification results for every single location over the network at any time. However, this is not possible in the real world. Most traffic detectors are located at a fixed location along the travel network. Only a very limited number of detectors can generate vehicle

classifications. The vehicle class subset of the TMAS data set contains many results based on Scheme F classification. The detectors generating vehicle classification might be double loop detectors or Weigh-in-Motion detectors. However, single loop or radar detectors are more frequently equipped in practice. Thus, chapter I decides to use those detectors to estimate trucks. Since the information is limited, it is not possible to identify flows of 13 vehicle classes. Since most vehicles traveling on roads are either passenger cars (i.e., class 2) or semi-trailer trucks (i.e., class 9), Chapter I tries to identify the number of trucks assuming the average length of all trucks are similar to that of semi-trailer trucks (i.e., class 9). Chapter II tries to identify trucks given CCTV cameras. Since the video is low-resolution, it is also infeasible to identify all 13 vehicle classes. In the experiment in Chapter II, three classes are identified: (1) "passenger cars" (i.e., class 1-4); (2) small "truck" (i.e., class 5-7); (3) large trailer trucks (i.e., class 8-13). In the future, by applying the proposed method, it is possible to identify more vehicle classes. For example, it is possible to identify special vehicles (e.g., fire engines, police cars, USPS delivery cars, school buses, etc.). Although not aligned with the 13-class Scheme F, the potential of CCTV cameras in identifying vehicles is immense, making it possible to develop applications for traffic management.

Chapter III provides a method to efficiently impute the traffic flow to the traffic network. The imputed values can be the share or flow of each vehicle type. This method gives a spatial view of the traffic system for different vehicle classes (e.g., class 5 "delivery trucks," class 9 "semi-trailer trucks," etc). Finally, Chapter IV predicts the freight traffic mode. Among the five mentioned freight modes (i.e., for-hire truck, privately owned truck, parcel service, air, others), three of them (i.e., for-hire truck, privately owned truck, parcel service) are related to trucks. Although many parcel products may be shipped by smaller delivery trucks (e.g., class 6), most products are shipped by semi-trailer trucks (i.e., class 9). The privately owned truck may have their advertisement or logos on the back or side of trucks, making it possible to identify them using a vision-based approach. Similarly, the UPS and USPS trucks are also identifiable from images. In summary, the relationships between vehicle classes are visualized in Figure 1.

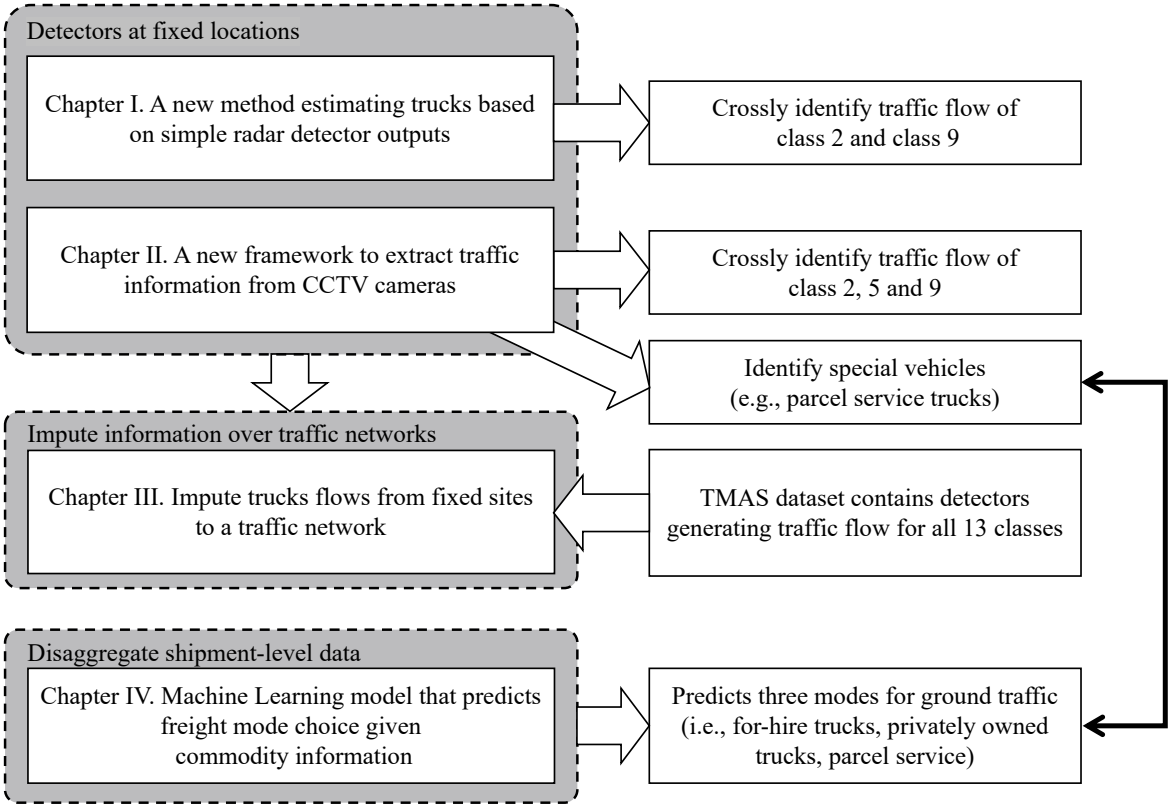


Figure 1. Relationships between different chapters in the dissertation.

CHAPTER I. ESTIMATING TRUCK VOLUME FROM RADAR DETECTORS CONSIDERING TRAFFIC HEURISTICS: A LINEAR PROGRAMMING APPROACH

Abstract

The traditional singly loop detectors or their equivalent (e.g., radar detectors) play an essential role in understanding the overall traffic. Although speed and occupancy readings are directly measured, the traffic speed outputs are estimated with the assumption of Mean Effective Vehicle Length (MEVL). This assumption leads to inaccurate estimations for both speed and long truck volume. In this study, assuming the traffic fleet is composed of two classes (i.e., Short Vehicles and Long Trucks), a Linear Programming problem is formulated based on physical constraints and heuristics to solve the best combination of Short Vehicles (SVs) and Long Trucks (LTs). Three experiments are designed to estimate the model's performance compared with measured ground truth for trucks and travel speed. Compared with the classical lane-to-lane correlation method and time-correlation method, the method has a better performance in estimating LTs in many cases. Moreover, it provides a simple but effective view of estimating LT volume based on aggregated volume and occupancy readings. By solving results for all input combinations, the method is straightforward to apply to any site given occupancy and volume records. Furthermore, a preliminary example is given to estimate truck flows over a long highway corridor. The technique can help identify truck traffic flow for analyzing truck activities, modeling traffic emissions, etc.

Introduction

For decades, the single loop detector (SLD) has become the most widely equipped detector in the United States. More recently, radar detectors become another substitute for the SLD. Unlike SLD, which embeds the conductive loop sensors within the road pavement, the radar detector is located on the side of the road, making it easier to repair or substitute the device without interfering with traffic. Despite using different sensor technology, both detectors can be viewed as the same kind of detector that only measures the total traffic volume and total traffic occupancy aggregated at 20 or 30-second intervals by each traffic lane. In this study, radar and single-loop detectors are referred to as SLD without further clarification. To be specific, traffic occupancy denotes the percentage of time during which detectors are occupied by a vehicle. The range of occupancy is between 0 and 100 to represent the percentage of time the detector is occupied. This study aggregates all records by 30-second time bins by default.

As mentioned, SLD is the most vastly equipped detector across many transportation agencies in the United States. For example, the Tennessee DOT's Radar Detector System (TN-RDS) contains around 979 radar detectors near its four major metropolitan regions (i.e., Knoxville, Chattanooga, Nashville, and Memphis). In other words, there is usually a detector for every 0.5-1 mile along the major highway corridors. Data collected by those detectors are the most critical data sources for evaluating and operating traffic because they cover most highway systems near metropolitan areas.

Although the SLD measures total volume and occupancy, the detector cannot directly measure travel speed. The total fleet length and vehicle classes are unknown without knowing the speed. Thus, an algorithm is needed to estimate LTs. Although several algorithms estimate truck flows given SLD recordings, the performance of those algorithms is not thoroughly tested under

multiple scenarios and transportation contexts. Recently, a Linear Programming (LP) approach has been formulated to solve the best combination of Short Vehicles (SVs) and Long Trucks (LTs). This idea is inspired by the observation that most vehicles traveling on highways are either small (e.g., passenger cars, small SUVs) or long trucks (e.g., trailer trucks). The previous study shows that the method is more capable of identifying LTs than those traditional methods. However, the LP method still has some limitations. One major issue is that the algorithm tends to overestimate traffic flow when the traffic is congested. In this study, on top of the former proposed method, an improved method tries to further develop the method with the following objectives:

(1) Improve the formulation of the LP problem by integrating more physical constraints and heuristics, considering headways, measurement errors, etc.
(2) Evaluate the performance of LT or speed estimations by comparing them against ground truth measurements from Weigh-in-Motion or dual loop detectors.
(3) Evaluate the performance of speed outputs by comparing them against ground truth measurements from dual loop detectors.

Literature Review

As mentioned, SLDs are the most equipped detectors in the United States. Moreover, those detectors cannot directly measure traveling speed. Thus, assumptions were made to estimate the traveling speed. The traditional approach assumes that all vehicles have a Mean Effective Vehicle Length (MEVL), giving the distribution of vehicle length a fixed number. Many researchers followed the concept of MEVL to provide a method of improving its accuracy. For example, Wang and Nihan (2000) proposed the concept of the g-factor to treat the inverse of EMVL as a function of traffic volume and occupancy. The study fits a log-linear regression model for mean vehicle length estimation. In practice, the MEVL concept is far from the actual case because the average vehicle length within 30 seconds is too volatile to be close to the average MEVL statistics in many cases.

Instead of dynamically estimating the EMVL, it is necessary to design other methods to directly estimate the number of trucks, given measurements of occupancies and volumes. One classical method is to utilize the speed correlation relationship along a sequence of readings (Wang and Nihan, 2004). Three major assumptions are made: (1) the average traveling speed is a constant number within 5 minutes; (2) there are only two types of vehicles: SVs and LTs; (3) among the 15 consequent recordings within 5 minutes (i.e., 20 seconds aggregation time), at least 2 of them are truck-free. Under these assumptions, truck volume can be estimated by comparing the ratio between volume and occupancy. In this chapter, this classical method is referred to as the Time Correlation Method (TCM).

Besides looking at the correlation in the same traffic lane over time, it is also possible to utilize it across different traffic lanes. For example, Kwon et al. (2003) proposed a method to estimate LTs based on lane-to-lane speed correlation. The algorithm assumes that the innermost lane should be a truck-free lane. Furthermore, the speeds are highly correlated across different lanes at the exact location of the detector. For example, after estimating the speed of the innermost lane, assuming all vehicles are SVs, the traveling speed of the other lanes can be considered as

95% that of the innermost lane. In this study, the method is referred to as Lane Correlation Method (LCM).

Other methods assume it is possible to get the "on-time" of every vehicle traversed on the detector. Coifman and Neelisetty (2014) proposed a method to improve highway speed estimation under traffic conditions with heavy truck traffic. Coifman and Kim (2009) propose a method to estimate vehicle length based on the moving median vehicle length and sequence methods. The moving median method is taken from the consequent readings of the last 33 vehicles. The sequence method acknowledged that most vehicles are of two classes: a passenger vehicle class centered at about 20 feet and a trailer truck centered at 70 feet. The sequence method is used by default. However, the alternative moving median method is used when the traffic is congested. In practice, most transportation agencies or vendors can't provide unaggregated records of each vehicle.

Gu et al. (2022) recently proposed an LP optimization method to estimate LTs given SLD data. The method is evaluated against three "ground truths": SUMO simulation, PeMS Weigh-in-Motion detector, and a one-hour video sample. Similar to (Kwon et al., 2003), the core idea is that vehicle length distributions are bivariant centered around 17 feet for small passenger vehicles and 70 feet for large trucks. The results for LTs are close to WIM data, although the performance was a bit off during peak hours on very congested highway roads. The method is further tested and compared against the LCM method and TCM method (Gu, 2023). A later study (Gu, 2023) shows that the LP method outperforms the other three methods in many cases.

Both former studies only test their algorithm performance on a single site by comparing it against dual loop detectors or Weigh-in-Motion data (Gu, 2023; Gu et al., 2022). Weigh-in-Motion data stations can generate estimates close to the ground truth by measuring axle weights and spacings (Hernandez et al., 2016). As technology advances, it becomes possible to compare estimations of LTs with many other new data sources. For example, Tennessee's I-24 Motion project can also generate high-quality vehicle trajectory data sets given the input of high-resolution video data and deep learning algorithms (Gloudemans et al., 2023). Data gathered from probe vehicles can be helpful for the evaluation of speed or travel time along traffic corridors. For example, the National Performance Management Research Data Set (NPMRDS) can measure the corridor travel time as ground truth data (Kaushik et al., 2015). It is also possible to self-evaluate the results by comparing across close-spaced detectors along a traffic corridor.

In summary, a comparative study shows that previous methods of identifying LTs based on SLD records are not as good as the proposed LP method in most cases (Gu, 2023). Moreover, many of those methods usually make assumptions that are only suitable under a restricted set of conditions. Some methods are difficult for practitioners because they require statistics by looking at different lanes, consecutive recordings, or raw unaggregated detector signals. Thus, the target of this study is to further develop the former LP method with the following objectives:

(1) Design an LP formulation to solve the combination of LTs and SVs considering multiple empirical knowledge.
(2) The algorithm should be simple but relatively accurate under almost all traffic conditions and road geometries.

(3) Solve all LP problems to generate a lookup table instead of solving LP problems on request.

Methodology

The methodology section introduces the method in four sub-sections. The first part talks about the five empirical assumptions to be integrated into the system. The second part formulates the problem with all constraints and the objective function. The third part builds up an efficient pipeline of estimating the approach by precomputing all scenarios. The fourth part introduces the virtual vehicle trajectory approach to generate traffic information.

Empirical knowledge to exploit in the problem

(1) The relationship between Vehicle lengths and classes

Figure 2 shows observations collected from the Weigh-in-Motion (WIM) detector for the relationship between vehicle types and axle-to-axle distance. By observations, the histogram shows a clear bimodal distribution. The first mode corresponds to cars or SUVs. The second mode is almost solely composed of trucks. Thus, SVs or LTs correspond to the number of vehicles near two modes. SV refers to passenger cars, SUVs, pickup cars, minivans, etc. LTs mainly refer to trucks, although there is a chance for other vehicles like couches to fall into this category.

Results show that most of the vehicles are either SVs or LTs. Although there are other genres of vehicles (e.g., motorcycles, school bus.), the frequency of those vehicles is lower. Thus, we assume that the vehicles are either SVs or LTs in this study. Thus, we define the following constraints:

$$13 \leq l_{SV} \leq 19 \quad (1)$$

$$65 \leq l_{LT} \leq 75 \quad (2)$$

where l_{SV} and l_{LV} corresponds to the length of SVs and LTs, respectively. The length of SVs (l_{ST}) is between 13 and 19 feet. The length of LTs is between 65 and 75 feet. The ranges of vehicle length are higher than that of Figure 2 above because the WIM measures spacings between the first axle and the last axle, whereas in our case the SLDs are measuring the distance of the entire vehicle. As an averaged value of the vehicle length, the constraining ranges are also wide to compensate for the occasional occurrence of other sized vehicles (e.g., motorcycle, school bus.) at times.

(1) Time bounds considering both occupancy and headways

Headways are defined as the seconds between two consecutive vehicles assuming a stationary traveling speed. In the real world, for daily traffic, it is impossible for a fleet of vehicles to safely form a fleet with a headway of less than 2 seconds (Vogel, 2003). Furthermore, LTs like trucks are slow to accelerate or decelerate, and experienced drivers would drive at a higher time-to-collision or headway. Another study finds that at high-capacity rate the car-car headways are around 1.9 seconds, and the car-truck or truck-car headway is over 2 seconds. Thus, we assume

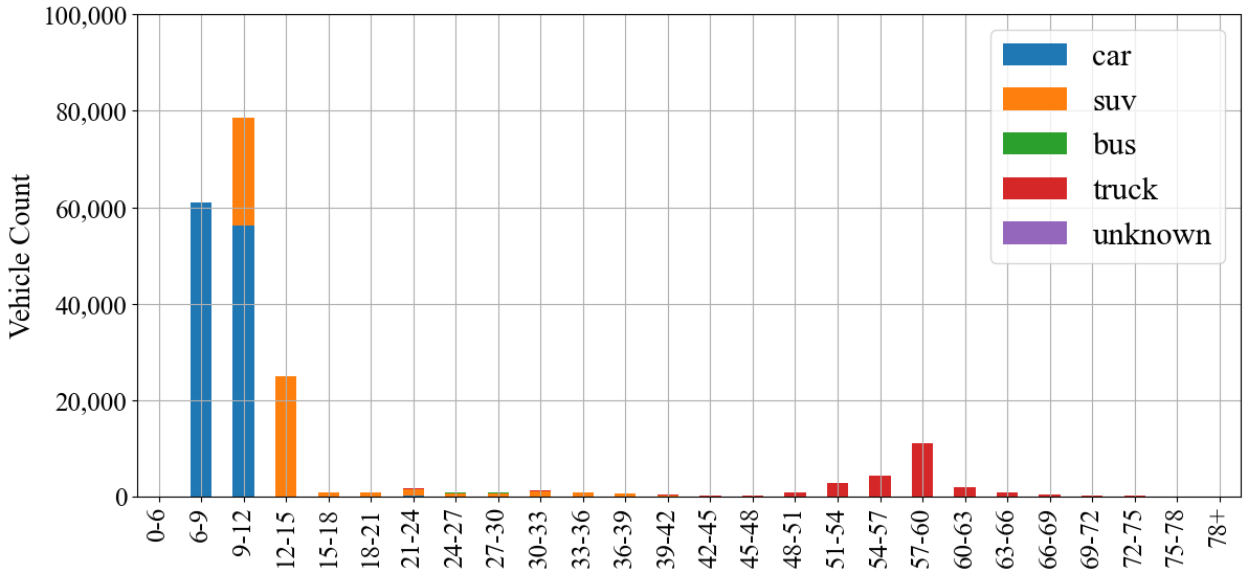


Figure 2. A histogram of vehicular axle-to-axle distance with respect to different vehicle types generated by results from WIM detectors

headways time assigned to each small vehicle are 2 seconds for SVs and 3 seconds for LTs. In other words, SVs and LTs have a 1-second and 1.5-second buffer at both the front and back of the vehicle, respectively. The constraint is formulated as below:

$$L_{tot}P + 2 \cdot n_{SV} + 3 \cdot n_{LT} \leq B \tag{3}$$

where L_{tot} denotes the total length of the fleet. P is the pace of the fleet. $L_{tot}P$ just gives the total time the fleet occupies the detector within an aggregation time length of B. This constraint assumes that each SV and LT should have at least 2 seconds and 3 seconds of cleared headways behind, respectively. This is a conservative assumption because according to some studies, even for high-volume cases the shortest headways are 1.9 seconds between cars. The headway for the truck-car relationship is always above 2.5 seconds for all cases (Ye and Zhang, 2009). Note that the constraint tends to be effective only for congested cases when the occupancy becomes high. The total time occupied plus the headways for all vehicles should be below B, the observation time length. Without considering the headways, the constraint is simply formulated as:

$$L_{tot}P \leq B \tag{4}$$

which means that all fleet's time occupying the detector is below the total observation time length of B. Although this is a redundant constraint, it is worth mentioning in the paper because the constraint above without considering headways is too loose to alleviate many unreasonable cases where all drivers are driving close to each other at high speed. Finally, the following constraint is formulated to bridge the connection between total length and vehicle combination:

$$L_{tot} = (n_{SV} \cdot l_{SV} + n_{LT} \cdot l_{LT}) \tag{5}$$

where m and n are the number of SVs and LTs during the observed time bin. The sum of SVs and LVs should equal to the total number of observed vehicles also:

$$N = n_{SV} + n_{LT} \tag{6}$$

where N denotes the total number of vehicles in the fleet. Since m and n denotes the number of vehicles, they must be integers between 0 and N as shown below:

$$n_{SV} \in \{0, 1, 2, \dots, N\} \tag{7}$$
$$n_{LT} \in \{0, 1, 2, \dots, N\} \tag{8}$$

(2) Prior speed assumption

The occupancy can be treated as the percentage of time a detector is occupied. In general, a traffic lane with higher occupancy means a higher chance of congestion. Moreover, the concept of traveling speed, traveling pace, and occupancy per SV are convertible each other. In this section, we start to regulate the speed using the concept of occupancy per standard vehicle. An empirical artificial curve is constructed to control the pattern of occupancy per SV vs. total occupancy.

$$o_{sv} = o_{ffsv} + 1.5 \cdot o_{ffsv} \cdot \left(\frac{O}{o_{crit}}\right)^{p} \tag{9}$$

where o_{sv} and o_{ffsv} correspond to the occupancy per SV and free flow occupancy per SV. Assuming a traveling speed of 70 mph, the SV's pace is estimated as follows:

$$P_{ffsv} = \frac{1}{70\frac{mile}{hour} \cdot 5280\frac{feet}{mile} \cdot \frac{1}{3600}\frac{hour}{second}} = \frac{1}{100.98}\frac{second}{feet} \tag{10}$$

Thus, assuming SV's lengths of 17 feet, the occupancy percentage it takes for a vehicle travel to across the detector becomes:

$$o_{ffsv} = 17 \cdot \frac{P_{ffsv}}{B} \cdot 100 = 17 \cdot \frac{1}{100.98} \cdot \frac{1}{30} \cdot 100 \approx 0.56 \tag{11}$$

In the previous empirical function, o_{sv}, o_{crit} and p denotes two control parameters controlling the shape of the line. o_{crit} and p defines a "critical" occupancy and curvature, respectively. In this study, we choose $o_{crit} = 20$ and $p = 3$ to generate a prior assumption of the traveling speed given occupancy. The following relationship between total occupancy and per vehicle occupancy is visualized in the left subplot Figure 3. The right subplot shows the corresponding relationship between occupancy and speed. Furthermore, the red dashed line further simplifies the regulated speed curve using three linear segments as denoted below:

$$v = \begin{cases} min\{65, 90 - 3O\}, 0 \le O < 20 \\ max\{110 - 4O, 1\}, 20 \le O \end{cases} \tag{12}$$

Compared with the previous speed assumption, the new assumption (i.e., the red dashed line) is more reasonable in terms of estimating LTs. The speed is regulated when occupancy is greater than 20 because it is assumed that the traffic is congested in most cases for an occupancy greater than 20. The speed is also regulated with occupancy smaller than 10 for the convenience of computation. The occupancy per SV can be converted to speed, which is represented as the solid black line in the second figure. In practice, occupancy over 25 becomes rare, and the speed prior believe is artificially chosen because high occupancies are more likely to cause by congestion instead of LTs. For congestion cases, by regulating speed, the observed high occupancy is explained by congestion (i.e., low speed) instead of high frequency of LTs.

Formulating The Linear Programming Problem

In this last section, all assumptions and the corresponding constraints to formulate the problem are introduced. In this section, the Linear Programming problem is formulated to solve the best combination of SVs and LTs as well as solving their lengths as shown below:

$$\min|L_{tot} \cdot P - T| \tag{13}$$

Subject to:

$$L_{tot} = (n_{SV} \cdot l_{SV} + n_{LT} \cdot l_{LT}) \tag{14}$$

$$N = n_{SV} + n_{LT} \tag{15}$$
$$L_{tot} \cdot P \leq B \tag{16}$$
$$T = O \cdot \frac{30}{100} \tag{17}$$
$$o_{sv} = 0.56 + 1.5 \cdot 0.56 \cdot \left(\frac{O}{20}\right)^3 \tag{18}$$
$$v = \frac{100 \cdot 17}{o_{sv} \cdot B} \tag{19}$$
$$v = \begin{cases} min\{65, 90 - 3O\}, 0 \leq O < 20 \\ max\{110 - 4O, 1\}, 20 \leq O \end{cases} \tag{20}$$
$$P = \frac{1}{v} \tag{21}$$
$$13 \leq l_{SV} \leq 19 \tag{22}$$
$$65 \leq l_{LT} \leq 75 \tag{23}$$
$$n_{SV} \in \{0, 1, 2, \dots, N\} \tag{24}$$
$$n_{LT} \in \{0, 1, 2, \dots, N\} \tag{25}$$
$$B = 30 \tag{26}$$
$$L_{tot} \cdot P + 2 \cdot n_{SV} + 3 \cdot n_{LT} \leq B \tag{27}$$
$$\delta = |L_{tot} \cdot P - T| \tag{28}$$
$$\delta \leq 0.6 \tag{29}$$

In this study, three versions of models are provided for evaluation. The first version ("V0") is the original LP model under speed assumption. The second version ("V1") is the original LP model under regulated speed assumption. The third version ("V2") is the new LP model under regulated speed assumption. The meanings of all constraints are explained in the previous sections. As a summary, the meaning of all decision variables involved are summarized in Table 2.

Solving the LP Problem at scale

Although the search scope of the problem is relatively small for modern computers, it is still time-consuming to compute the results one by one in real time considering that there are hundreds of such traffic stations generating results every 30 seconds by each traffic lane. Instead of solving each problem, all possible combinations are precomputed in advance to generate a query table. For such efforts, all combinations of volume N and occupancies O are enumerated to solve the problem. A typical range of N is all integers between 0 and 30. For the occupancy O, all integers between 0 and 50 are used to precompute results. In such cases, there are $30 \times 50 = 1{,}500$ different LP problems to solve. Although occupancy can be decimal values, occupancy is rounded to whole numbers for simplification.

Strictly speaking, the LP problem in this study is a Mixed Integer Linear Programming (MILP) problem where some decision variables are continuous values and others are integers only. In practice, instead of generating one optimal solution solving the MILP directly, we just plug in each vehicle combination (i.e., the integer decision variables) and the problem becomes a standard LP problem. After getting the solutions for all combinations, we rank them as the optimal solution for each vehicle combination. If two solutions reach the same costs in the

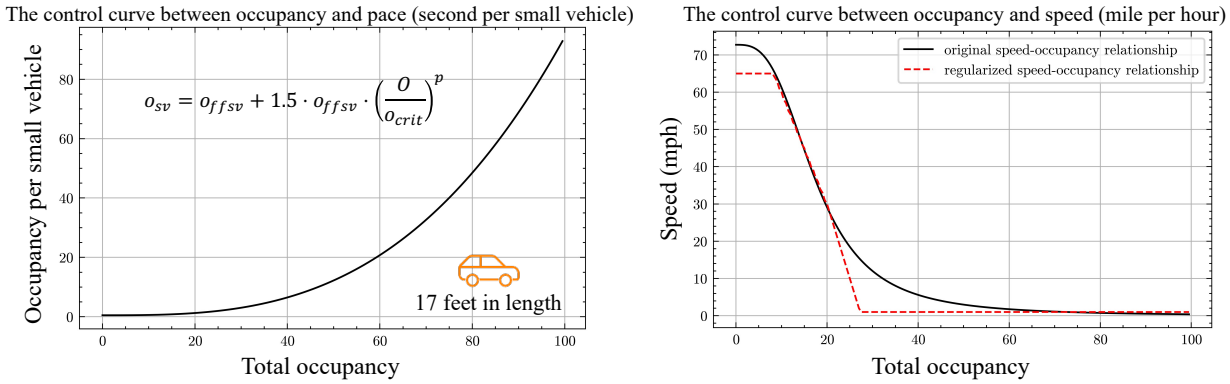


Figure 3. The artificial control curve between traveling pace and occupancy (left figure). The same relationship is converted to the right figure between traveling speed and occupancy

Table 1. The assumptions of three different versions of LP models

Model version	Speed assumption	LP constraints	Descriptions
V0	Normal	(13)-(17); (18)-(19); (21)-(29)	Original algorithm
V1	Regulated		
V2 (proposed method)	Regulated	(13)-(18); (20); (21)-(29)	Consider headways, upper bound flow, etc.

Table 2. A summary of all parameters involved in the Linear Programming problem

Parameter	Description	Unit
L_{tot}	Total vehicle length of the fleet	
P	Travel pace of the fleet	second/feet
T	Number of seconds the fleet occupying the detector	
N	Total number of vehicles	
m	Number of SVs	
n	Number of LTs	
l_{SV}	Average vehicle length for SV	feet
l_{LV}	Average vehicle length for LT	feet
N	Number of observed vehicles	
O	Total occupancy of the fleet	
o_{sv}	Occupancy time takes for a vehicle to travel length of one SV	
K	Density of the fleet	vehicle/mile
q^{ub}	Upper bound of effective traffic flow rate	vehicle/hour
B	Time length of the aggregation time bin	second

objective function (e.g., cost of zero), then solutions with a higher number of SVs are preferred. All infeasible solutions are dropped. If no solution is found (i.e., all problems are infeasible), we assume all vehicles are SV with lengths of 16 feet. Thus, the solution ranked top is used as the solution for that scenario.

Following this routine, the number of LTs for each pair of volume and occupancy can be visualized in Figure 4. Figure 4 (b)-(d) shows the results for the number of LTs given each combination of occupancy and speed. The x-axis and y-axis represent occupancies and traffic volume, respectively. These results are represented using a similar set of settings. The length of time aggregation bin B is 30 seconds in this study. When the number of vehicles is fixed at 4, the number of LTs increases as the occupancy increases from 1 to 10. Then, the number starts to decrease as the occupancy further increases. This is mainly because there is not enough time for all LTs to pass traveling at a lower speed, especially considering that on average there needs at least 2.5 seconds of headways between consecutive vehicles in the fleet. Another way of interpreting the diagram is to keep the occupancy fixed. If occupancy is fixed at 10, the number of LTs also increases as the number of vehicles increases. Later, when there were more vehicles, the number of LTs dropped. This is because if the number of LTs remains the same, the total length of vehicles gets higher to make the traveling speed higher. In such cases, the density of vehicles gets higher than a feasible case. Thus, a contradiction is reached assuming there the number of LTs stays the same or gets increased.

Evaluation Metrics

In this study, two sets of results are of interest. The first is the LT estimation, and the second is speed or travel time along a corridor. For the LT estimation, the following two metrics, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are applied as follows:

$$MAE = \frac{\sum_{i=0}^{\Lambda} |\widehat{m}_i - m_i|}{\Lambda} \tag{42}$$

where Λ denotes the number of records to be evaluated. m_i and $\widehat{m}_i$ denotes the measured and observed LTs, respectively. Similarly, the RMSE is defined as follows:

$$RMSE = \frac{1}{\Lambda}\sqrt{\sum_{i=0}^{\Lambda} (m_i - \widehat{m}_i)^2} \tag{43}$$

For evaluation of traffic speed, the evaluation metrics are the same except the notations of m_i and $\widehat{m}_i$ are substituted to s_i and $\hat{s}_i$ with s denotes the travel speed. For travel speed estimation, Mean Absolute Percentage Error (MAPE) is also included for evaluation:

$$MAPE = \frac{1}{\Lambda}\sum_{i=0}^{\Lambda} \left|\frac{s_i - \hat{s}_i}{s_i}\right| \tag{44}$$

Experiments

Since different regions and roads have different road geometries, traffic, and vehicle distributions, it is important to do more experiments to solve the problem. A total of four different experiments are designed. The first three experiments try to compare against different

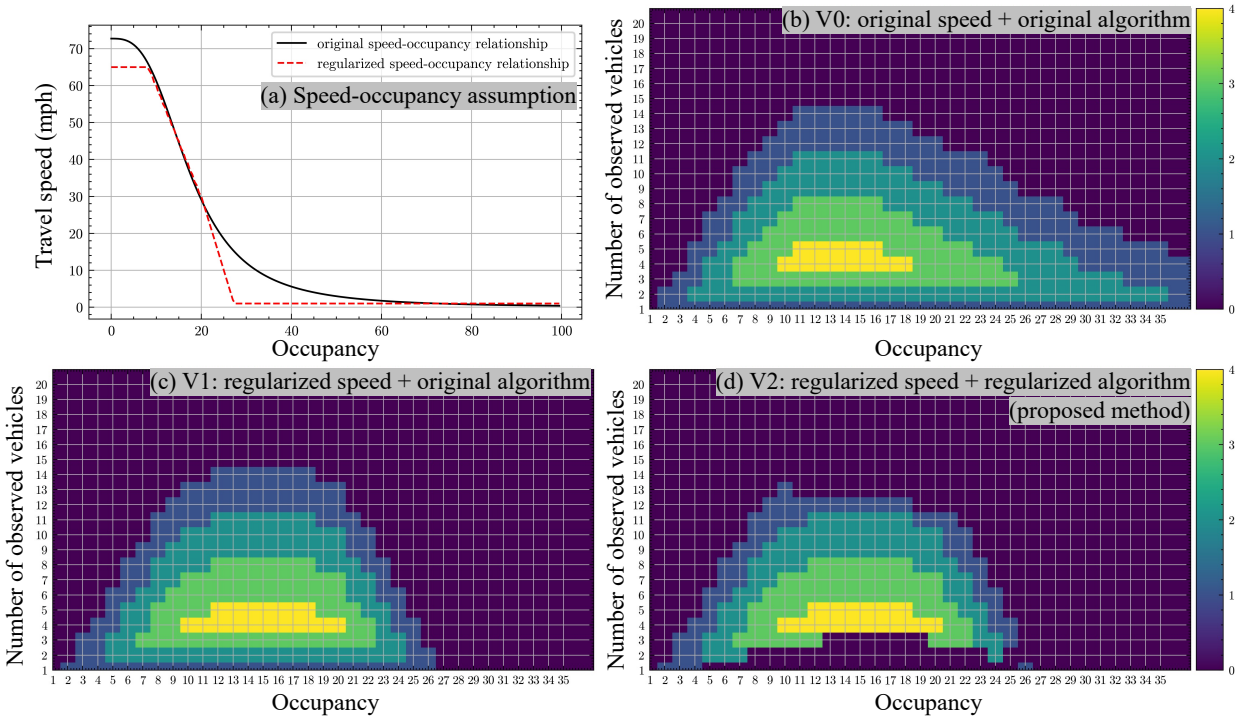


Figure 4. Results of the number of LTs for any combination of observed vehicles and occupancy with respect to different configurations

detectors including WIM detector, dual loop detector, and vehicle trajectory data. The fourth experiment is a study case to utilize the problem at a long corridor level.

A quick summary of the benefits of the precompute strategy

Before discussing experiments in detail, some results are presented in this section to get some general sense. As mentioned, all combinations of results can be estimated and stored at once for fast query instead of compute in time. **Table 3** shows the comparison of computational time under different scenarios. The first column shows the computational time of pre-computing all combinations. There are 75 different speed inputs from 1 mph to 75 mph. There are 30 volume inputs from 1 to 30. There are 50 occupancy inputs from 1 to 50. In other cases, all vehicles are considered as small vehicles. In summary, the V1 and V2 model takes 87.9 and 102.3 seconds to compute all combinations at once. By checking the precomputed matrix, it only takes 3-5 seconds to check results for all observations at a traffic observation station for a week's data. By contrast, solving LP optimization problems in real time takes 32.1-38.2 for the WIM case study and 82.4-89.1 seconds for the PeMS dual loop study. Thus, it is time-efficient to query results from a precomputed matrix instead of running the LP optimization in request. For all the experiments, Python programming language is used. The query table is stored and queried in the Numpy package, and the Linear Programming results are computed using the GurobiPy package.

Besides understanding the advantages of precomputing and querying results, it is also important to visually grasp the relationship between the observed occupancy and speed for each traffic lane and the pre-computed results.

Evaluate LTs estimation by comparing against WIM ground truth

One of the detectors that can generate accurate vehicle classification data is the WIM detector. The detector can measure the spacings and weight for all axles of each vehicle. Thus, WIM can be used as the ground truth data to compare against estimated LTs for the SLD detector. In this study, such a pair of SLD and WIM detectors are found using Caltran's PeMS dataset. In PeMS, the identification code for WIM is "34630" and the identification code for SLD/VDS is "3412041". The first week in April 2024 is observed for each hour by each traffic lane. Note that hourly aggregated counts by lane are the finest aggregation level for WIM data outputs in PeMS. Results show that the count data for all vehicles is almost the same. For WIM, 13 different vehicular types are identified. In this study, truck types from type 6 to 13 combined are considered as LTs.

The comparison results are visualized in Figure 5. Three subplots are made for the time series observations at each traffic lane. The dashed cyan lane and dashed black line correspond to the total vehicles observed by WIM and LDS, respectively. Similarly, the grey solid line and cyan line correspond to the WIM ground truth and the estimated results from the proposed algorithm, respectively. By observation, lane 1 (the innermost lane) is a truck-free lane, whereas the other 2 lanes have more LTs. By comparison, the proposed algorithm just slightly overestimates the number of LTs.

Besides visualization, Table 4 measures the MAE and RMSE results. The error for lane 1 is very small as an LT free lane. The error for lane 2 and lane 3 becomes larger. By observation, these metrics are aligned with the visualization in Figure 5.

Figure 6 shows the frequencies for each combination of occupancy and volume in this diagram. The red dots show the number of observed records for each combination of occupancy and volume. The background maps show the pre-computed results for the number of LTs for each combination. By observation, almost all records are out of the scope without any LT involved. The second lane has a similar pattern, although some records show that there are one or two LTs within the range. The third lane, the outermost lane, has much smaller LTs compared with the first two lanes. In many cases, there are over 3 LTs within each 30 seconds for each lane.

Evaluate the LP algorithm by comparing against dual loop detector

Although WIM detectors generate a precise estimate of vehicle type and vehicle length, they are often established in suburban or rural regions. It only provides a summarization of vehicle class and axle distributions aggregated by hour. For more detailed information, dual loop detectors can be used because they can measure speed information. Thus, we can compare the speed information estimated by the LP algorithm and by measurements from the dual loop detector. For the study, a detector near the intersection of Eastbound I-580 and North Flynn Road is selected. The VDS detector ID is "1031110" in the PeMS system. Data during the first week of April 2024 (April 1st - 7th) is observed. Figure 7 compares the measured speed and estimated speed. The dark green curve represents the 10-minute moving average measured speed, whereas the red dark curve represents the 10-minute moving average speed estimated by the algorithm. The light red and green correspond to the raw readings speed at 30-second intervals. Results show that the speeds between the two series are similar most of the time. Overall, the speed for the proposed LP algorithm makes a conservative assumption for the speed prior. This is because the design philosophy of the algorithm is that the increased occupancy is attributed to the speed decrease whenever possible. Only in cases where a speed drop cannot explain the increase in occupancy, those increased amounts of occupancy should then be attributed to the present LTs. The LP algorithm tries to be conservative when identifying LTs. Table 5 further evaluates the difference using MAE and RSME metrics. Lane 2 has the largest error for speed estimation with the largest RMSE. Lane 5 also has a relatively large error. The speed observation varies a lot for lane 5 because this lane is very close to an upstream exit ramp and a downstream entry ramp. Overall, the predicted speed is good enough to generate travel time information.

Figure 8 visualizes the record frequency in terms of occupancy and volume, although the ground truth results of LTs are not available. The lanes are numbered in ascending order numbered from 1 to 5. The overall explanation and patterns in Figure 8 are similar to that in Figure 6. Figure 9 further zoomed out the observation scope to show the average speed for each combination of occupancy and volume. Results show that the traveling speeds for occupancies above 25 are almost always lower than 20 miles per hour when the volume is low. For such congested cases, it is always assumed that estimating the number of LTs becomes a very hard problem to solve. Thus, a good practice is to regulate the speed instead of overestimating LTs when occupancy is high.

Table 3. A comparison of computational time for three scenarios: (1) precompute all combinations; (2) case study of WIM; (2) case study of dual loop detectors.

	Precompute all combinations	WIM	dual loop detector
Number of records	112,500	60,468	99,550
V0 (compute)	87.9	22.6	83.7
V1 (compute)	87.9	32.1	89.1
V1 (query)	N.A.	**3.0**	**4.6**
V2 (compute)	102.3	38.2	82.4
V2 (query)	N.A.	**3.2**	**4.6**

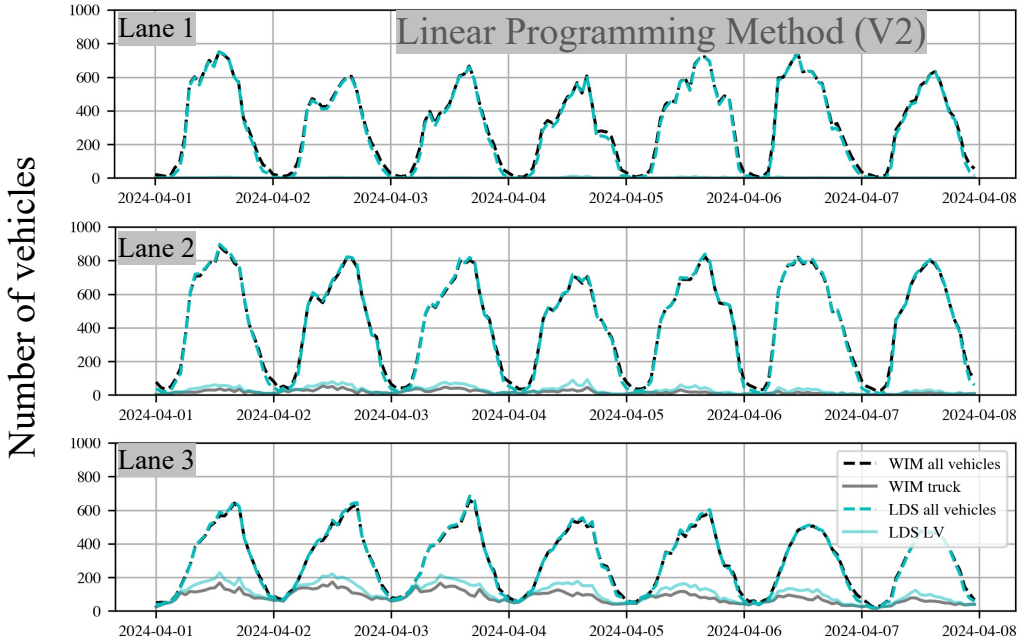


Figure 5. Comparing LP results from Loop detector against WIM ground truth data

Table 4. Evaluation of the prediction error by comparing between WIM measured results and the proposed method

Method	Lane 1		Lane 2		Lane 3		All lanes	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LCM	**0.04**	**0.22**	21.77	27.12	38.60	48.03	20.13	31.84
TCM	6.01	10.41	**8.35**	**11.13**	35.49	43.88	16.61	26.81
V0	4.08	7.12	22.45	29.55	38.31	47.13	21.61	32.38
V1	1.05	2.48	12.08	16.46	27.65	33.65	13.59	21.67
V2	0.99	2.18	11.53	16.23	**25.07**	**32.62**	**12.53**	**21.07**

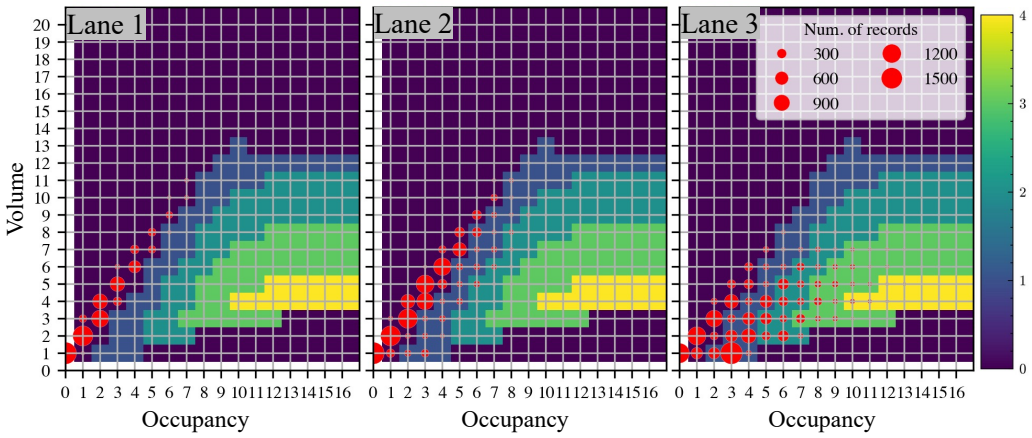


Figure 6. A visual comparison of the results between occupancy and volume for each traffic lane in the WIM example.

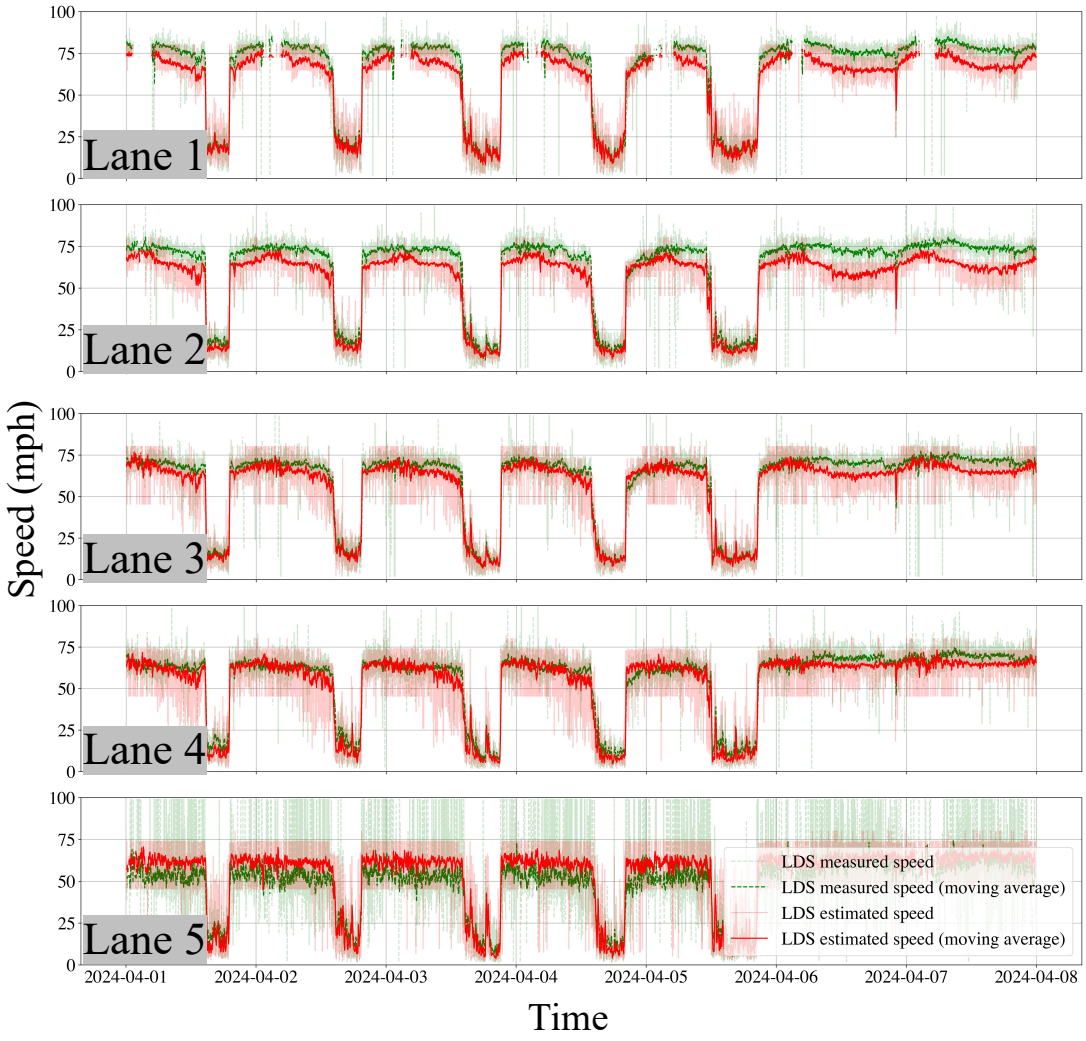


Figure 7. A comparison between dual loop detector measured speed and observed speed. The transparent corresponds to raw inputs, whereas the untransparent lines are 10 minutes moving average

Table 5. Evaluation of the moving average speed prediction error between dual loop detector measured results and the proposed method

Model version	Metric	Lane 1	Lane2	Lane 3	Lane 4	Lane 5
LCM	MAE	3.77	3.81	2.76	3.53	5.76
	RMSE	4.23	4.71	3.40	4.43	6.99
	MAPE	0.076	0.068	0.057	0.074	0.129
TCM	MAE	4.58	11.05	6.43	15.40	11.52
	RMSE	5.70	12.96	8.26	18.89	14.41
	MAPE	0.073	0.178	0.106	0.282	0.249
V0	MAE	5.82	5.23	2.48	3.73	8.70
	RMSE	6.71	6.04	3.20	4.59	10.00
	MAPE	0.091	0.100	0.052	0.094	0.187
V1	MAE	7.24	7.17	3.75	3.64	6.99
	RMSE	8.30	8.10	4.61	4.44	8.26
	MAPE	0.111	0.124	0.068	0.094	0.160
V2	MAE	6.82	7.53	3.84	3.56	6.14
	RMSE	7.66	8.39	4.62	4.35	7.36
	MAPE	0.105	0.129	0.069	0.093	0.144

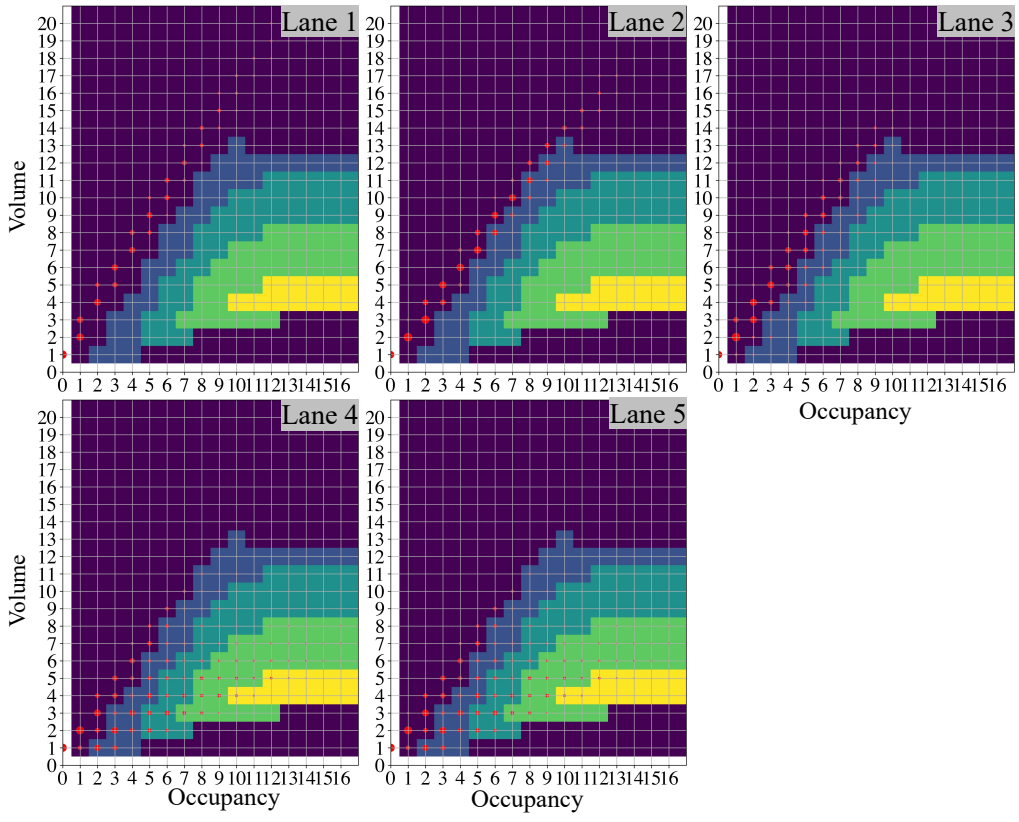


Figure 8. A visual comparison of the results between occupancy and volume

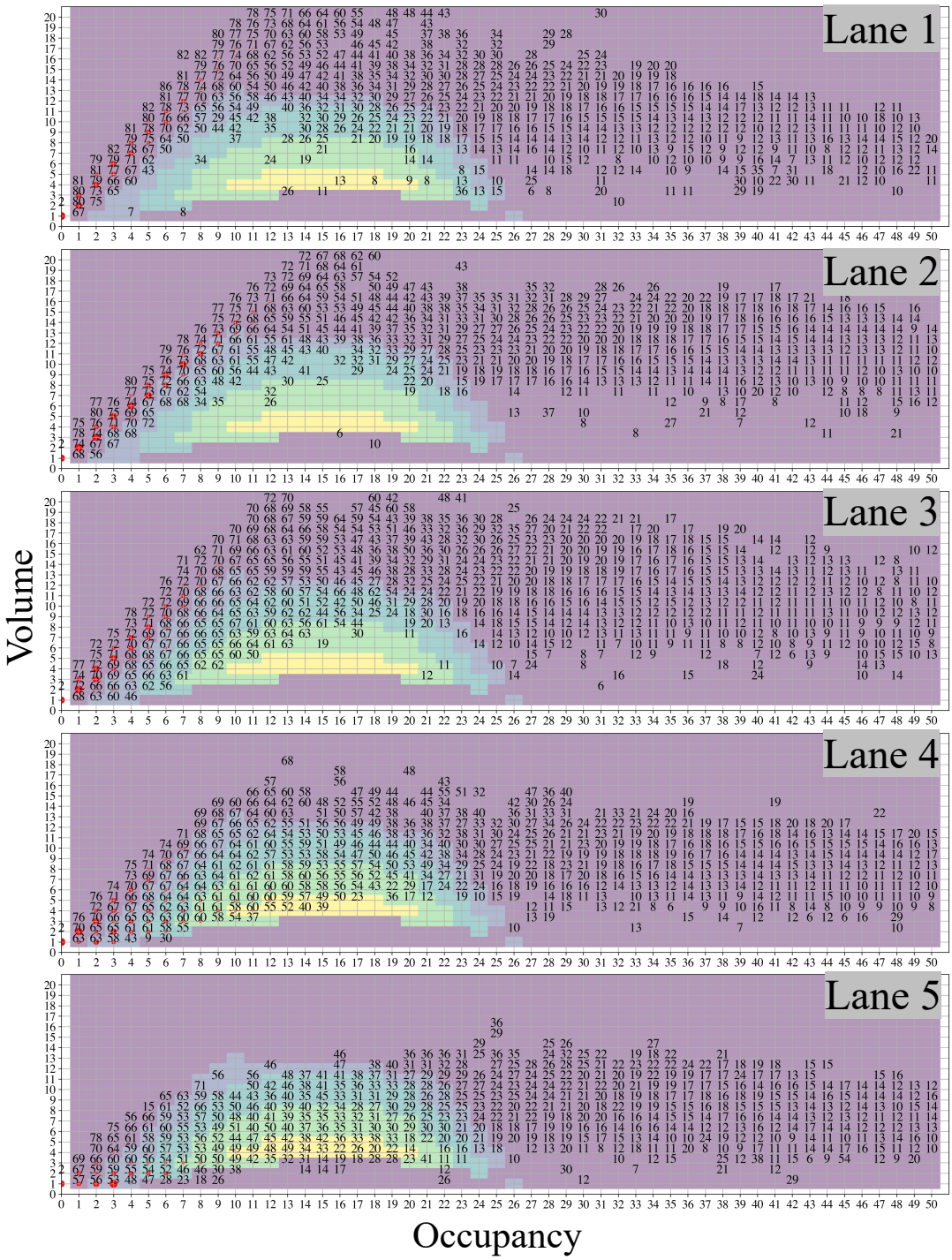


Figure 9. The overview of all average speed for each observed combination of occupancy and volume

Evaluate the LP algorithm with I-24 Motion data in Nashville, Tennessee

I-24 Motion is a new project to identify vehicle trajectories over a 4-mile highway segment at the I-40 corridor by using 294 ultra-high-resolution cameras (Gloudemans et al., 2023). Based on those high-resolution videos, deep learning algorithms are developed to identify and track vehicles with very detailed information generated. The output data includes but is not limited to the vehicle's trajectory, speed, size (i.e., length, width, and height), type, etc. Up till now, raw trajectory data of the morning peak data for 10 different days are publicly available. These 10 days include 8 workdays and 2 Thanksgiving holidays. Except for one holiday day, all other days are recording the inbound traffic (i.e., westbound) between 6-10 AM. In this study, those trajectory data can be used as the ground truth data.

Overall, Figure 10 below shows the spatiotemporal distributions of identified traffic trajectories. The top-left subplot shows the trajectory of a single vehicle identified and traced for 4 seconds. The down-left subplot shows the 10,000 trajectories identified within 14 minutes at the 4-mile corridor.

The comparison results are visualized in Figure 11. Along the 4-mile segment, 5 mile markers (i.e., 59.5, 59.9, 60.3, 60.7, and 62.6) corresponding to 5 RDS detectors are identified. To mitigate synchronization errors between I-24 Motion and RDS, results are compared by aggregated by 10 minutes across all traffic lanes. The green lines correspond to I-24 Motion counts for all vehicles passing the mile marker, whereas the dashed green line corresponds to the number of vehicles observed by RDS. The readings are similar except for mile marker 59.9. The red solid line and dashed line correspond to the I-24 Motion observation and the RDS estimates. Overall, the difference between the two lines is small. The cyan solid line corresponds to the estimated results by rebuilding RDS information from the I-24 ground truth. Overall, the number of LTs is overestimated at the mile marker 60.3 and 60.7 at different mile markers. These estimates tend to overestimate the number of LTs. The reason for the bias is related to the stochastic or oscillation nature of detector readings. This is an extreme test case where the traffic is busy and congested with almost all vehicles being SVs. Indeed, it is very challenging to identify LTs while the average traveling speed is only around 15 mph.

More detailed results of different model versions are summarized in Table 6. Overall, the V2 model outperforms the other two versions at all 5 sites. The speed estimations for all models are similar each other.

Conclusion

Loop detectors or their equivalents (e.g., radar detector systems) are the most widely equipped traffic detectors along the highway systems. However, the detector cannot directly measure travel speed and truck flows. Previously, an LP optimization method was proposed to estimate LTs and traffic speed simultaneously. This study uses an improved method to estimate LTs and truck traffic, considering more empirical constraints, such as vehicle headways, flow rate, etc. Moreover, by enumerating all combinations, the new method can be applied to all detectors to generate a spatiotemporal view of traffic status. Compared with previous truck estimation methods, the method's assumptions are very straightforward.

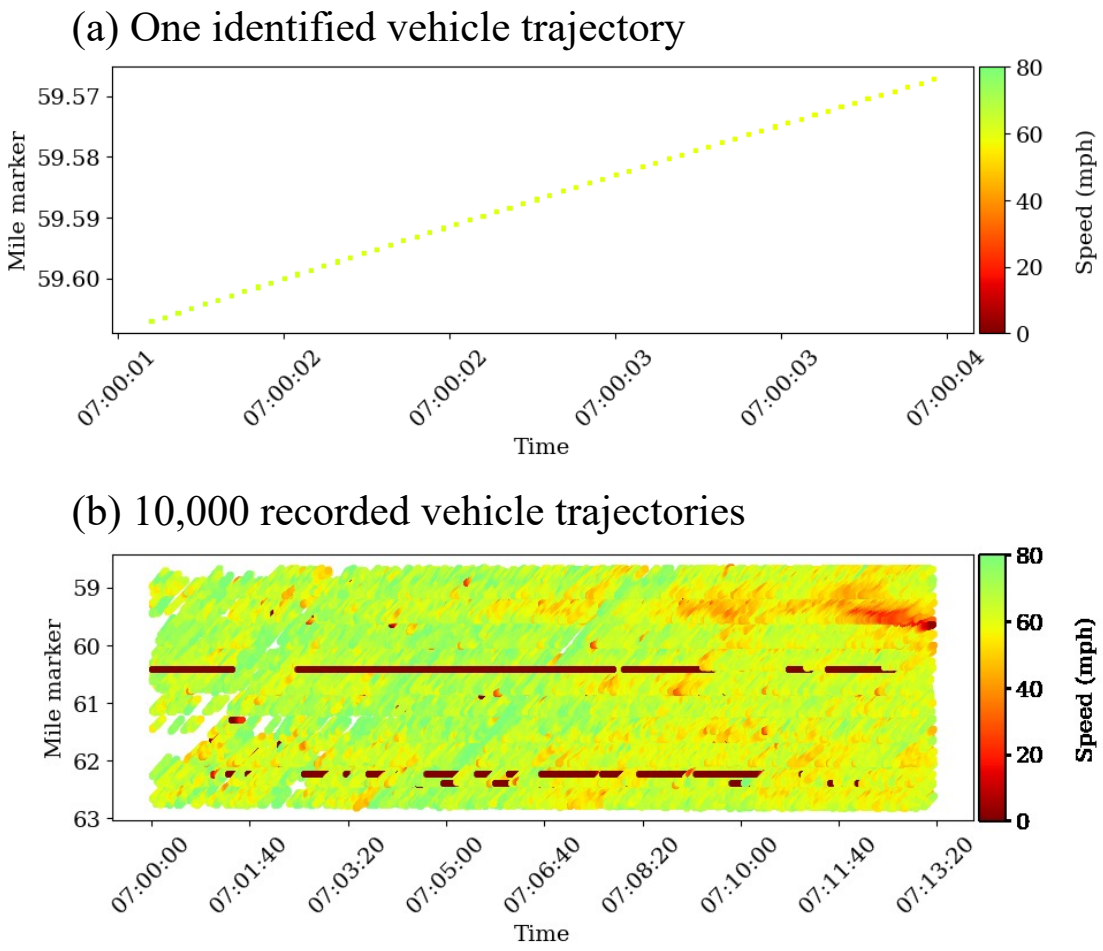


Figure 10. Trajectory examples and lateral distributions of I-24 Motion vehicle trajectories

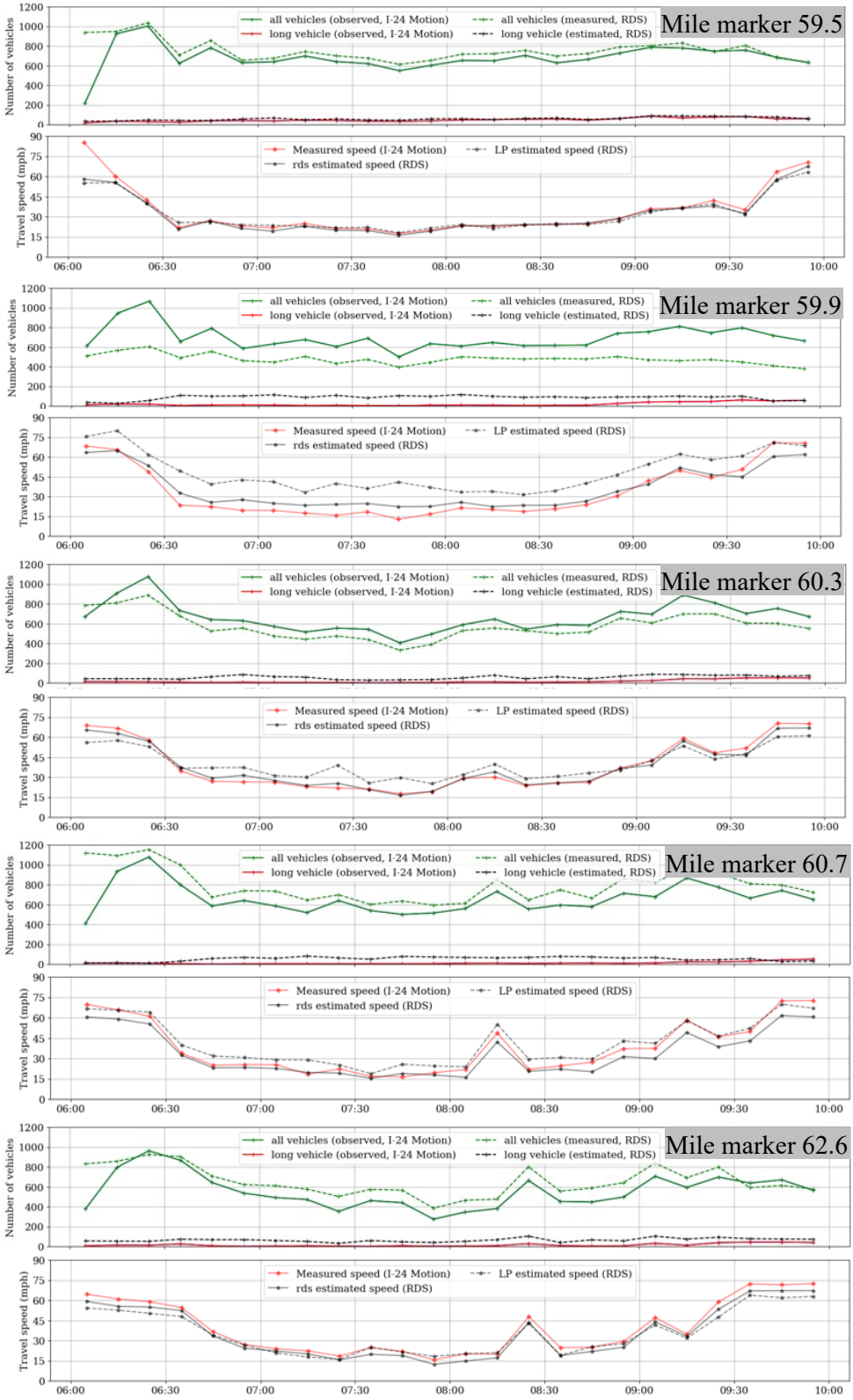


Figure 11. A comparison between RDS estimated results and I-24 Motion vehicle trajectories

Table 6. Evaluation results for estimating speed and LTs using the proposed LP algorithm for morning peak traffic in I-24, Nashville

Mile marker	Methods	MAE	RMSE	MAE	RMSE	MAPE
		LTs		Speed		
MM 59.5	LCM	69.12	72.58	3.16	6.01	0.070
	TCM	31.92	43.37	7.03	9.62	0.186
	V0	27.54	28.86	1.92	**2.26**	0.067
	V1	20.17	22.22	**1.90**	2.56	**0.058**
	V2	**11.21**	**13.66**	2.11	2.84	0.061
MM 59.9	LCM	127.08	139.64	17.07	19.63	0.556
	TCM	**56.42**	**66.35**	**12.38**	**15.58**	**0.385**
	V0	83.71	92.37	17.74	19.03	0.803
	V1	71.83	79.88	15.64	16.96	0.715
	V2	67.62	75.48	15.53	16.84	0.711
MM 60.3	LCM	**18.50**	**22.98**	**5.68**	**8.52**	**0.153**
	TCM	36.04	42.98	6.25	9.81	0.163
	V0	52.38	54.76	7.45	8.53	0.274
	V1	45.12	47.99	6.53	7.55	0.226
	V2	40.58	43.69	6.72	7.71	0.229
MM 60.7	LCM	**33.71**	**38.51**	11.05	15.93	0.325
	TCM	65.54	80.35	10.51	13.98	0.341
	V0	61.25	69.77	5.51	6.32	0.212
	V1	52.92	59.67	4.92	5.70	0.186
	V2	43.79	49.82	**4.35**	**5.11**	**0.166**
MM 62.6	LCM	65.96	75.49	10.72	14.58	0.354
	TCM	54.12	61.13	9.71	12.47	0.288
	V0	66.42	67.92	**3.51**	**4.31**	**0.090**
	V1	57.71	60.04	3.89	5.01	0.093
	V2	**49.38**	**51.18**	4.39	5.60	0.103
All sites	LCM	62.87	36.04	9.54	6.20	0.292
	TCM	48.81	27.07	9.18	5.59	0.273
	V0	58.26	29.56	7.23	4.47	0.289
	V1	49.55	25.57	**6.58**	**4.04**	0.256
	V2	**42.52**	**22.71**	6.62	**4.04**	**0.254**

The proposed method is evaluated and tested in four different experiments. For the first three experiments, the model is compared against two other classic algorithms (i.e., LCM, TCM.) Results show that the new LP method can outcompete the two classical methods in truck estimation for many cases. Furthermore, the proposed method is less likely to generate erroneous estimations given that many radar detectors do not accurately measure traffic volume. By precomputing results for all combinations, the method is easy to use and has many potential applications in transportation management, including a better estimate of traveling time. With the assumption that most LTs are large trailer trucks, monitoring truck activities in metropolitan regions becomes possible based on the most equipped loop detectors or their equivalents.

To the authors' best knowledge, the authors are the first and only team to propose the LP optimization approach. The method can outperform the classical methods, TCM and LCM, for truck flow estimation in many scenarios. Compared with these two classical methods, the method can estimate LTs without jointly looking at consequent or neighboring detectors. Compared with the other methods, the proposed LP approach can integrate all physical constraints and assumptions under one formulation. By contrast, it is hard for other traditional statistical methods to incorporate so many empirical constraints into one formulation. In this study, all LP constraints are based on empirical data from other studies related to traffic flow theory. In other words, it doesn't rely on more specific information changing across different sites. Furthermore, by providing results of all combinations of occupancy and volume, one can look up the number of LTs without solving the optimization problem in real time. Compared with the first version of the approach, the method has improved by introducing new physical constraints in space and time and new regulated prior inputs of speed. In summary, the method depicts a unique picture of estimating the presence of trucks given traffic occupancy and flow.

There are still some limitations to the methodology. First, the algorithm always generates the same result for the same pair of occupancy and volume. However, this might not be true because there might be more than one vehicle type and size combination to reach the optimal value for the objective function. Second, besides estimating speed for a single data output, it is possible to measure the speed estimation for consecutive outputs over time to better estimate traveling speed. This could be formulated by integrating the problem with some stochastic process or adding a hierarchical LP optimization layer. It is also possible to consider medium-sized vehicles with lengths between 25 and 45 feet. In such cases, a reasonable empirical assumption might be that there can be at most one medium-sized vehicle during 30 seconds per lane. Finally, it is possible to fit a better prior speed belief considering occupancy and volume using data for dual loop detectors. Although those mentioned aspects are all potential future research directions, the current approach has the benefits of being simple in formulation and performing well in practice simultaneously.

Other limitations of the method come from the data source itself. For loop detector systems or their equivalents, data quality is a concern. There are often detectors generating low-quality, inconsistent readings. Sometimes, the poor quality is reflected by comparing against neighboring detectors. Some radar detectors only round the results to a whole number, losing some valuable information for occupancy when the occupancy is low.

References

Coifman, B., Kim, S., 2009. Speed estimation and length based vehicle classification from freeway single-loop detectors. Transp. Res. Part C Emerg. Technol. 17, 349–364. https://doi.org/10.1016/j.trc.2009.01.004

Coifman, B., Neelisetty, S., 2014. Improved Speed Estimation From Single-Loop Detectors With High Truck Flow. J. Intell. Transp. Syst. 18, 138–148. https://doi.org/10.1080/15472450.2013.801708

Gloudemans, D., Wang, Y., Ji, J., Zachár, G., Barbour, W., Hall, E., Cebelak, M., Smith, L., Work, D.B., 2023. I-24 MOTION: An instrument for freeway traffic science. Transp. Res. Part C Emerg. Technol. 155, 104311. https://doi.org/10.1016/j.trc.2023.104311

Gu, Y., 2023. Mining Crowdsourced Data for Pavement Maintenance, Incident Management, and Traffic Monitoring.

Gu, Y., Liu, D., Stanford, J., Han, L.D., 2022. Innovative Method for Estimating Large Truck Volume Using Aggregate Volume and Occupancy Data Incorporating Empirical Knowledge Into Linear Programming. Transp. Res. Rec. J. Transp. Res. Board 2676, 648–663. https://doi.org/10.1177/03611981221094569

Hernandez, S.V., Tok, A., Ritchie, S.G., 2016. Integration of Weigh-in-Motion (WIM) and inductive signature data for truck body classification. Transp. Res. Part C Emerg. Technol. 68, 1–21. https://doi.org/10.1016/j.trc.2016.03.003

Kaushik, K., Sharifi, E., Young, S.E., 2015. Computing Performance Measures with National Performance Management Research Data Set. Transp. Res. Rec. J. Transp. Res. Board 2529, 10–26. https://doi.org/10.3141/2529-02

Kwon, J., Varaiya, P., Skabardonis, A., 2003. Estimation of Truck Traffic Volume from Single Loop Detectors with Lane-to-Lane Speed Correlation. Transp. Res. Rec. J. Transp. Res. Board 1856, 106–117. https://doi.org/10.3141/1856-11

Vogel, K., 2003. A comparison of headway and time to collision as safety indicators. Accid. Anal. Prev. 35, 427–433. https://doi.org/10.1016/S0001-4575(02)00022-2

Wang, Y., Nihan, N.L., 2004. Dynamic Estimation of Freeway Large-Truck Volumes Based on Single-Loop Measurements. J. Intell. Transp. Syst. 8, 133–141. https://doi.org/10.1080/15472450490492815

Wang, Y., Nihan, N.L., 2000. Freeway Traffic Speed Estimation with Single-Loop Outputs. Transp. Res. Rec. J. Transp. Res. Board 1727, 120–126. https://doi.org/10.3141/1727-15

Ye, F., Zhang, Y., 2009. Vehicle Type–Specific Headway Analysis Using Freeway Traffic Data. Transp. Res. Rec. J. Transp. Res. Board 2124, 222–230. https://doi.org/10.3141/2124-22

Appendix

A: A probability analysis of corner cases

In Figure 4, one concern is that for all solutions, there are 4 LTs for most combinations of occupancy and volume. A preliminary analysis can be executed to analyze the chance of being filtered out. Assuming total vehicles observed within 30 seconds and the probability of a vehicle being LT are known numbers, the distribution of the number of LTs should follow a binomial distribution with the following probability mass function:

$$f(N; m, p) = \binom{N}{m} p^m (1-p)^{N-m}$$

where N and m are the total observed vehicles and observed trucks within 30 seconds, respectively. p is the share of LTs among the vehicle population. Assuming $p = 0.15$, the probability for more than 5 vehicles in the fleet is:

$$P(m \geq 5) = \sum_{m=5}^{N} \binom{N}{m} p^m (1-p)^{N-m}$$

For a truck-rich traffic lane with 15% of vehicles being LTs among the number of observed vehicles (i.e., $p = 0.15$), Table 7 estimates the probabilities of 5 or more vehicles being LTs. In practice, although it is not very likely to observe more than 7 LTs within 30 seconds, it is still possible to observe 5-7 LTs. By contrast, the chance for more than 7 LTs is rare in practice. This can be analyzed using headways between vehicles, as demonstrated in the last column of Table 7. With more than 13 vehicles involved, the headways become shorter than 2 seconds if more than 5 LTs are observed. The total vehicle length is observed assuming SVs and LT's lengths are 17 and 70 feet, respectively. Thus, the total length of the fleet is $70 \cdot 5 + 17 \cdot (N - 5)$. The "on-time" is computed assuming the fleet travels at 60 mph. Finally, the headways are calculated using $(T - OT)/(N - 1)$. Note that all computations above are conservative to show that the chance of observing over 5 LTs is relatively small, no matter the value of N. In the future, it is likely that more automated driving LTs deliberately form a long fleet to reduce drag. However, such scenarios are still not massively deployed today.

Table 7. A preliminary analysis for the probability of greater than 5 LTs appeared considering probabilities and headways

N	$P(m \geq 5; p = 0.10)$	$P(m \geq 5; p = 0.15)$	$P(m \geq 5; p = 0.20)$	Total vehicle length (feet)	On-time; 60 mph (seconds)	Headways (seconds)
	$p = 0.10$	$p = 0.15$	$p = 0.20$	$m = 5$	$m = 5$	$m = 5$
5	0.000%	0.000%	0.000%	350	3.98	6.51
6	0.000%	0.001%	0.006%	367	4.17	5.17
7	0.001%	0.007%	0.037%	384	4.36	4.27
8	0.002%	0.024%	0.123%	401	4.56	3.63
9	0.006%	0.063%	0.307%	418	4.75	3.16
10	0.015%	0.138%	0.637%	435	4.94	2.78
11	0.030%	0.266%	1.165%	452	5.14	2.49
12	0.054%	0.464%	1.941%	469	5.33	2.24
13	0.092%	0.753%	3.004%	486	5.52	2.04
14	0.147%	1.153%	4.385%	503	5.72	1.87
15	0.225%	1.681%	6.105%	520	5.91	1.72
16	0.330%	2.354%	8.169%	537	6.10	1.59
17	0.467%	3.187%	10.570%	554	6.30	1.48
18	0.642%	4.190%	13.292%	571	6.49	1.38
19	0.859%	5.370%	16.306%	588	6.68	1.30
20	1.125%	6.731%	19.579%	605	6.87	1.22

CHAPTER II. DETECTING TRAFFIC INFORMATION FROM LOW-RESOLUTION VIDEOS COMBINING MULTIPLE DEEP LEARNING ALGORITHMS

Abstract

Recently, Convolutional Neural Networks (CNN) algorithms like YOLO have been massively deployed for transportation applications given CCTV surveillance data. However, the pre-trained YOLO model cannot perform very well under low-resolution cameras. In this study, we propose a framework to combine several different deep learning models to extract traffic information given low-resolution videos. Four separate modules are designed for four sub-tasks: vehicle identification, vehicle classification, vehicle tracking, and trajectory processing. Each task has its own model. A supervised model (e.g., YOLO) is used to identify vehicles. The model is retrained with a vehicle label without identifying the specific vehicle class. For vehicle classification, another self-supervised model (e.g., SimCLR) is used solely for learning vehicle representation. On top of vehicle representation, a small model is trained to identify vehicle classes given hundreds of input datasets. Next, the ByteTrack object tracking algorithm tracks vehicles and generates a time series of coordinates as travel trajectories. Finally, trajectories are converted to aerial view to generate traffic information given the knowledge of the perspective view. Results show that the framework can identify almost all vehicles on the near side of the camera pole on a low-resolution camera in 320×240 dimensions at 15 frames per second. The framework is also good at identifying traveling speeds in both directions. Considering the substantial amount of established CCTV video stream data for traffic monitoring purposes, the proposed framework is constructive with many potential downstream applications, including automatic incident alerts, detecting disabled vehicles, ramp metering, etc.

Key words: contrastive learning; self-supervised learning; YOLO; computer vision; vehicle identification; vehicle classification; CCTV camera

Introduction

For decades, CCTV cameras have become one of the most critical data sources for traffic management. Hundreds of thousands of cameras are established at different locations along major highways in metropolitan areas for many transportation agencies. Indeed, the most effective way to know what exactly happens at a site is by manually checking the site through established CCTV cameras. Thus, many traffic agencies have hundreds of thousands of cameras along the roads. With new technological advancements, it is now possible to process camera streaming data to get knowledge about traffic by running algorithms.

Although there seem to be no limitations for identifying and tracking vehicles using deep learning algorithms, there are many challenges in practice. First, the running speed of deep learning algorithms is relatively slow, even for the well-known YOLO algorithm. With hundreds of cameras, extracting traffic information for all sites at once becomes hard. For example, the data bandwidth of processing hundreds of cameras is extremely high. It is also hard for agencies to preserve these video data because usually there is not enough storage available. To deal with these challenges, a good strategy is to use low-resolution video data.

There are many benefits of using low-resolution videos. First, by using lower-resolution data, processing speed can be much faster. For example, by decreasing the resolution from SXGA (1280×960) to QVGA (320×240), the computing resource for processing one camera's data can now process 16 cameras at once. Moreover, the lower-density data can be publicly streamed to the public by DOTs. For example, traffic agencies like Caltrans (i.e., California's DOT) or Tennessee DOT have hundreds of sites streaming on the website for travelers to check traffic status in advance. There are fewer privacy concerns using low-resolution videos because the resolution is too low to identify unique information like license plates. On the other hand, these low-resolution CCTV cameras are good enough for traffic experts to understand traffic situations. Thus, we believe that low-resolution videos are enough for computers to automatically collect high-quality traffic information with the help of modern deep learning methods. By doing so, algorithms can automatically identify incidents or congestions. Deploying such algorithms can potentially decrease the response time after traffic incidents, which can be very beneficial for mitigating congestion and reducing fatalities.

Extracting information from low-resolution data is not as easy as it seems. Pretrained object recognition models like YOLO may not be sufficient for identifying vehicles under low resolution. Thus, in this study, we propose a framework to extract traffic information with the following objectives:

(1) Identify and classify vehicles over videos using deep learning methods
(2) Track identified vehicles over time to generate vehicle trajectories
(3) Considering the perspective view, holograph transformation can generate an aerial view of vehicle trajectories.
(4) Aggregate trajectory records to generate traffic information, including traveling speed, volume, etc.

Literature Review

To identify vehicles, algorithms should be able to detect multiple objects for each image. The problem has become an important research topic for a long time (Cheng and Chen, 2006). Since

the introduction of AlexNet (Krizhevsky et al., 2017) and VGG Net (Simonyan and Zisserman, 2015), the image classification problem has been dominated by Convolutional Neural Networks (CNN). Since then, many improvements have been made to detect multiple objects at once. He et al. (2015) proposed the Spatial Pyramid Pooling net (SPPnet) for various image sizes. Girshick (2015) developed the Fast Region-based Convolutional Network method (Fast R-CNN). Based on the VGG16 network, the model trains ten times faster and becomes more accurate than SPPnet. He et al. (2018) proposed that the Mask R-CNN generate a mask based on Fast R-CNN. Redmon et al. (2016) proposed the YOLO model, which has a faster running time and is less likely to predict false positives in the background.

The YOLO model has become popular for practical applications since its high performance and fast running speed. In other words, the trained YOLO model can provide good object-detecting algorithms that are out of the box to solve lots of practical problems. Since the introduction of YOLO, the corresponding research team has been updating the model consistently. YOLOv2 adds a multiscale training method that can run at varying sizes (Redmon and Farhadi, 2016). YOLOv3 introduces more design changes using a model with 53 convolutional layers and predicts boxes at three different scales (Redmon and Farhadi, 2018). YOLOv4 further improves the YOLO to reach a balance between speed and accuracy. Several innovative features are used to enhance the YOLO performance. Data augmentation is also used during the training process. Since YOLOv5, different sizes of the model from small to large including nano (YOLO5n), small (YOLO5s), medium (YOLO5m), large (YOLO5L), extra (YOLO5x). Larger sized model has more accuracy but less running speed. YOLOv6 further improves the network architecture with innovations like the Bidirectional Concatenation Module (BiC). YOLOv7 claimed to surpass other models in both speed and accuracy. YOLOv8 and YOLOv9 are also released in recent years. Overall, the YOLO series of models has evolved to be decently helpful for many daily applications. For more information, Terven et al. (2023) review the history of the YOLO series in detail.

Since the introduction of YOLO, a family of YOLO algorithms have been designed to finetune the algorithms for some special cases related to vehicle detection. For example, for foggy weather, applying another set of image processing modules to augment object edges significantly improved the performance of YOLOv3 (Liu et al., 2022). Using image degradation operations (e.g., blur, rotation, flip, crop, etc.), the YOLO model shows a better capacity and robustness in identifying traffic signs (Liu et al., 2018). Mahmood et al. (2020) trained the YOLO model for infrared images and compared the results with the K-means++ clustering algorithm, another method for multitarget detection based on a convolutional neural network. Another model named YOLO-DS is proposed by modifying based on YOLOv5s. The model YOLO-DS announces a speed gain compared with YOLOv5s to make it feasible to run 640×640 image data on the mobile phone within 35 milliseconds. Al-qaness et al. (2021) demonstrated the possibility of extracting traffic information based on a YOLO-based traffic monitoring system. Laroca et al. (2018) designed a robust model to identify license plates based on the YOLO model. S. Liang et al. (2022) discussed the possibility of applying YOLO for edge computing besides cloud computing. Padilla Carrasco et al. (2023) improve YOLO for tiny vehicle detection over parking lots using cenital-plane cameras monitoring the entire parking lots. Lin et al. (2021) implemented a real-time traffic monitoring system counting vehicles with speed and class estimation over the

virtual detection zone. Bin Zuraimi and Kamaru Zaman (2021) combine YOLO for identification and DeepSORT for object tracking.

Recently, self-supervised learning has become popular because it is not reliant on manually labeled data, making it easy to directly self-trained on the massive unlabeled image resources. (Chen et al., 2020) propose a simple framework, SimCLR, for contrastive learning for visual representation tasks. The model learns the features by pulling data from the same image together and pushing data from different images away. The experiment found that data augmentations are critical in predicting tasks. A separate projection head is introduced to map the model output representations to a vector space. Another finding is that a larger batch size can increase the contrastive learning algorithm results. A linear classifier can be used to substitute the model for classification tasks. Results show that the model performs best among semi-supervised or self-supervised models. After SimCLR, many other self-supervised models are proposed for better performance. For example, the self-distillation with no-labels model (DINO) tries to integrate Vision Transformer (ViT) architecture for self-supervised tasks (Caron et al., 2021). Features learned by DINO can be excellent for k-NN classifiers.

Based on objects identified at each frame of the video, more information can be extracted by tracking objects/vehicles across frames. There are many methods capable of tracking objects after identifying objects for each frame. One algorithm is the Simple Online and Realtime Tracking (SORT). Wojke et al. (2017) proposed the DeepSORT algorithm, which further improves the SORT algorithm by providing a deep learning approach to generate association metrics. Several studies used DeepSORT algorithms to track vehicles on highways (Bin Zuraimi and Kamaru Zaman, 2021; Kapania et al., 2020). Recently, a new tracking algorithm, ByteTrack, has been proposed with a better performance than DeepSORT (Zhang et al., 2022). Abouelyazid (2023) compares the performance of three classical algorithms (i.e., SORT, DeepSORT, Byte) over the problem of tracking vehicles. The results show that ByteTrack consistently outcompetes SORT and DeepSORT in object tracking. In conclusion, although there have been breakthroughs in vehicle identification and tracking, there are some limitations with the previous studies in extracting transportation information from videos. A study focuses purely on efficiently identifying traffic information using low-resolution CCTV data. The objective of the paper is to design a flexible framework to combine the benefits of established models from deep learning society. To be specific, the objectives of the proposed framework are sketched below:

(1) Finetune YOLO models for low-resolution. Instead of identifying and classifying the object at once, the YOLO model only focuses on identifying one class for all vehicles.
(2) After identifying the bounding box for a vehicle, a representation vector is learned for each vehicle using a contrastive learning model. A classification model is built on top of the embedding vector given a limited number of labeled samples.
(3) A pre-trained object tracking model (e.g., ByteTrack) is plugged in to track vehicles and generate traffic trajectories along the road.
(4) Together, the three modules mentioned above form one model pipeline to identify, classify, and track vehicles using three different models.
(5) Evaluate the results generated by the framework. The identified information, like traffic volume and speed, can be compared against traditional detector results to evaluate the performance of the algorithms. The quality of tracked trajectories is also evaluated.

Methods

Overall Framework

To tackle the challenging task of identifying, tracing, and classifying vehicles, a framework is established to solve the problem with four separate modules. The first module is to identify vehicles regardless of vehicle class. A model like the YOLO model is effectively used for this purpose. The second module is to learn a vector representation of each identified vehicle from Module 1. After learning vehicle representation, a 3-layer Multilayer Perceptron (MLP) model was given a small set of manually labeled samples. Finally, the fourth module uses an object tracker (e.g., ByteTrack) to track vehicles in the video to generate vehicle trajectories. Based on trajectory results over time, helpful traffic information is gathered, including traffic volume, traffic speed, etc.

Training YOLO model over low-resolution data given samples generated by the pretrained YOLO (teacher-student model)

In this study, YOLOv8 is used for vehicle detection for low-resolution data. Although YOLOv8 models trained by corresponding research teams can be helpful for many applications, there is a need to retrain the YOLO model under low-resolution video data. A qualitative study shows that only a fraction of vehicles can be detected under low-resolution CCTV video. Many vehicles are not detected during the whole lifetime of traveling within the camera's scope. Moreover, many output classes are not helpful for transportation applications. By default, the YOLO trained on the COCO dataset contains the following types related to highway traffic, including passenger cars, motorcycles, buses, and trucks. This kind of classification might not be of interest to traffic managers. Thus, there is a need to retrain the YOLO model.

A teacher-student approach is employed to distill the knowledge from a large model (e.g., YOLOv8x) to a smaller network (e.g., YOLOv8s). The process is summarized in Figure 13. A pre-trained YOLOv8x model is used as the "teacher" to identify vehicles on available low-resolution videos. All identified objects for the 4 vehicle classes (i.e., motorcycle, car, bus, and truck) are treated as vehicles. To control quality, only identified objects with over 90% confidence are passed forward to the dataset for training the smaller student model. Furthermore, an empirical filter is applied to filter out unreasonable identifications. For example, boxes that are over half of the width and height cannot be good samples of a single vehicle. Objects too far apart near the end of the perspective at the top of the camera are not considered good samples since they are too small. Furthermore, all regions in the images with annotated text information should be carved out because they can be mistakenly identified as vehicles sometimes. A manual check is encouraged to filter out unreasonable identification. After applying the rules to control the sample's quality, the identified vehicles are saved to the new custom dataset. All records in the dataset are treated as the "vehicle" type regardless of their original type. A YOLOv8s model is used to train on the new custom dataset to predict the only target class of "vehicle." Thus, a smaller network is built to identify vehicles. Compared to the default YOLOv8x, the newly trained YOLOv8s are much smaller and faster to run. Moreover, a quantitative study shows that the new model can better identify vehicles for low-resolution images.

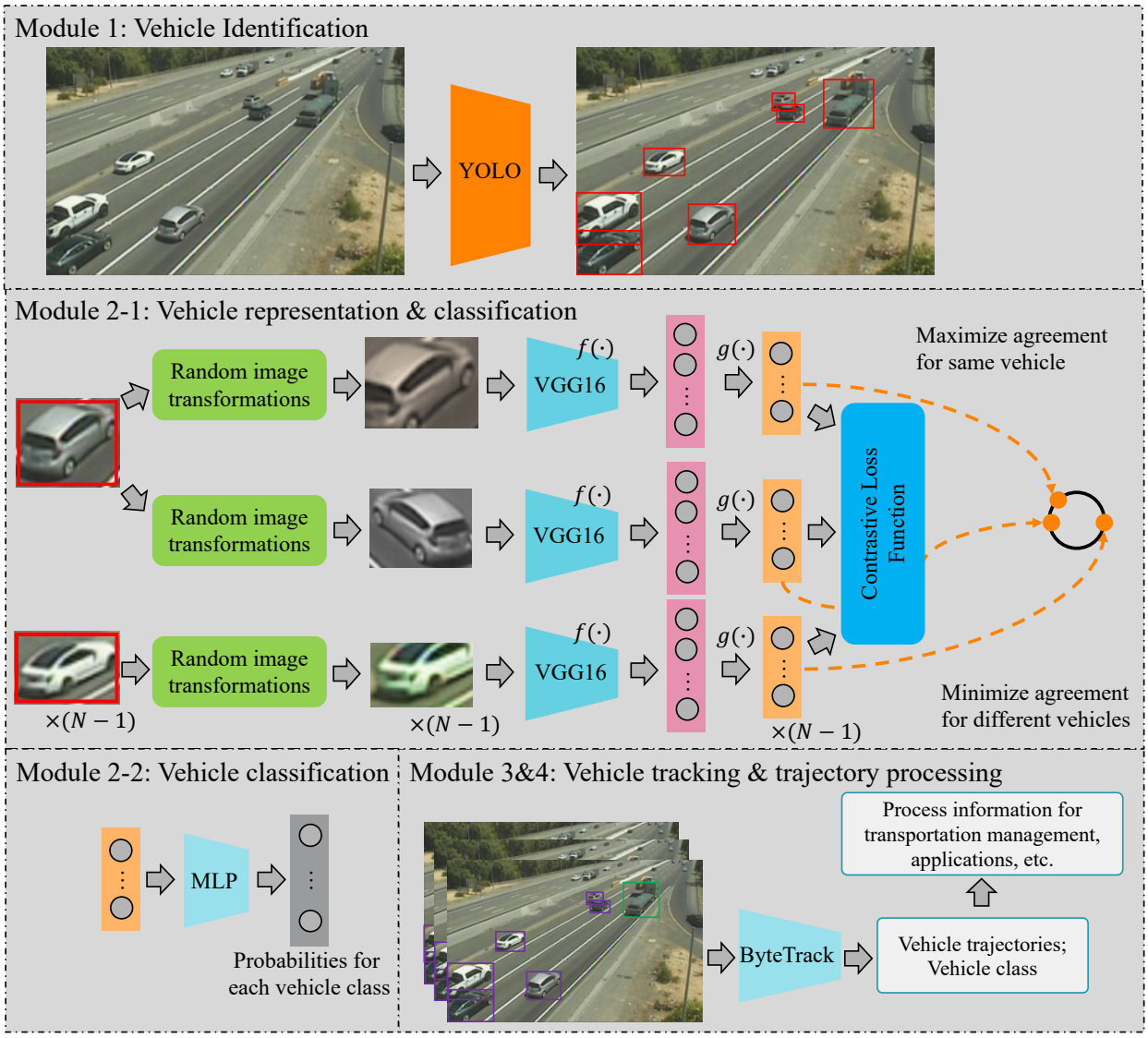


Figure 12. A framework to fusion YOLO and self-supervised model for information detection.

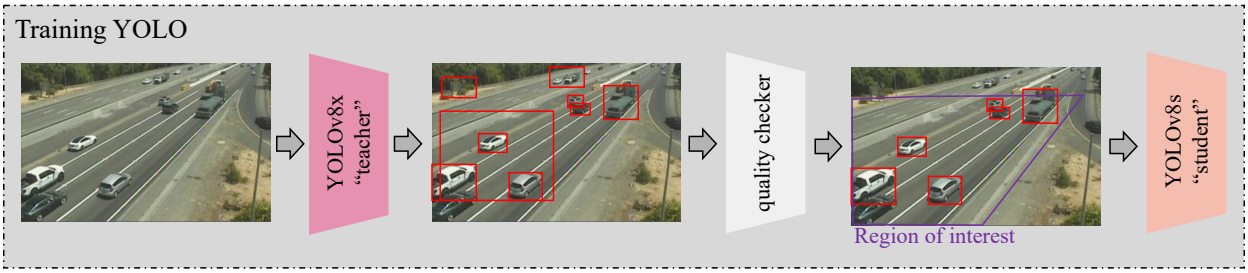


Figure 13. Training YOLOv8s (small student) model with custom dataset generated by pretrained YOLOv8x (large teacher) model.

Learning vehicle image representations using self-supervised model

Traditionally, thousands of labeled results are needed to train a deep learning model. This makes it very time-consuming to label and train a model. Moreover, for many small classes like motorcycles or ambulances, it is hard to find enough image samples for training purposes. Thus, a self-supervised approach is preferred to learn the vector representation of the vehicle without labeled results. In this study, the standard SimCLR model is applied to demonstrate the ability to learn vehicle representations, although other self-supervised models are also possible. The idea is to demonstrate that the framework works well even with the basic self-supervised model. Figure 14 below reviews the core idea of the SimCLR framework. Each image x corresponds to the cropped image representing one vehicle. A series of image transformations are randomly applied to the image. Those transformations include pixel jitter, color to greyscale, flip, crop, resize, etc. Thus, $\tilde{x}_i$ and $\tilde{x}_j$ corresponds to two augmented or "polluted" images after applying transformations. For both augmented images, a CNN $f(\cdot)$ is applied to learn an output h_i as the representations of the image. Another 2-layer MLP transformation $g(\cdot)$ is applied as the projection head to learn final embeddings $\tilde{z}_i$ and $\tilde{z}_j$ for images $\tilde{x}_i$ and $\tilde{x}_j$, respectively. Two embeddings, $\tilde{z}_i$ and $\tilde{z}_j$, should be as close as possible. For embeddings getting from other images, $\tilde{z}_k$, the distance between $\tilde{z}_i$ and $\tilde{z}_k$ or between $\tilde{z}_j$ and $\tilde{z}_k$ should be as far as possible as they represent different objects.

Given the main idea of contrastive learning, a core metric is designed for modeling training named the contrastive loss function:

$$l_{i,j} = -\log \frac{\exp\left(sim\left(z_i, z_j\right)/\tau\right)}{\sum_{k=1}^{2N} 1_{[k\neq i]} \exp\left(sim(z_i, z_k)/\tau\right)}$$

where i and j corresponds to the index of two images. $sim(\cdot)$ corresponds to the cosine similarity between two vectors embedding vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ as $sim(\boldsymbol{u},\ \boldsymbol{v}) = u^T v/\|u\|\|v\|$. $1_{[k\neq i]}(\cdot)$ equals to one if $k \neq i$, and τ represents the temperature parameter. Each time, a large batch of N images are used (e.g., 1,000 images). Among $2N - 1$ representations to compare against, only the one with index j is from the same image. The other vector embeddings are all from other vehicles. Note that in practice for vehicle identification, a heuristic is employed to generate a training sample every 4 seconds apart to make sure that one vehicle is not captured twice from the YOLO model.

Training prediction classifiers as prediction heads

After learning the vehicle representations, it is expected that different vehicles should have different representations. On the contrary, the same vehicle model at different angles should still have very similar vector representations. Therefore, with only a limited number of data, building a classification predicted on top of the embedding result becomes possible. A small custom dataset is gathered with three classes: 1) small vehicle, 2) small truck, and 3) large truck. 200 records are prepared for each class. Small vehicles are mainly composed of passenger cars and pickup cars. Small and large trucks correspond to delivery cars and trailer trucks. Another 300 records are prepared as the testing set. Results show that with only ten images per sample, the performance over the testing set is 85%. Using 50 images, the performance for testing increases to 94%. This efficiency makes it easy for stakeholders to train different classifiers for different

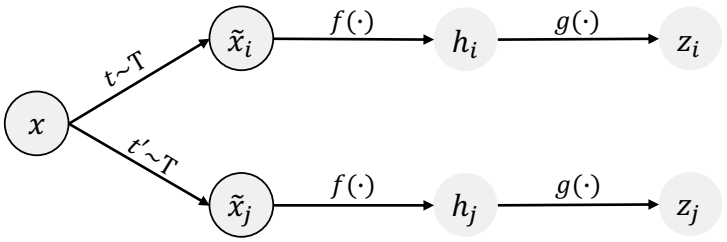


Figure 14. A review of the core SimCLR process to generate pairs of visual representations.

applications. Furthermore, the same MLP is trained to the pre-trained VGG model to compare the model with the other model.

From Raw observation to transportation insights

Finally, given the vehicle trajectories and classes collected over time, one can extract helpful transportation information. A cordon line can be drawn near the bottom of the image. By observation, trajectories are clustered for each lane, making it possible to cluster vehicles by one lane.

Given that the CCTV camera is a stationery video with linear perspectives, it becomes possible to transform data to aerial view using spatial transformation. Figure 15 shows the step of converting trajectories from perspective to aerial view using a perspective transformation matrix. The matrix can be computed using the "cv2.getPerspectiveTransform(src, dst)" for the OpenCV package where "src" and "dst" denote the source and destination point. The 3×3 transformation matrix M is used to represent the matrix transition:

$$\begin{bmatrix} t_i x_i' \\ t_i y_i' \\ t_i \end{bmatrix} = M \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Where (x_i, y_i) corresponds to the coordinates in the origin image with perspective. (x_i', y_i') corresponds to the target coordinate of the origin coordinate (x_i, y_i). By defining four corners of the detection zone, the detection zone can be transferred to the transfer matrix M.

In this study, the detection zone is transferred into a matrix with the size of 100×100 (see left bottom image). By identifying the distance of the detection zone, one can estimate the length of each pixel. Thus, with a reference distance, vehicle speed can also be calculated. In the United States, the interstate highway usually has a lane separation line of 10 feet separated by 30 feet between the separation white lines. Note that the low-quality identifications are filtered out if a vehicle is only identified for less than four frames, given a video is 15 frames per second. Practitioners can manually measure the length of the detection zone using interactive map applications like Google Maps, etc.

Experiments

Although the Methods section introduces the steps and some results for each module, it is necessary to study the performance and validity of the proposed method further. Three experiments are designed. The first experiment compares the effectiveness of contrastive learning methods in model classification over traditional methods. The second experiment compares the performance of the retrained YOLOv8s model with the pre-trained YOLOv8x model. The third experiment compares the detected speed and volume information identified by the algorithm with radar detectors.

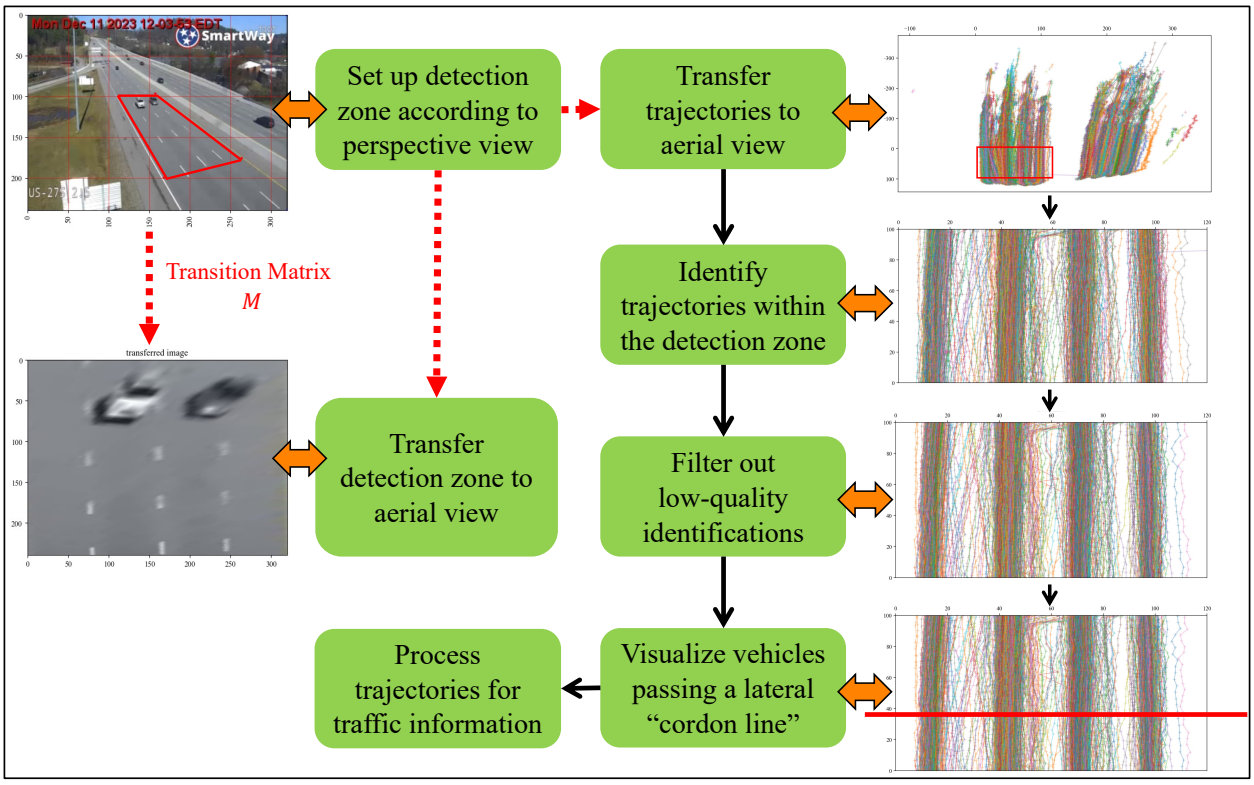


Figure 15. Steps of converting raw trajectories to projected trajectory.

Experiment 1. Comparing self-supervised learning with finetuned model for classification tasks

In this study, a self-supervised method is proposed to classify vehicles. In this study, identified vehicles from 20 hours of videos are used by using a pre-trained YOLOv8x model. Only vehicles identified with 90% confidence can be used for training the data. Figure 16 shows the augmented results of six different vehicles. By contrasting between samples, the model gradually learns how to distinguish two kinds of models. In this study, Figure 17 demonstrates the three vehicle classes that can cover most of the vehicles on the road, including (1) small vehicles, (2) small trucks, and (3) large trucks. Small vehicles and large trucks correspond to passenger vehicles and trailer trucks. Small trucks correspond to large vehicles that are not trailer trucks (e.g., delivery trucks, engineering vehicles, etc.)

After using the SimCLR framework to learn the model, Figure 18 visualizes the representation in two dimensions. Another classifier is trained to visualize three types of vehicles: small vehicles, small trucks, and large trucks. The small vehicles are passenger sedans and small SUVs. The large truck refers to all trailer trucks with at least a removable trailing part. Between the size of a small vehicle and a large truck, the small truck in the middle refers to trucks without a trailer part—for example, the delivery truck, sand carrying truck, concrete truck, etc. Special vehicles like small SUVs with trailer parts, ambulances, and engineering vehicles are all considered small trucks in this classification. In this study, only 200 training samples are selected for each class of vehicles, and 100 vehicles are selected as the testing sample for each class.

After getting the vehicle representation and training data, a classifier is trained on top of the SimCLR output representations by adding a 3-layer MLP model with the output of predictions of the three different classes. Figure 18 shows the representations denoted in 2 dimensions using the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm. The left and right subplot corresponds to the representations for the training set and testing set, respectively. The visual representations show that the passenger vehicles are easy to distinguish as they have unique representations. The large truck and small truck representations are mixed since they have more similarities. However, in many cases, there are some small clusters for large and small trucks. This means that each class is composed of several small trucks. Note that since the input data is different, the results of the t-SNE algorithm are different. The first plot's vehicle representations are transferred.

A suitable property of self-learning is that the model has learned a lot of useful features to distinguish between different vehicles by applying the self-training (i.e., contrastive learning) approach. Thus, the model only needs a handful number of records to get good results. In this example, by using ten labeled data as input, the performance in the testing set is around 85%. With 20 and 50 training data, the performance in the testing set is over 92%. Finally, a confusion matrix of the model performance for the proposed contrastive learning method is given in Table 8. By contrast, using all 200 images as training data to finetune the pre-trained VGG model, the performance is only 63.9% among testing datasets, although the accuracy for the training set is over 98%. Thus, the contrastive learning method can effectively learn vehicle representation and improve classification results.



Figure 16. Six pairs of augmented images for self-supervised training under SimCLR model.



Figure 17. New classification of small vehicle ("sv"), small trucks "st", and large truck "lt".

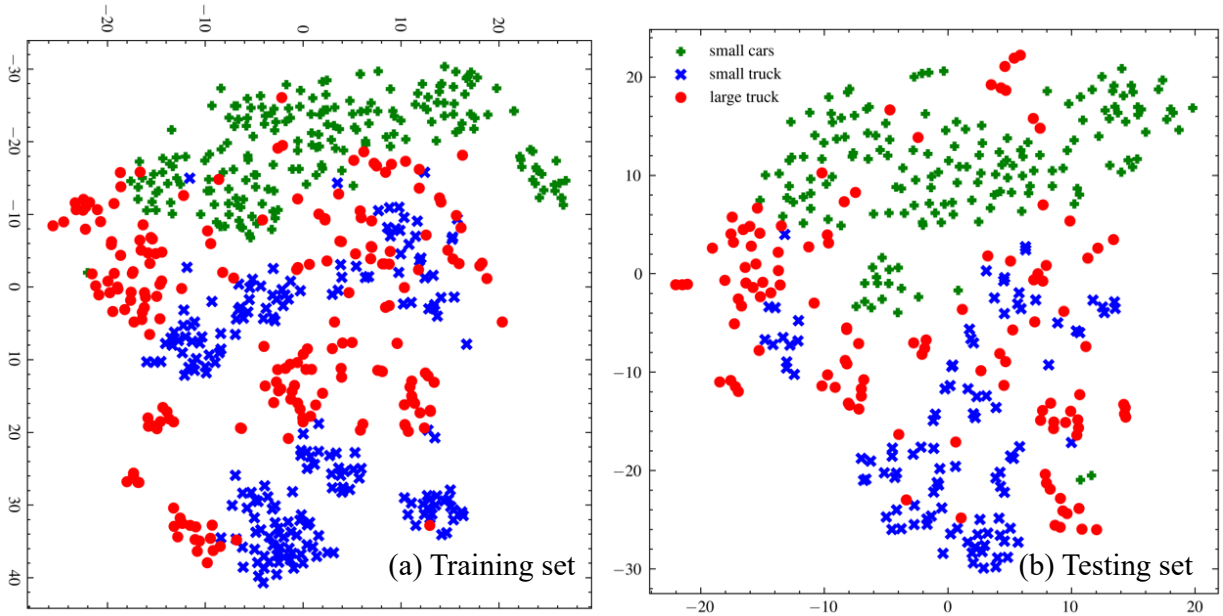


Figure 18. A Visualization of Embeddings Learned by SimCLR model for three types of vehicles.

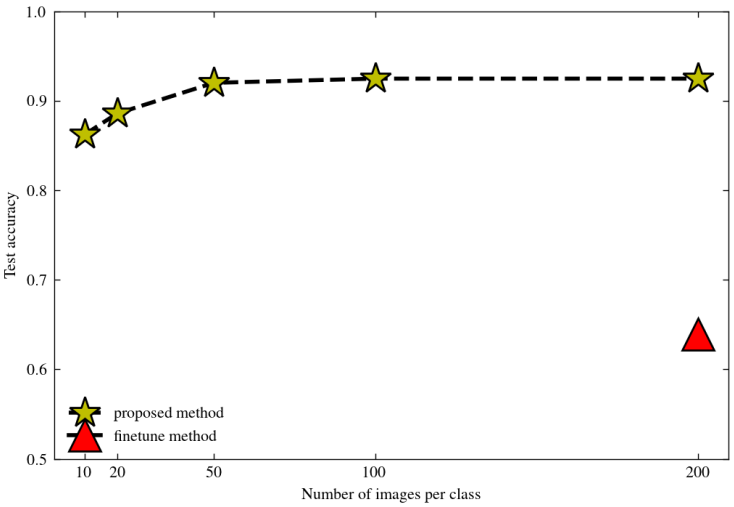


Figure 19. Extract transportation information given trajectory outputs.

Table 8. A confusion matrix by different vehicle class for the proposed method

	Small vehicle	**Large vehicle**	**Large truck**
Small vehicle	**158 (95.75%)**	0 (0.00%)	6 (5.13%)
Small truck	1 (0.61%)	**94 (91.26%)**	7 (5.98%)
Large truck	6 (3.64%)	9 (8.74%)	**104 (88.89%)**
Total number	165	103	117

Experiment 2. A qualitative study between proposed framework and pretrained YOLO

As mentioned, the pre-trained YOLOv8x, the largest, most capable version of the YOLOv8 series, cannot handle low-resolution images well. Figure 20 provides a qualitative comparison between the pre-trained YOLOv8x model and the proposed framework. The upper row represents the annotated results of the pre-trained YOLOv8x model, whereas the lower row represents the result of the proposed method. Each column corresponds to the same video frame. By comparison, the proposed method has a better capability of identifying vehicles.

For the proposed method, the classification module based on self-supervised learning provides a reasonable estimate of vehicle classes. However, the proposed classification method doesn't work well in some corner cases. Figure 20 (d) wrongly classifies a small black passenger vehicle as a large truck because the vehicle is truncated in the image. Similarly, image Figure 20 (b) wrongly annotates two small vehicles as small trucks. This might be because they are close to the border of the image, and their shape is similar to vans or utility vehicles. This classification error can always be mitigated by providing more examples. Figure 21 shows six consecutive snapshots ordered from (a) to (f) at the same site. Each snapshot is separated apart about 0.33 seconds apart. All vehicles in the snapshots are small vehicles. However, two vehicles are identified as small trucks when they travel to the bottom of the image as their shape is distorted. In short, one should classify vehicle classification from multiple snapshots instead of one snapshot. The following experiment will discuss vehicle tracking and trajectory extracting to get more detailed traffic information.

Table 9 compares the running speed within the same framework using a Nvidia GeForce 3090 graphic card. The YOLOv8s method corresponds to the retrained method. The running time is composed of three parts. The first part is running the YOLO algorithm to detect vehicles. The second part is to run the classification model for each vehicle based on SimCLR. The third part corresponds to tracking vehicles across different frames of the video. There is a tiny extra cost of recording the trajectories, etc. The running speed in this experiment is around 140 frames per second. This running speed means that if videos are 15 frames per second, the same graphic card can process videos from over 8-9 cameras at the same time. The pre-trained algorithm using YOLOv8x, a larger model, can only process video around 60 frames per second. The proposed method can run 2.3 times faster than the default method.

Experiment 3. Comparing extracted traffic information with radar detectors and NPMRDS data set

Besides visually comparing results with the default YOLO algorithm, comparing results with other data sources is also convincing. Figure 22 shows the study site with both CCTV camera and radar detector systems available. The site is at mile marker 380 along I-40 highway in Knoxville, Tennessee. The study period is a Thursday from 17:30 to 18:30 on July 12th, 2021. During the one-hour peak hour, the traffic is very busy. For eastbound, the traveling speed is very low at the video's beginning and end periods. The westbound traffic travels at a high speed, although the flow rate is also high. Overall, it is a challenging site since several different traffic statuses are involved within one hour. The information on this test site is never used during the training process. Besides comparing results with radar detectors, the lower left plot shows the corresponding NPMRDS link generating traveling speed. "121+04131" and "121-04131"

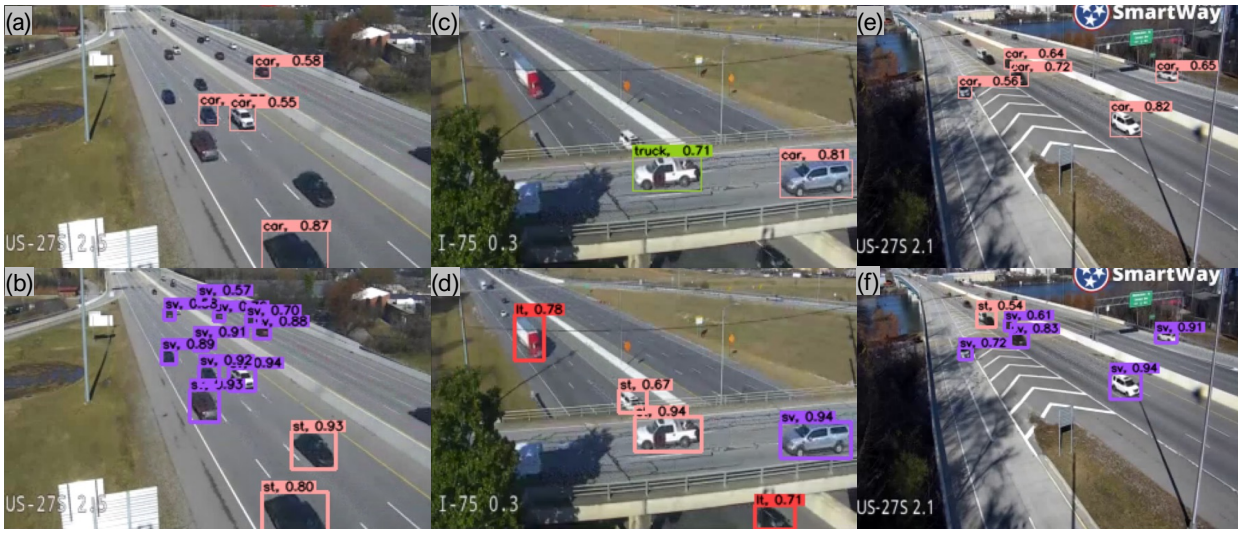


Figure 20. A qualitative comparison between YOLOv8x algorithm (above row) and the proposed framework (below row).



Figure 21. Six consecutive snapshots of a traffic fleet and identification results at the same site observing vehicle types of the traffic fleet.

Table 9. Comparison of running speed of different algorithms (unit: frames per second)

Running speed	Proposed framework+YOLOv8s (proposed method)	Proposed framework+YOLOv8x (pre-trained method)
Video 1	142	62
Video 2	138	60
Video 3	146	61

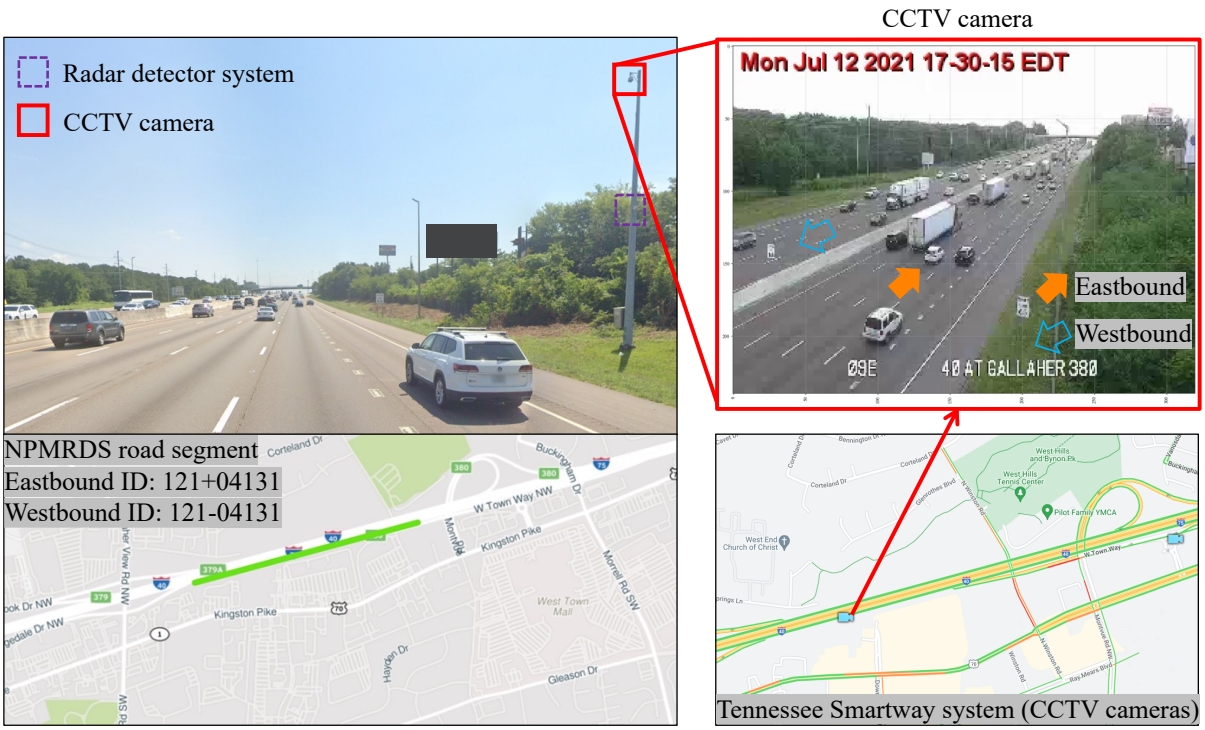


Figure 22. The study site at mile marker 380.0 at I-40 highway near Knoxville, Tennessee.

correspond to the Eastbound and Westbound traveling direction overlay with the CCTV's observing region.

Figure 23 compares the traveling speed estimated or measured by different approaches. The upper plot and bottom plot show the speed distribution of the eastbound (near side) and westbound (far side), respectively. The x-axis and y-axis correspond to the time of the day and traveling speed in miles per hour, respectively. For each plot, the black solid line represents the observed speed aggregated every 30 seconds. The dashed red line corresponds to the moving average values of traveling speed with a window size of 10 minutes. The cyan line with cross dots corresponds to the speed output of the NPMRDS data set from this travel segment. The traveling speed is aggregated every 10 minutes. Results show that the speed estimated by the proposed method is close to the NPMRDS. Since the NPMRDS dataset is processed from GPS recordings of traveling vehicles, the speed measured by NPMRDS can be viewed as a reasonable speed measurement. Finally, the blue dotted line corresponds to the RDS readings. In this case, the readings are overestimated by this RDS detector. The RDS speed is very coarsely estimated by assuming mean vehicle length. Overall, the proposed video recognition algorithm is surprisingly good at measuring speed. Note that there is a shift in speed pattern across different approaches. This is because the various systems have their clocks, and for these systems on the field, it is normal for there to be a time shift of several minutes.

Figure 24 shows the time series of traffic volume using different methods for eastbound (near side) and westbound (far side). The blue dotted line corresponds to the RDS volume for each 30 seconds. Although the RDS cannot directly measure the traveling speed, it is good at estimating the number of vehicles passing a site at each traffic lane. Thus, the RDS readings in the dotted blue line can be approximately viewed as the ground truth. The red dashed and black solid lines correspond to the proposed method with retrained YOLO and the pre-trained YOLOv8x model. For this test case, the proposed YOLOv8s performs slightly better in identifying vehicles than the pre-trained YOLOv8x model, although the proposed model is smaller. For this challenging case with a minimal video resolution of 320×240, it is hard to identify all vehicles correctly using a computer vision algorithm. On the near side (eastbound), the proposed algorithm did recognize most of the vehicles. On the far side (westbound), the proposed algorithm failed to identify many vehicles, although the algorithm is slightly better than the pre-trained YOLO model. Overall, the criterion for identifying a vehicle is conservative in this study because only trajectories crossing the middle lane of the detection zone are identified as vehicles. A qualitative check shows that most vehicles are identified at least once, even on the farther side. It is just that they are not identified at the center of the detection zone.

Besides comparing the quantity of identified vehicles, it is also possible to check the quality of identified vehicle trajectories by their length within the detection zone. Overall, the longer the trajectory, the better the observations. Figure 25 plots the survival curve for the proposed and pre-trained YOLO methods. The x-axis denotes the length of detection in the number of pixels. Since the detection zone is transferred to 100 pixels along travel directions, 100 indicates that the trajectory is observed from the beginning to the end within the detection zone. The first row shows the number of records remaining at a given pixel length, and the second row shows the percentage of records. By comparison, the proposed method finds more vehicles. The survival

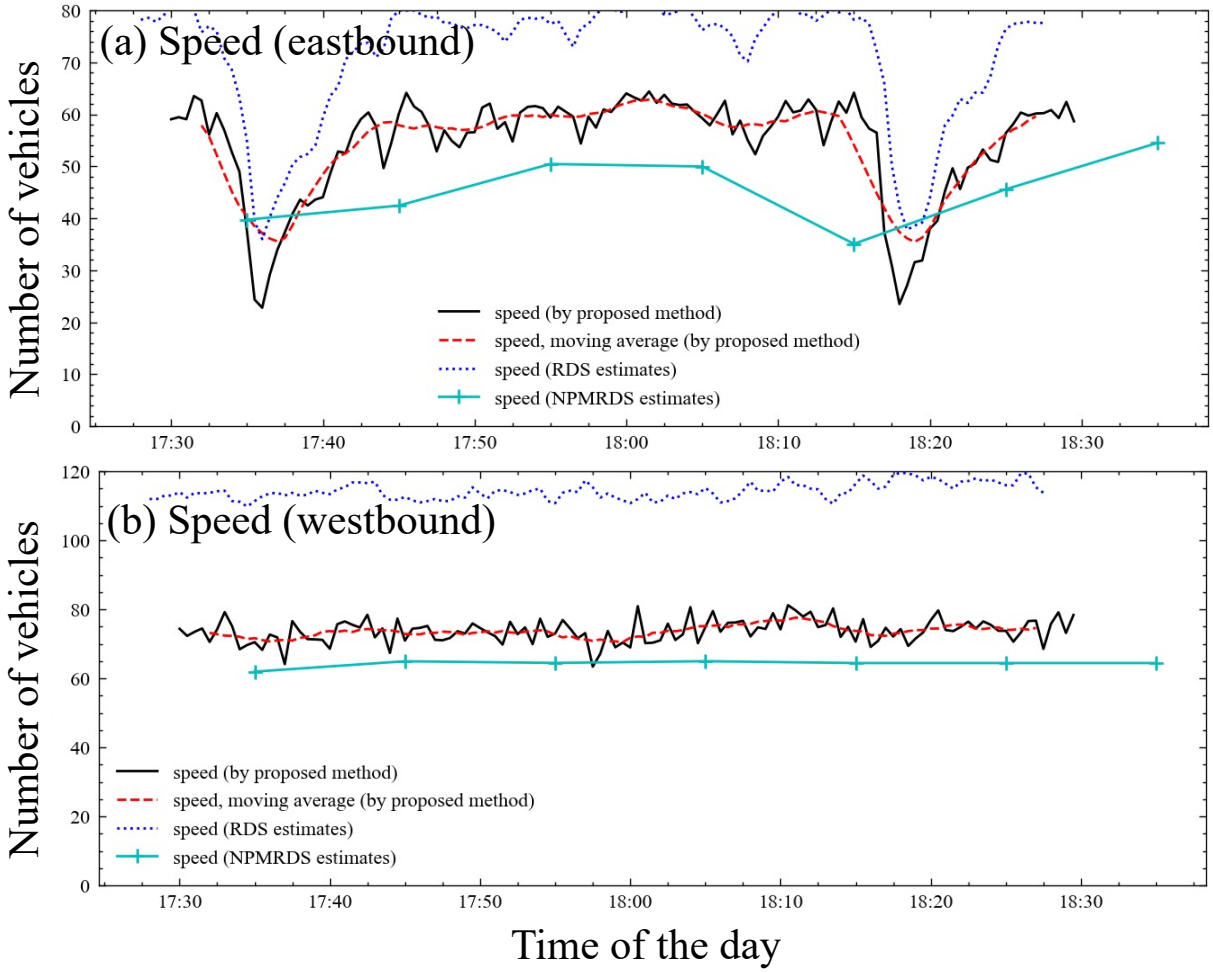


Figure 23. Speed comparison between proposed method, RDS detector, and NPMRDS records for eastbound and westbound at the studied site.

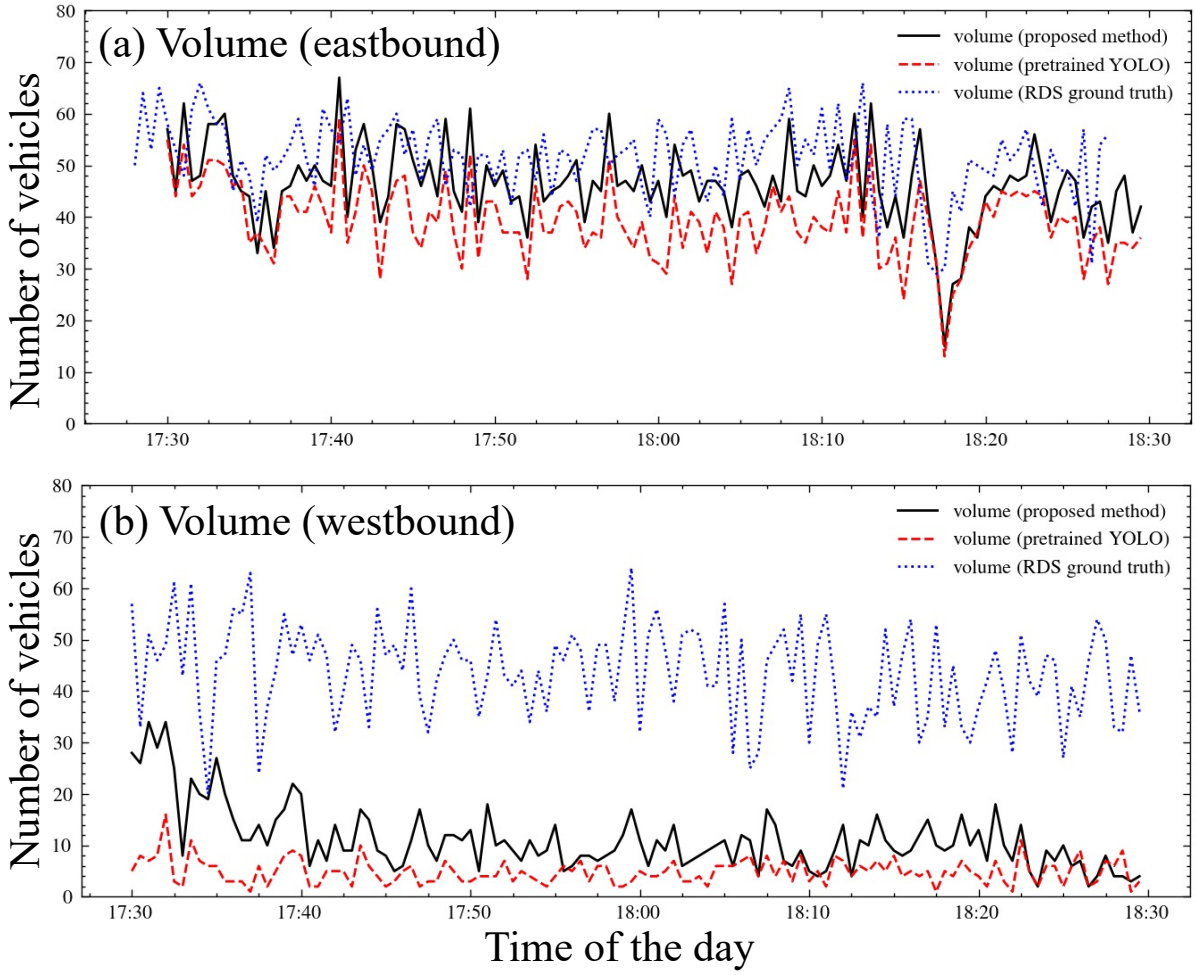


Figure 24. Comparison of Identified vehicles for east bound and west bound directions at the studied site.

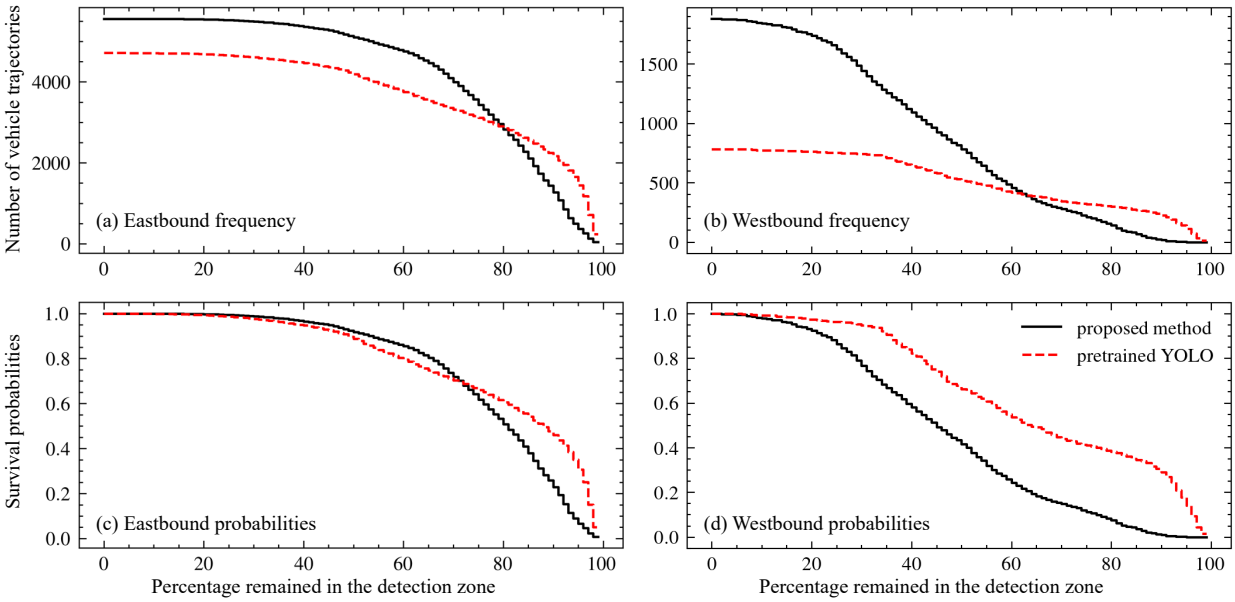


Figure 25. A quality check for the vertical length of identified trajectories within the detection zone.

curve shows that over 80 percent of vehicles can be tracked for 50% of the detection zone eastbound.

Finally, Table 10 summarizes the number of vehicles for each traffic direction using a naïve method. The details of the naïve method are outlined below. For each frame, the vehicle is identified as one out of the three classes (i.e., small vehicle, large truck, mid-truck). For a trajectory, the vehicle is classified at each frame. The mode of all classification results is used as the final class estimation. Results show that the naïve method recognizes most of the semi-trailer trucks. Note that the numbers in Table 10 correspond to trajectories passing a cordon line at the center of the transferred aerial view.

Conclusion

In conclusion, a new framework is proposed to tackle the challenging practical problem of identifying vehicles given low-resolution videos from CCTV cameras. The issue is essential since the low-resolution traffic video is one of the most publicly shared data. Compared with higher-resolution videos, low-resolution videos have advantages for data storage, transfer, streaming, etc. One core philosophy for designing a framework is that, unlike algorithms for autonomous driving, the composition approach of combining the good parts of multiple established models within an organized framework should be sufficient for extracting helpful information for understanding traffic situations. Under the resolution of 320×240 at 15 frames per second, the model can track most vehicles at the near side, and the speed output is also very accurate.

Many innovations can be used to train a system under the proposed framework. First, a YOLOv8s algorithm is retrained solely to identify vehicle objects. Using YOLOv8x pre-trained model as a teacher, training samples are generated for vehicles to train the YOLOv8s. The sole role of YOLOv8s is to identify vehicles without the burden of identifying specific vehicle types. Second, for an identified vehicle, another contrastive learning model is trained solely to learn the representations of each vehicle. With representations learned, new features are used to predict vehicle classes using a small prediction model. Finally, another advanced model, ByteTrack, is used to track the moving vehicles traveling along the road. Combining the good parts of each model, the proposed model gets a consistent performance in identifying, classifying, and tracking vehicles along the road. Finally, all identified vehicle trajectories are transferred to an aerial image based on the perspective knowledge of the camera to extract basic traffic information, including speed, volume, and vehicle class. Compared to the version solely using pre-trained YOLO, the fusion code runs faster because the YOLOv8s model is much smaller than YOLOv8x without sacrificing performance. The contrastive learning model is also fast by running on the pixels cropped by YOLO images, which is only sent to the contrastive learning model with a shape of 40×40 in width and height.

The potential applications of the proposed algorithms are immense. Theoretically, most of the cameras for traffic monitoring are low resolution because the camera usually needs to monitor traffic across many lanes in both directions. Thus, the number of pixels for each vehicle is generally small. With trajectories being identified, lots of other applications are now feasible. For

Table 10 A comparison of the number of large trucks (i.e., semi-trailer trucks) by applying a naïve classification algorithm.

	Ground truth	Identified (proposed algorithm)	Percentage identified
Eastbound (all)	6,191	5,559	89.8%
Eastbound (truck)	610	453	74.3%
Westbound (all)	5,239	1,881	35.9%
Eastbound (truck)	527	84	15.9%

example, vehicle trajectories can bring more information than traditional loop or radar detectors. On highways, for control strategies like various speed limits or ramp metering, it is vital to directly measure the number of queued vehicles or the fleet's density. A better automatic traffic monitoring algorithm is also helpful for safety studies. For example, it can monitor the case of intruding animals, identify stopped vehicles or fired vehicles, and generate reports to the traffic control centers. The potential benefits of alerting incidents seconds or minutes earlier are immense.

There are still some limitations to the current approach. Although the framework is suitable for transportation CCTV cameras, there is a potential for improvements for each specific module. One limitation is that the standard self-supervised learning SimCLR prefers color-to-grey image transformation to ensure the model doesn't rely on colors when identifying objects. In other words, it is color invariant. However, it is sometimes beneficial to consider color in our model as the number of pixels is limited. For example, most school buses are either yellow or white. Another significant issue is about the bounding boxes. One idea is that using the centroid of the bounding box is not enough to identify vehicle trajectories because the vehicle is a 3d object in a 2d perspective image. Like the I-24 Motion project, better identifying bounding boxes in 3 dimensions needs to be implemented and tested in the future. To extract traffic information, human decisions are necessary to divide vehicles by their travel lane for now. It is also possible to combine information from other models to automatically learn the background, road lane positions, and image perspectives using deep learning algorithms.

References

Abouelyazid, M., n.d. Comparative Evaluation of SORT, DeepSORT, and ByteTrack for Multiple Object Tracking in Highway Videos.

Al-qaness, M.A.A., Abbasi, A.A., Fan, H., Ibrahim, R.A., Alsamhi, S.H., Hawbani, A., 2021. An improved YOLO-based road traffic monitoring system. Computing 103, 211–230. https://doi.org/10.1007/s00607-020-00869-8

Bin Zuraimi, M.A., Kamaru Zaman, F.H., 2021. Vehicle Detection and Tracking using YOLO and DeepSORT, in: 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). Presented at the 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, Penang, Malaysia, pp. 23–29. https://doi.org/10.1109/ISCAIE51753.2021.9431784

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging Properties in Self-Supervised Vision Transformers.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A Simple Framework for Contrastive Learning of Visual Representations.

Cheng, F.-H., Chen, Y.-L., 2006. Real time multiple objects tracking and identification based on discrete wavelet transform. Pattern Recognit. 39, 1126–1139. https://doi.org/10.1016/j.patcog.2005.12.010

Girshick, R., 2015. Fast R-CNN.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2018. Mask R-CNN.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37, 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

Kapania, S., Saini, D., Goyal, S., Thakur, N., Jain, R., Nagrath, P., 2020. Multi Object Tracking with UAVs using Deep SORT and YOLOv3 RetinaNet Detection Framework, in: Proceedings of the 1st ACM Workshop on Autonomous and Intelligent Mobile Systems. Presented at the AIMS '20: The 1st ACM Workshop on Autonomous and Intelligent Mobile Systems, ACM, Bangalore India, pp. 1–6. https://doi.org/10.1145/3377283.3377284

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 84–90. https://doi.org/10.1145/3065386

Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Goncalves, G.R., Schwartz, W.R., Menotti, D., 2018. A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector, in: 2018 International Joint Conference on Neural Networks (IJCNN). Presented at the 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, Rio de Janeiro, Brazil, pp. 1–10. https://doi.org/10.1109/IJCNN.2018.8489629

Liang, S., Wu, H., Zhen, L., Hua, Q., Garg, S., Kaddoum, G., Hassan, M.M., Yu, K., 2022. Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles. IEEE Trans. Intell. Transp. Syst. 23, 25345–25360. https://doi.org/10.1109/TITS.2022.3158253

Lin, C.-J., Jeng, S.-Y., Lioa, H.-W., 2021. A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO. Math. Probl. Eng. 2021, 1–10. https://doi.org/10.1155/2021/1577614

Liu, C., Tao, Y., Liang, J., Li, K., Chen, Y., 2018. Object Detection Based on YOLO Network.

Liu, W., Ren, G., Yu, R., Guo, S., Zhu, J., Zhang, L., 2022. Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions. Proc. AAAI Conf. Artif. Intell. 36, 1792–1800. https://doi.org/10.1609/aaai.v36i2.20072

Mahmood, M.T., Ahmed, S.R.A., Ahmed, M.R.A., 2020. Detection of vehicle with Infrared images in Road Traffic using YOLO computational mechanism. IOP Conf. Ser. Mater. Sci. Eng. 928, 022027. https://doi.org/10.1088/1757-899X/928/2/022027

Padilla Carrasco, D., Rashwan, H.A., García, M.Á., Puig, D., 2023. T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks. IEEE Access 11, 22430–22440. https://doi.org/10.1109/ACCESS.2021.3137638

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement.

Redmon, J., Farhadi, A., 2016. YOLO9000: Better, Faster, Stronger.

Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition.

Terven, J., Córdova-Esparza, D.-M., Romero-González, J.-A., 2023. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr. 5, 1680–1716. https://doi.org/10.3390/make5040083

Wojke, N., Bewley, A., Paulus, D., 2017. Simple Online and Realtime Tracking with a Deep Association Metric.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X., 2022. ByteTrack: Multi-Object Tracking by Associating Every Detection Box.

CHAPTER III. ANALYZING TRUCK'S ACTIVITY PATTERNS BY IMPUTING POINT DETECTORS TO TRAFFIC NETWORKS USING TMAS DATASET

Abstract

Understanding the dynamics of truck volumes and activities across the skeleton traffic network is pivotal for effective traffic planning, traffic management, sustainability analysis, and policy making. Yet, relying solely on average annual daily traffic volume for trucks cannot capture the temporal changes over time. Recently, the Traffic Monitoring Analysis System dataset has emerged as a valuable resource to model the system by providing information on an hourly basis for thousands of detectors across the United States. Combining the average annual daily traffic volume from the Highway Performance Monitoring System and the Traffic Monitoring Analysis System dataset, this study proposes an elegant method of imputing information across the traffic network to generate both truck volumes and vehicle class distributions. Specifically, the proposed method uses average annual daily traffic values as an initial assumption, iteratively refining Traffic Monitoring Analysis System information across neighboring traffic links to predict traffic information over the network. A series of experiments evaluated the model's performance on various spatial and temporal scales. Results showed that the performance of the model achieved an R^2 of more than 0.56 for truck volume, outperforming benchmark methods such as the Kriging method. In summary, the proposed method underscores the importance of using network topologies and integrating diverse data sources to enhance our understanding and modeling of traffic activities.

Keywords: AADT, TMAS, vehicle class estimation, traffic flow estimation, data imputation, missing data

Introduction

Understanding the dynamics of freight flows is vital not only for effective transportation planning and management but also for assessing their implications on energy consumption and environmental sustainability. Recent advancements in the field have yielded a suite of tools tailored for the analysis of freight emissions analysis (Shiledar et al., 2024). Additionally, the emergence of electric vehicles has been touted for their potential economic and environmental advantages, particularly in energy conservation (Weiss et al., 2020). Consequently, there exists a pressing demand to systematically evaluate the efficacy of various policies and investments within the trucking and freight industries. However, a considerable portion of existing research relies on synthetic, simulated, or proprietary trucking data, underscoring the necessity to develop a data-centric approach to comprehensively model truck flows and activities over the traffic network.

To analyze traffic patterns in the United States, researchers often use the Highway Performance Monitoring System (HPMS) dataset, a widely recognized source of information. Alongside fundamental roadway characteristics, the US Department of Transportation mandates the reporting of annual average daily traffic (AADT) figures annually. Ideally, a comprehensive assessment would involve daily traffic counts at each location throughout the year to derive accurate average traffic volumes. However, practical constraints often limit this approach, particularly for smaller roads where AADT values are typically approximated through a 3-day observation period within a given year. Usually, a limited number of detectors are rotated across smaller road segments over the course of the year. Subsequently, the observed values undergo correction processes to determine AADT, accounting for various factors such as seasonal variations and weekday fluctuations. AADT values serve as crucial metrics for federal, state, and

metropolitan planning agencies (Krile et al., 2016). Nonetheless, it is still necessary to enhance the precision of daily, weekly, and monthly traffic fluctuation modeling through more refined observational and analytical techniques.

To address the imperative for a more detailed nationwide traffic flow database, the Traffic Monitoring Analysis System (TMAS) dataset has been established to systematically monitor traffic patterns across the United States ("2022 Traffic Monitoring Guide," 2022; Krile, 2016). Predominantly derived from sensors strategically positioned along major highways or arterials, the TMAS dataset comprises three primary subsets: (1) the volume dataset, detailing the count of vehicles traversing a designated point along a roadway in a consistent direction; (2) the class dataset, elucidating the distribution of vehicle classes; and (3) the weight dataset, delineating the distribution of vehicle weights.

Figure 26 illustrates the geographical distribution of traffic observation stations according to the TMAS dataset compiled in 2021. The dataset comprises approximately 4,558 stations for measured traffic volume and 3,573 stations for vehicle class assessments. Each yellow dot depicted in Figure 26(a) represents a site for volume observation, whereas each green dot in Figure 26(b) indicates a site for vehicle class observation. The geographic locations of weight stations are not shown because they are out of the scope of this study. For this study, only the class datasets were used for analysis, providing insights into traffic volume across the Federal Highway Administration's 13 vehicle classes on an hourly basis (Pagan-Ortiz, 2014). Note that some states did not report class datasets to TMAS for 2021, but all states uploaded their volume information.

Although TMAS data offer extensive coverage of geographical areas, their recordings are confined to predetermined locations or points at the traffic networks. In this investigation, our aim was to estimate traffic volume across the entire traffic network by leveraging TMAS recordings from fixed points. Specifically, the research team conducted imputation on the Freight Analysis Framework (FAF) networks as delineated in previous studies (Hwang et al., 2021, 2016; Department of Transportation et al., 2017). Additionally, we integrated AADT truck volume reported from the HPMS into the FAF network as weighting factors to propagate information over the network.

Figure 27 illustrates a simplified example to demonstrate the process of imputation on a grid network with directed edges, originating from the lower left corner and extending to the upper right corner. Truck percentages are set at 10% and 100% at the starting and ending edges, respectively. Initially, all nodes within the grid network, except for the starting and ending edges, lack information. Using an iterative algorithm, missing data within the grid network can be gradually imputed by iteratively running the algorithm multiple times. As depicted in the center of Figure 27, which represents 100 iterations, the proportion of trucks steadily transitions from 10% to 100%. Subsequently, the results converged after about 200 iterations, as shown on the right of Figure 27. Through the implementation of the algorithm, inferring the distribution of vehicle classes becomes feasible.

(a) Locations of detectors generating traffic volume (b) Locations of detectors identifying volume by vehicle class

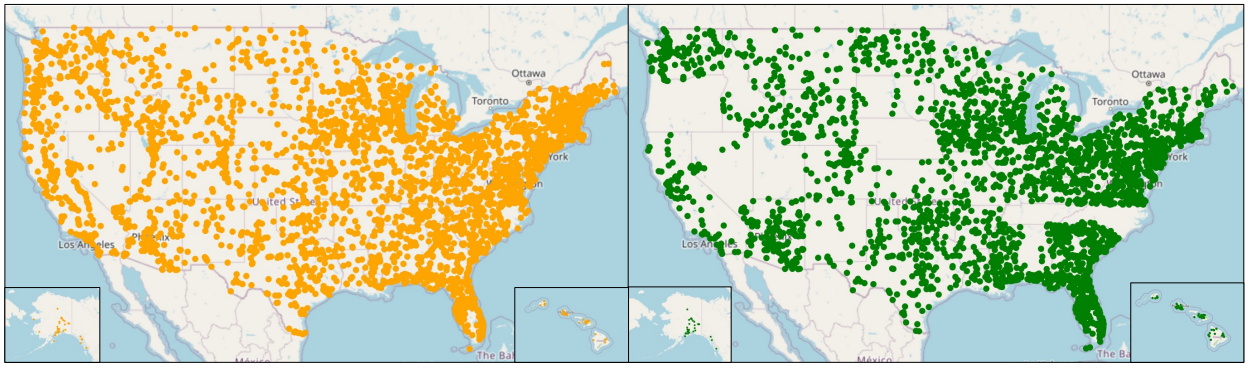


Figure 26. Locations of TMAS detectors for both (a) volume observation and (b) vehicle class observation.

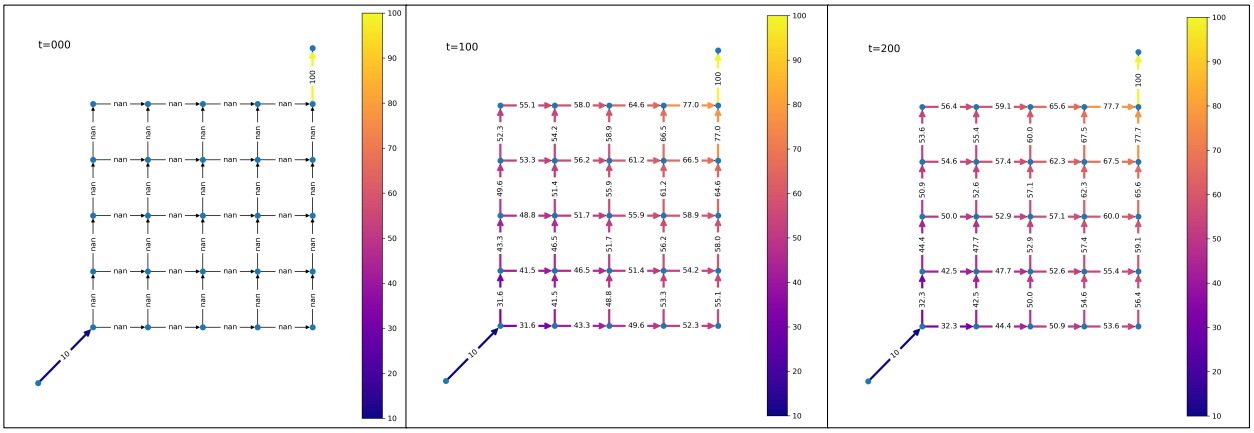


Figure 27. A simple example of imputing missing traffic volume in a grid network traveling from bottom left to top right.

The remainder of this work is structured as follows: Section 2 provides a comprehensive review of prior methodologies and approaches in traffic volume estimation. Section 3 presents the formulation of the proposed algorithm along with associated data processing steps necessary to facilitate its implementation. Section 4 presents the background and outcomes of a series of experiments conducted to assess efficacy of the algorithm from diverse analytical viewpoints. Finally, Section 5 concludes this study by delving into its contributions, potential implications, inherent limitations, and avenues for future research.

Literature Review

Traffic detectors are conventionally positioned at various intervals along roadways, necessitating the estimation or interpolation of traffic data across the entire network. Prior studies have predominantly used statistical methodologies to forecast the AADT values of roads, drawing upon factors such as roadway characteristics, socioeconomic indicators, and proximity to highways (Anderson et al., n.d.; Caceres et al., 2018; Chang and Cheon, 2019; Mohamad et al., 1998; Wang and Kockelman, 2009; Zhao and Chung, 2001). These regression techniques, termed *aspatial methods*, neglect the topology of the traffic network, thereby presenting certain limitations. Moreover, to improve performance, these methodologies need special socioeconomic data collection for each localized analysis zone within a metropolitan area or city. They rely solely on statistical modeling by using input features and do not integrate roadway geometries or network topologies.

In addition to the aspatial methods, various approaches exist for estimating roadway information by leveraging data from neighboring roadways. One prominent technique is geographically weighted regression (GWR), which entails computing a weighted average that considers geographical proximity. GWR has widespread application across diverse domains, including traffic safety analysis and traffic management (Gu et al., 2023; Hezaveh et al., 2019; Xu et al., 2020), and has been adapted for estimating AADT values (Liu et al., 2019; Zhao and Park, 2004). The limitation of using the GWR for estimating AADT is that its performance might not be as effective in urban areas as opposed to rural areas.

Besides the GWR method, Kriging methods are also employed for traffic flow imputation. Originally devised for spatial interpolation of mining resources based on sampled points (Gotway et al., 1996), Kriging has been adapted for transportation applications to interpolate values along traffic corridors and networks (Lowry, 2014). An extended Kriging method is used in transportation studies to impute values in spatiotemporal dimensions across traffic networks (Bae et al., 2018; Yang et al., 2018).

More recently, graph neural networks have gained traction for modeling traffic based on network structures in various applications, including traffic prediction (Bui et al., 2022; Jiang and Luo, 2022; Li and Zhu, 2021), data imputation (Kong et al., 2023; Liang et al., 2022; Shen et al., 2023), event identification (Zhang et al., 2020), and so forth. Despite claims of state-of-the-art performance, graph neural networks exhibit limitations, notably a focus on metropolitan areas rather than statewide or nationwide analyses, in which detector densities may be insufficient to adequately capture the roadway geometry. Moreover, training such models at state or national levels may become very computationally intensive.

Considering the constraints elucidated in previous studies, the reminder of this work will introduce the proposed methodology and analytical framework with the following objectives: (1) to devise an efficient method for imputing traffic data throughout a traffic network, (2) to construct a versatile analytical framework capable of data imputation at varying temporal aggregation levels as desired, and (3) to thoroughly analyze the advantages and limitations of the model across different scenarios and potential usages.

Methodology

An overview of the data analysis framework

To systematically address the issue, we proposed an analytical framework comprising four primary stages, as shown in Figure 28. The first step was the generation of AADT values and their alignment with the FAF traffic network. The second step was the delimitation of the temporal scope of interest and aggregation of TMAS class datasets to acquire freight volume and class distribution insights at TMAS locations. Additionally, the second step included the correlation of these data with the nearest directional link within the FAF traffic network. The third step included comprehensive integration of all gathered information to impute TMAS data across the entire network for both volume and class distribution. Finally, the fourth step included implementation of cross validation techniques involving the masking of selected TMAS stations as validation sets, followed by a comparison of predicted values against observed ones within these validation sets. Furthermore, the study's outcomes could extend beyond the discussed parameters, offering potential benefits for various ancillary analyses such as emission modeling and activity pattern analysis through the provision of these more informed, data-driven estimations.

Integration of AADT values and TMAS information into a cohesive network representation is imperative for effective imputation. We employed the FAF network to amalgamate these datasets to form one graph representing all information. Although the HPMS network is another available option, the FAF network offers a more realistic geometric framework characterized by intricate geometries alongside a streamlined set of traffic links.

Figure 29 presents a visual representation of the differences between the two networks. The upper left histogram delineates the distribution of link lengths within the HPMS network across the United States, whereas the lower left histogram illustrates the distribution of link lengths within the FAF network across the same geographical scope. Note, in this study, most links of the HPMS network were under 1 mile, whereas a significant proportion of links within the FAF network was more than 1 mile. The map displayed on the right of Figure 29 exemplifies a typical instance of the divergence between the two networks. The green dots symbolize the end nodes of the TMAS network's links, whereas the blue dots signify those of the FAF network. Note, the HPMS network (green dots and links) was deliberately partitioned into smaller units. Consequently, a solitary FAF link may correspond to multiple HPMS links in practical application. In summary, the FAF network exhibited greater comprehensiveness with a diminished quantity of links but featured geometries that align more closely with real-world scenarios.
Table 11 further compares the disparities between the FAF network and the HPMS network regarding the quantities of links and nodes, the distribution of link lengths, and other parameters.

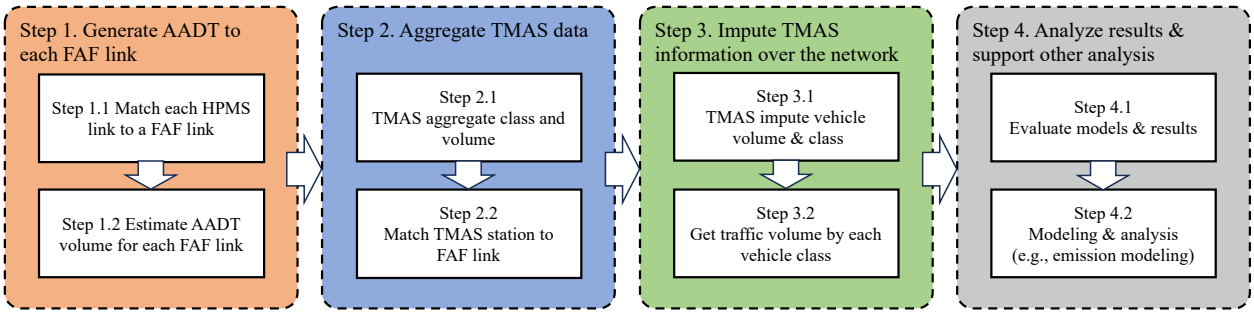


Figure 28. The proposed analysis framework for estimating vehicle volume and class distributions along traffic networks based on AADT values and TMAS datasets.

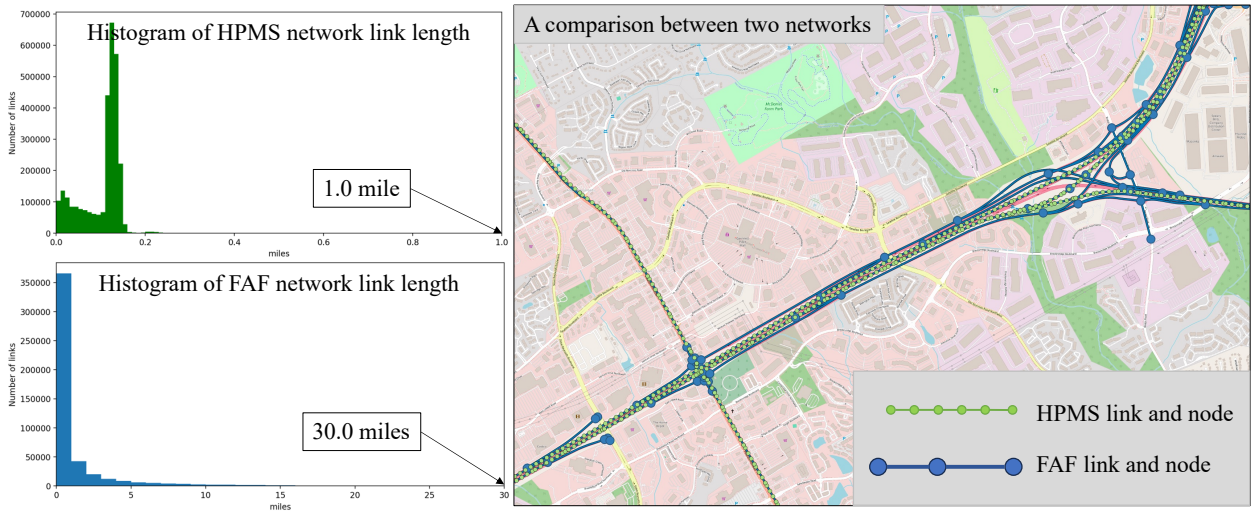


Figure 29. A comparison of two networks: HPMS and FAF. FAF networks are more comprehensive and tailored for freight traffic analysis.

Table 11. The overall structure of the network

Network	Number of links	Total length (miles)	25% link length	50% link length	75% link length
HPMS (National Highway System)	3,017,746	377,673	0.078	0.121	0.132
FAF	487,394	766,646	0.161	0.376	0.997

Note, the median distance among FAF links measured 0.376 miles, whereas the median distance for HPMS links was markedly lower at 0.121 miles. Consistent with the observations, as shown in Figure 29, the FAF network was more comprehensive compared with its HPMS counterpart. Section 0 provides further discussion on the procedural intricacies involved in aligning the HPMS network with the FAF network.

The second phase of the analysis was dedicated to obtaining aggregated TMAS data. Given that the TMAS dataset encompasses records for every hour throughout the year, researchers can specify their time window of interest before conducting aggregation. Table 12 describes several typical temporal intervals of interest. For instance, analysts may opt to scrutinize monthly traffic patterns by aggregating data monthly. Similarly, aggregation based on different times of the day or days of the week is feasible. Moreover, AADT values for trucks can be approximated by aggregating data spanning a full year. In essence, analysts can configure temporal intervals for aggregation in accordance with their research objectives.

Furthermore, the TMAS dataset encompasses traffic volume records categorized by vehicle class, enabling granular analysis at the level of individual vehicle classes. Within this study, the proportional representation of each vehicle class was carefully documented. Subsequently, employing spatial proximity and directional alignment, each TMAS station was precisely matched with the nearest link within the FAF network, a process elaborated on in Section 0.

In the third phase, the aggregated traffic volumes were converted into probabilities before the imputation process across the FAF network. Within the FAF network, the truck's AADT volume served as foundational data guiding the imputation and distribution of values throughout the network. Section 0 explains the proposed algorithm in detail.

After running the imputation, Phase 4 involves a wide range of quantitative and qualitative tools to thoroughly analyze the advantages and limitations of the tool. Most of the experiments and evaluations were based on cross validation methods, as described in Section 0.

Estimating AADT values by matching HPMS links to FAF links

As outlined in Section 0, the imputation algorithm was exclusively executed within the framework of the FAF network because of its superior geometric properties and enhanced comprehensiveness. However, the AADT values derived from the HPMS dataset were contained within their own HPMS network. To transfer the AADT volume information from the HPMS network to the FAF network, the development of a matching algorithm was needed to align the respective outcomes.

Before the introduction of the matching algorithm, it was essential to assess the AADT value for each FAF link using AADT values derived from HPMS links. We observed the value against two networks (e.g., the right map of Figure 29), and Figure 30 illustrates a representative case comparing two networks. As observed earlier, the FAF network delineates network geometries, whereas the HPMS network comprises numerous smaller segments with reduced geometric intricacies. Additionally, each HPMS link represents two directions unless the road is one way. Conversely, within the FAF network, each link of interstate highways solely represents traffic flow in a single direction.

Table 12. Aggregation options involved in the study and provided by the analysis pipeline

Aggregation level	Description
Vehicle class	Aggregate probability by each vehicle class
All time	Aggregation observations over the whole year
Time of the day	Aggregate observations in different periods of the day; specifically, in this study: 1. T0: Observations between 22:00 and 4:00 2. T1: Observations between 4:00 and 10:00 3. T2: Observations between 10:00 and 16:00 4. T3: Observations between 16:00 and 22:00
Weekday/weekend	Aggregate observations by weekdays and weekends
By month	Aggregate observations by different months

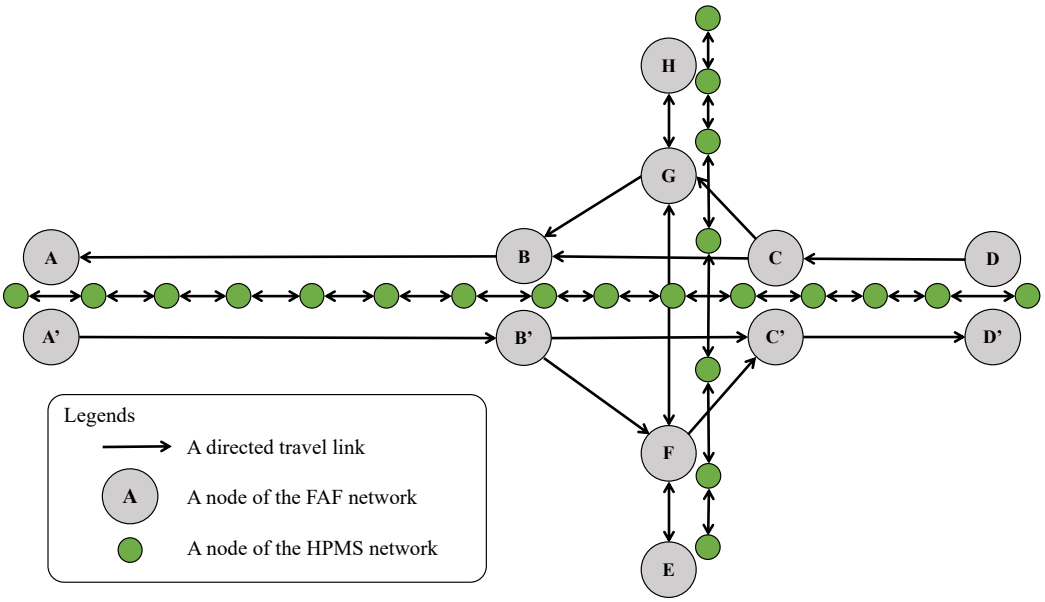


Figure 30. A typical local representation of an interstate segment with an overpass arterial using FAF and TMAS networks.

Figure 31 shows the six main steps for aligning HPMS links with FAF links. First, both networks were converted into directed graphs, wherein each edge represented a single travel direction. Given that AADT quantifies traffic volume bidirectionally, the metric was halved to reflect unidirectional traffic volume. Subsequently, a composite series of geographical operations was performed to ensure the fidelity of AADT matching outcomes.

The second stage allowed for creation of a buffer zone around the HPMS link, and all intersecting HPMS edges were systematically collated for each FAF link buffer. Subsequently, during Stage 4, only those HPMS edges demonstrating approximate alignment with the prevailing traveling direction (±15°) were retained. In Stage 5, the median HPMS value served as the reference HPMS value. Finally, in cases of FAF links without corresponding HPMS links, volume information was effectively imputed using the proposed algorithm.

Upon generating precise estimates of AADT for each FAF edge, data generated from TMAS stations were matched to the most reasonable FAF edge, considering spatial proximity and travel direction. Figure 32 illustrates the process of aligning a TMAS station with FAF network links. It was imperative to discern between the two opposing travel directions, even though they shared identical TMAS station identifiers, and align them with their respective directional links. After the procedures of AADT estimation and the spatial alignment of TMAS stations with the nearest links, the FAF network comprehensively incorporated all pertinent information, encompassing AADT values, TMAS data recordings, and other relevant metrics. Because the algorithm was running at a nationwide scale and the density of detectors was low, an edge could be used to store information for TMAS stations at a fixed point.

Imputing values over the network with prior weights

As described earlier, the FAF network was converted into a directional graph; each link corresponded to a specific travel direction, and each node established connections with two or more links to constitute a traffic network. Following the construction of such a network, values were interactively imputed across the network, as illustrated in Figure 33. The link was designated as the target for estimating imputed values. Initially, all incoming links to node C (i.e., $\overrightarrow{AC}$ and $\overrightarrow{BC}$) and outgoing links from node D (i.e., $\overrightarrow{DE}$ and $\overrightarrow{DF}$) were identified as neighboring links of $\overrightarrow{CD}$ (depicted in red in Figure 8). Subsequently, the values of $\overrightarrow{CD}$ were determined as the weighted average of the neighboring links, following the formula

$$\hat{y}_i = \frac{\sum_j w_j \hat{y}_j}{\sum_j w_j}, \forall\, y_j \in \{y_j : L_i \; and \; L_j \; are \; neighboring \; links\}, \quad (1)$$

where L_i represents the target link for imputation (e.g., $\overrightarrow{CD}$); L_j denotes the neighboring links of L_i; and $\hat{y}_i$ and $\hat{y}_j$ signify the information to be imputed or estimated within this context, respectively. In practical application, $\hat{y}_i$ could signify the probability associated with a specific vehicle class, such as the probability of Vehicle Class 5, or represent traffic volume. Furthermore, w_j refers to the weight attributed to each neighboring link. In the framework of this investigation, w_j represents the truck AADT value of a link, derived through the methodology outlined in Figure 31.

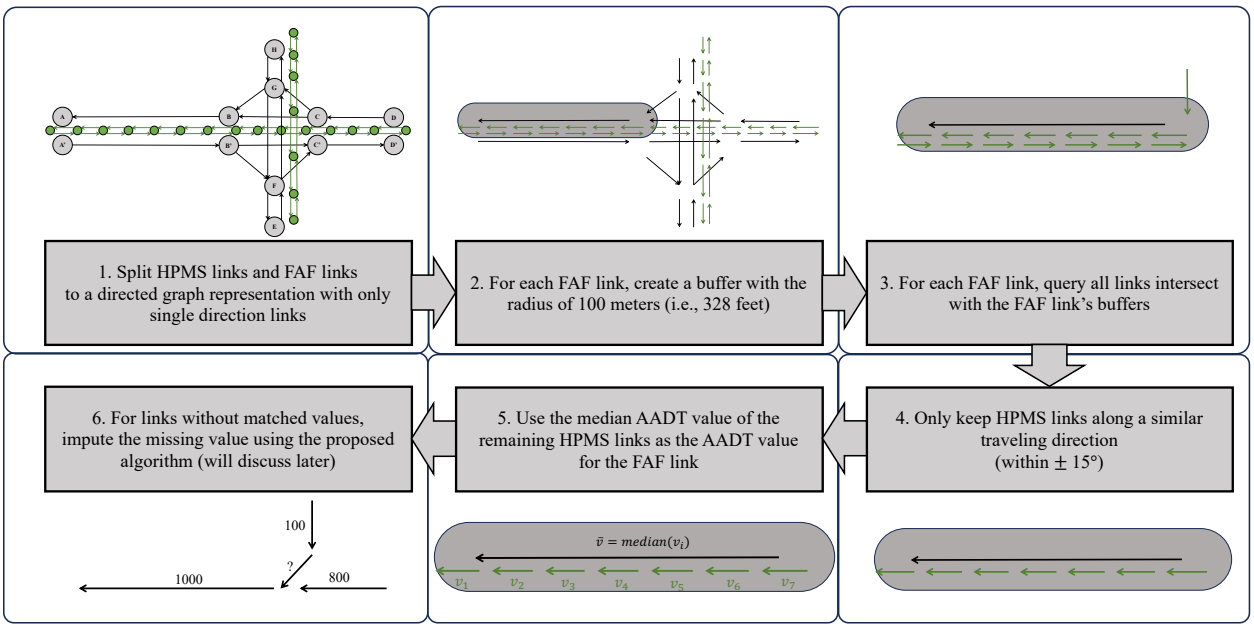


Figure 31. The procedure of estimating AADT values from HPMS networks.

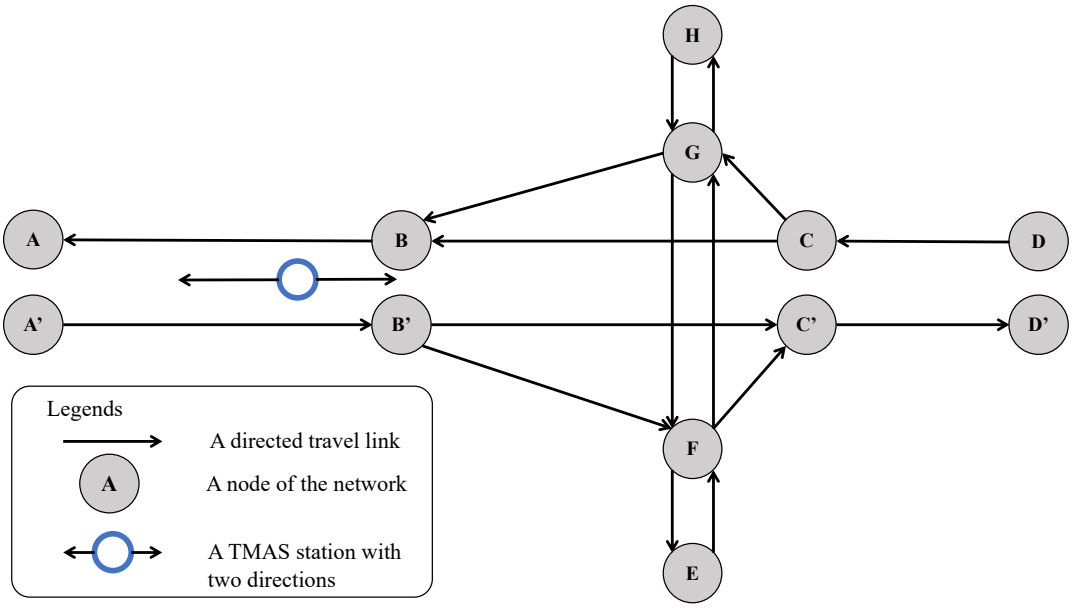


Figure 32. A typical local representation of a highway segment with an overpass arterial network. A TMAS station (the blue circle) is located along the interstate highway.

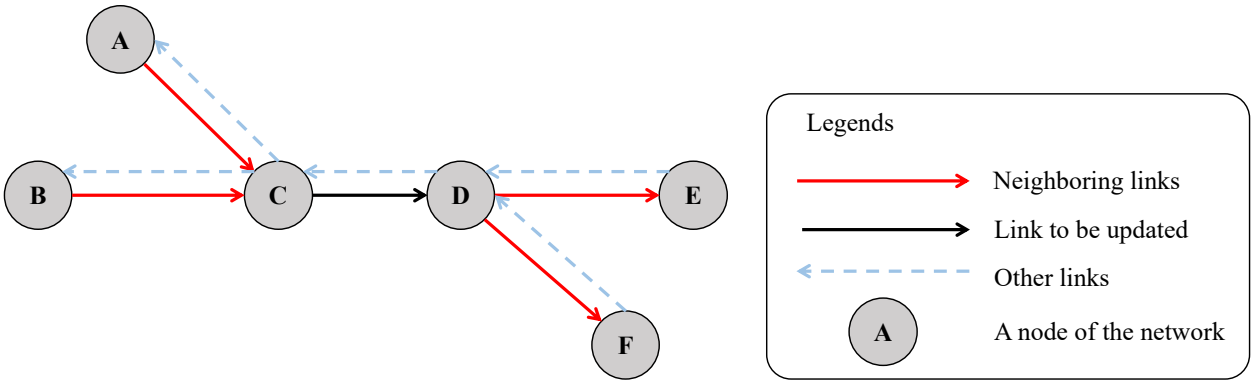


Figure 33. Graph representing a local road segment to demonstrate the neighboring links (in solid red) of a link of interest (in solid black).

Consequently, when imputing a probability distribution across the network, the target link's value is updated as a function of the weighted average of neighboring links, with weights determined by truck AADT volume. For the final step outlined in Figure 31, in instances where AADT values were either missing or deemed unrealistic, a similar imputation approach was employed, using the truck volume of the FAF network as link weights to estimate their AADT volumes, given existing neighboring links.

In addition to Equation (1), which serves as the foundational framework for imputing traffic values, the algorithm operated by systematically enumerating each link within the network, incrementally assigning values to links in a stepwise manner. Through repetition of this process, the algorithm converged toward a state in which link volumes approach equilibrium. Drawing inspiration from the renowned shortest path searching methodology elucidated in the Bellman–Ford algorithm (Dasgupta and Papadimitriou, 2006), a novel approach for traffic imputation was introduced, accompanied by the corresponding pseudocode presented in Figure 36.

The algorithm abstained from imputing a value if it was already an observed value corresponding to a TMAS station, according to the condition "if $e.y$ is observed value." The parameter K regulated the number of iterations the algorithm undergoes. The "valid_to_impute" function in Line 6 of the pseudocode described two exceptional scenarios in which imputation could not proceed until all links had been assigned values either upstream or downstream. One such scenario involved a merging link with multiple inbound links but only one outbound link.

The other scenario entailed multiple outbound links but only one inbound link. These instances represent specialized cases in which imputation is deferred until either all inbound (upstream) links or outbound (downstream) links are resolved. Note, all algorithms employed in this study are implemented using Python, whereas the graph of the FAF network leverages the NetworkX package (Hagberg et al., 2008).

Cross validation and evaluation metrics

Quantitative methodologies are essential for assessing the efficacy of the proposed algorithm. Specifically, a tenfold cross validation technique was employed for model evaluation. Figure 35 describes the steps involved in the cross-validation process to assess the algorithm's performance. All TMAS stations were randomly partitioned into 10 subsets. Within each iteration of imputation, one data subset comprising 10% of TMAS stations was masked (depicted by blue rectangles in the figure). The remaining 90% of detectors were used to propagate values across the network. Performance was gauged by comparing the observed and predicted values for the masked validation set. To evaluate the model performance, four primary metrics were used: mean absolute error (MAE), root mean squared error (RMSE), Pearson's R^2 values, and cross entropy loss (CEL). Additionally, further visual investigation tools, such as scatterplots and probability distribution maps, were employed to comprehend the model performance and the algorithm's behavior.

Three metrics were applied to evaluate the performance of traffic volume, including MAE, RMSE, and Pearson R^2 values. These three metrics are defined as

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N}, \quad (2)$$

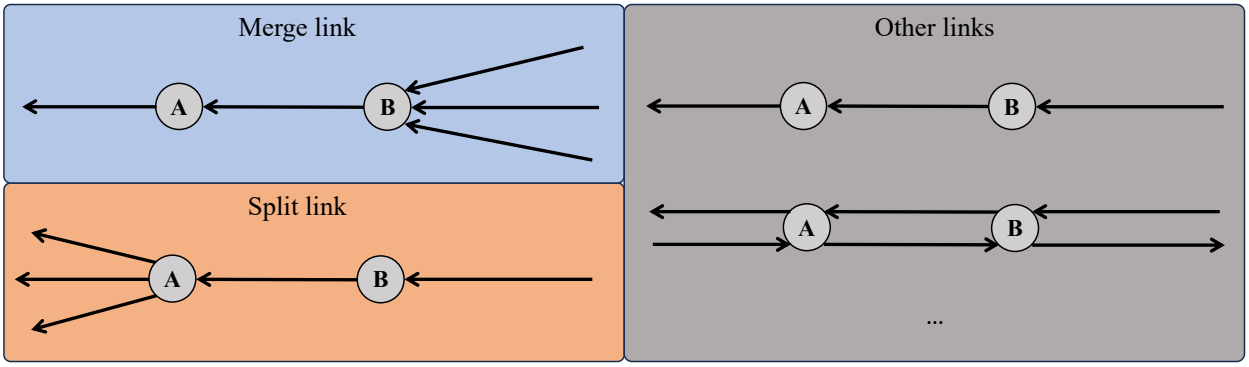


Figure 34. Two special kinds of traffic links with special rules for imputation.

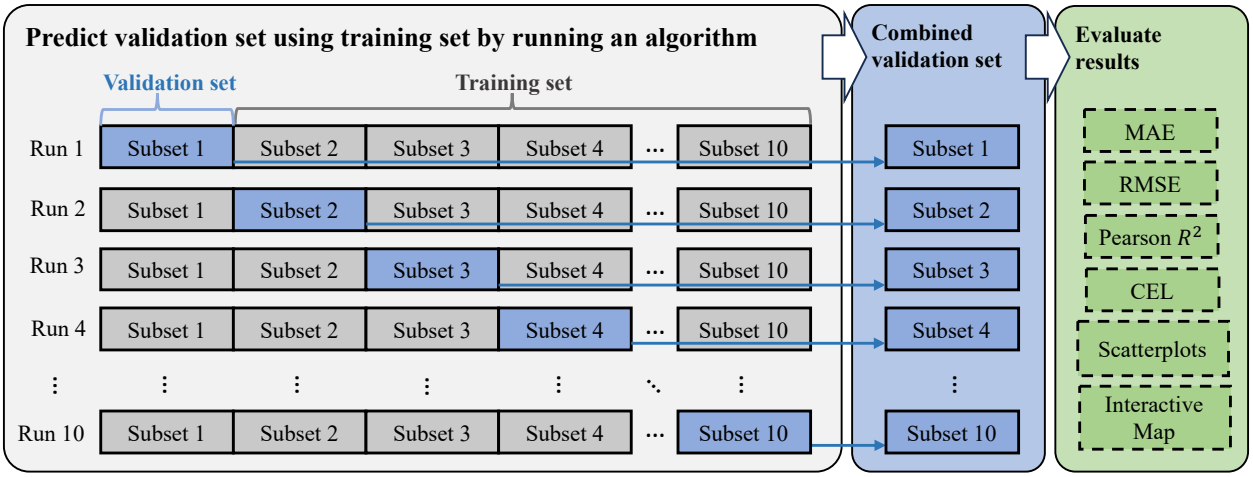


Figure 35. A visualization of algorithm evaluation using cross validation.

Algorithm 1: Iteratively imputing traffic information over the network

Input: Graph G, which contains AADT values for each edge e, observed link values for the corresponding edges; K is the number of epochs to impute data
Output: imputed values for all links in G
1: For i in {1, …, K}:
2: For each edge e in G:
3: if $e.y$ is observed value:
4: continue
5: ns = get_neighboring_edges(e, G)
6: if valid_to_impute(e, ns, G):
7: $e.y$ = impute_edge(e, ns, G)

Figure 36. The pseudocode of the graph algorithm running the imputation process.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}, \text{ and} \tag{3}$$

$$R^2 = \rho_{y_i,\hat{y}_i}^2 = \left(\frac{cov(y_i,\hat{y}_i)}{\sigma_{y_i}\sigma_{\hat{y}_i}}\right)^2, \tag{4}$$

where N is the number of records and y_i and $\hat{y}_i$ are the observed and predicted values, respectively. Meanwhile, $cov(y_i, \hat{y}_i)$ is the covariance between two series of data, and σ_{y_i} and $\sigma_{\hat{y}_i}$ correspond to the standard deviation of observed (i.e., true) values and predicted values, respectively.

In addition to the three metrics in Equations (2)–(4), a fourth metric, CEL, was used to quantify the disparities between the predicted and actual observed distribution of vehicle classes. In this study, CEL was used to measure the distance between two distributions:

$$CEL(f,g) = -\sum_{i=5}^{13} f(i = c)\log(g(x = i)) \tag{5}$$

where $f(i)$ and $g(i)$ are the observed and expected probability distributions for different vehicle type is, respectively.

Results

Data preparation: data selection and data quality check

To investigate the proposed method, our analyses relied on the 2021 TMAS dataset. For simplicity, only the class dataset was considered in this study, whereas the weight and volume datasets were excluded from our current framework. Consequently, all findings presented here are predicated solely on the class dataset.

A quality assessment was conducted to evaluate the availability and variability of the TMAS class dataset throughout the year, as depicted in Figure 37. The red dashed line delineates the average percentage of hours each state reported data per month. Most states exhibited a relatively

high and consistent percentage (above 75%) of records generated, with the exception of South Dakota.

Concurrently, the blue line in Figure 37 illustrates the average truck volume normalized by the number of TMAS stations, with a scale ranging from 0 to 100 vehicles per hour. Illinois exhibited the highest observed traffic volume throughout the year, followed by Louisiana, Texas, Alabama, and California, indicating the operational intensity of each station on average. The volume was standardized from 0% to 100% to facilitate comparison of variations across different months, denoted by the green line. The busiest month was designated as 100%. Although many northern states displayed a seasonal trend with significantly reduced traffic volume during winter months, the overall trend remained relatively stable throughout 2021.

TMAS data quality should be checked, and the volume inputs for the FAF network's edge weights should be reasonable. Figure 38 shows the AADT truck volume over the maps for Texas and Georgia. The color bar represents the truck AADT volume ranges from 0 to 10,000. A visual check over different places showed that the truck AADT flow trend was reasonable.

Impute traffic volume by each traffic class for each TMAS station

As previously discussed, a tenfold cross validation procedure was conducted to evaluate the model. In each iteration, 10% of TMAS stations were masked as validation sets. Subsequently, the imputed values corresponding to these validation sets were compared with the observed values for model assessment.

Figure 39 depicts the scatterplot analysis of the TMAS detectors. For each plot, the *x*-axis represents the observed values, whereas the *y*-axis denotes the predicted values. A reference line at a 45° angle illustrates the ideal scenario where predicted values align perfectly with observed values. The right subplot presents the aggregate volume data for all vehicles, and all subplots are visualized using a logarithmic scale. The logarithmic scale manifested an ellipsoidal pattern around the 45° line, suggesting the method's viability. The nine smaller subplots on the left depict scatterplots for individual vehicle classes ranging from 5 to 13. Notably, most large vehicles were either Class 5 or Class 9. Class 5 corresponded to two-axle trucks, such as single-unit trucks, whereas Class 9 comprised five-axle trucks, such as single-trailer trucks.

Breakdown traffic class distribution by different states

To comprehensively assess outcomes across various vehicle classes, it is advantageous to disaggregate the findings based on geographic locations, given that each state contributes its distinct dataset to the TMAS database. Figure 40 illustrates this disaggregation, presenting results segmented by individual states through a geo-facet plot. Each subplot corresponds to a specific state, identified by its abbreviation at the top. Within each subplot, three horizontal bars are depicted vertically in blue, green, and red, representing the R^2 values for all vehicle classes, the MAE for Class 5, and the MAE loss for Class 9, respectively. States shown without data did not submit information to the TMAS dataset. Additionally, several states—including New Mexico, Kansas, Alabama, and the District of Columbia—demonstrated comparatively poorer performance than others.

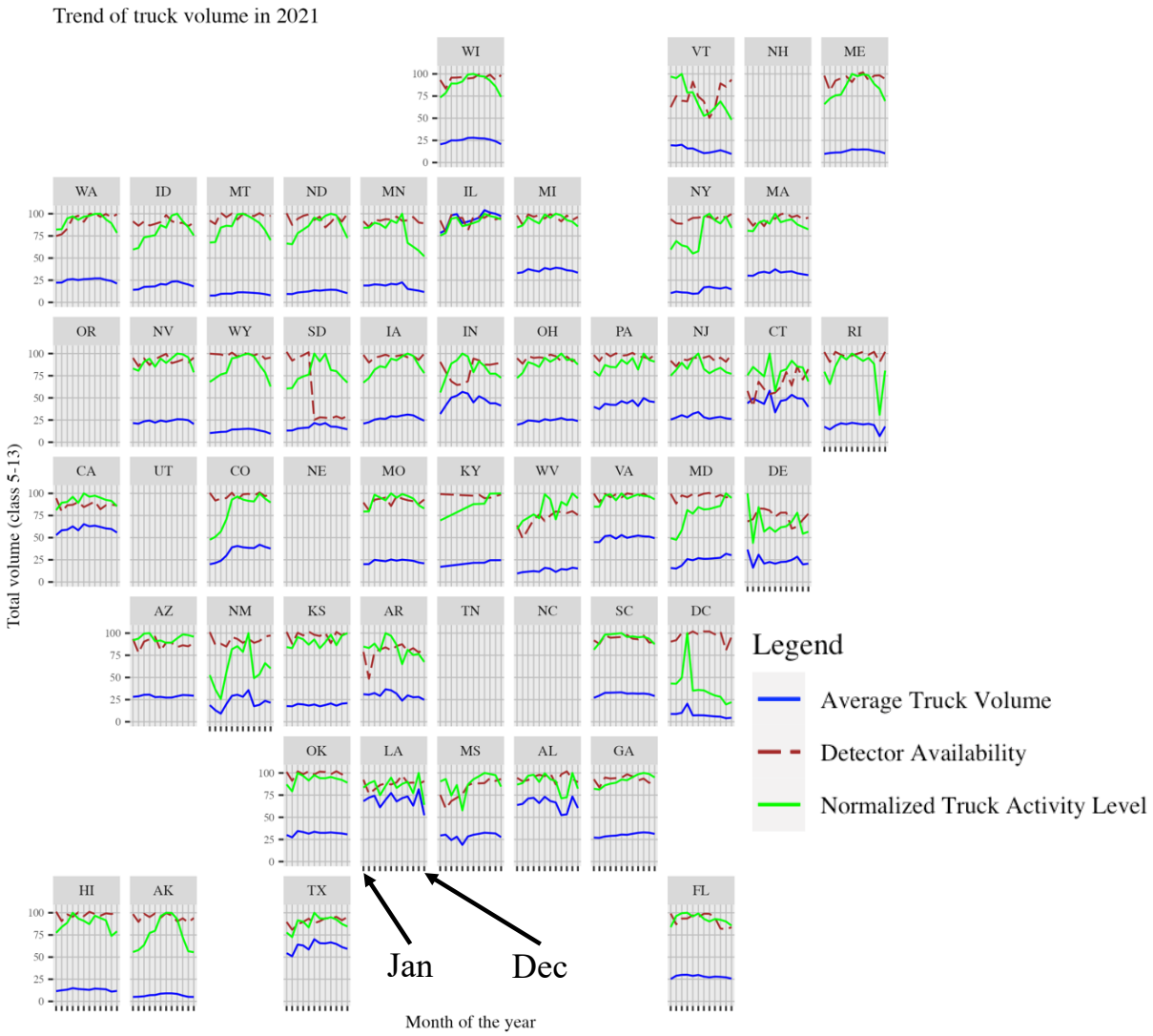


Figure 37. Quality check of detector availability, average truck volume (normalized between 0 and 1), and truck activity level for each month for each state.

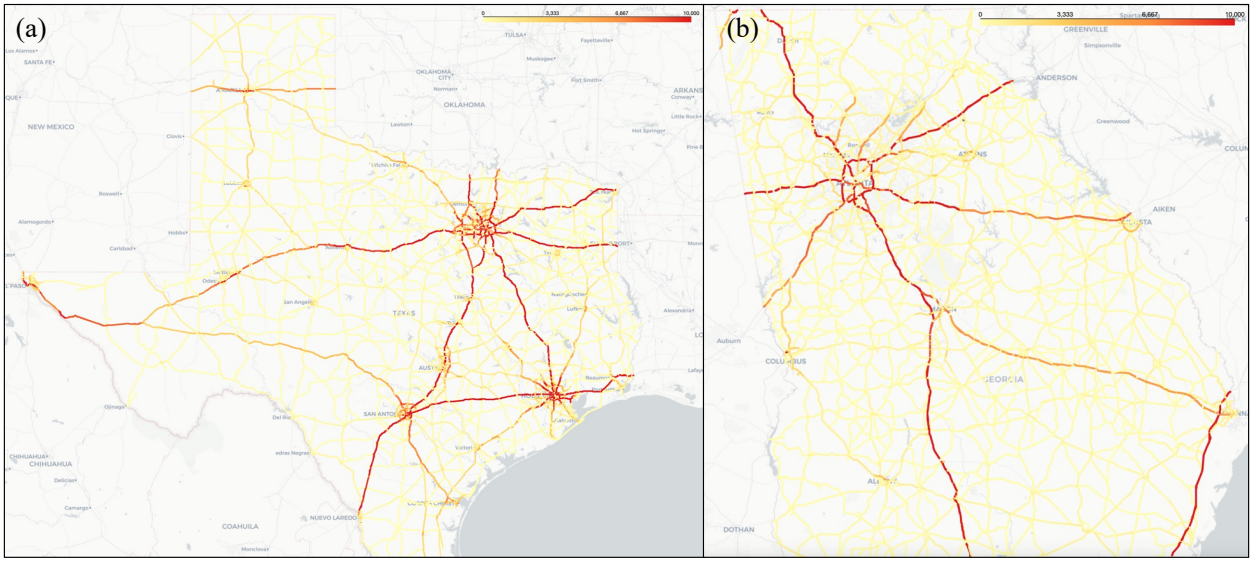


Figure 38. A visualization of directional AADT of trucks for (a) Texas and (b) Georgia. The AADT values are matched to FAF links from HPMS links.

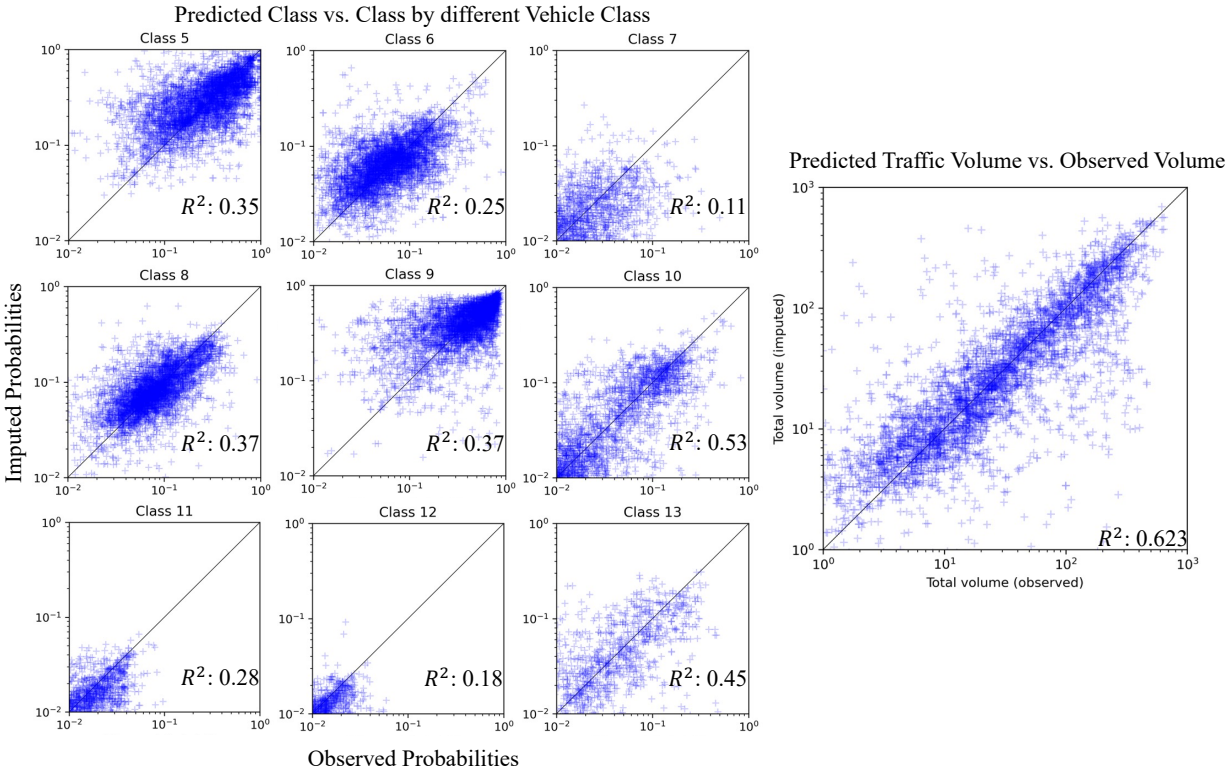


Figure 39. Analysis of the R^2 of detectors of traffic volume from Vehicle Classes 5–13.

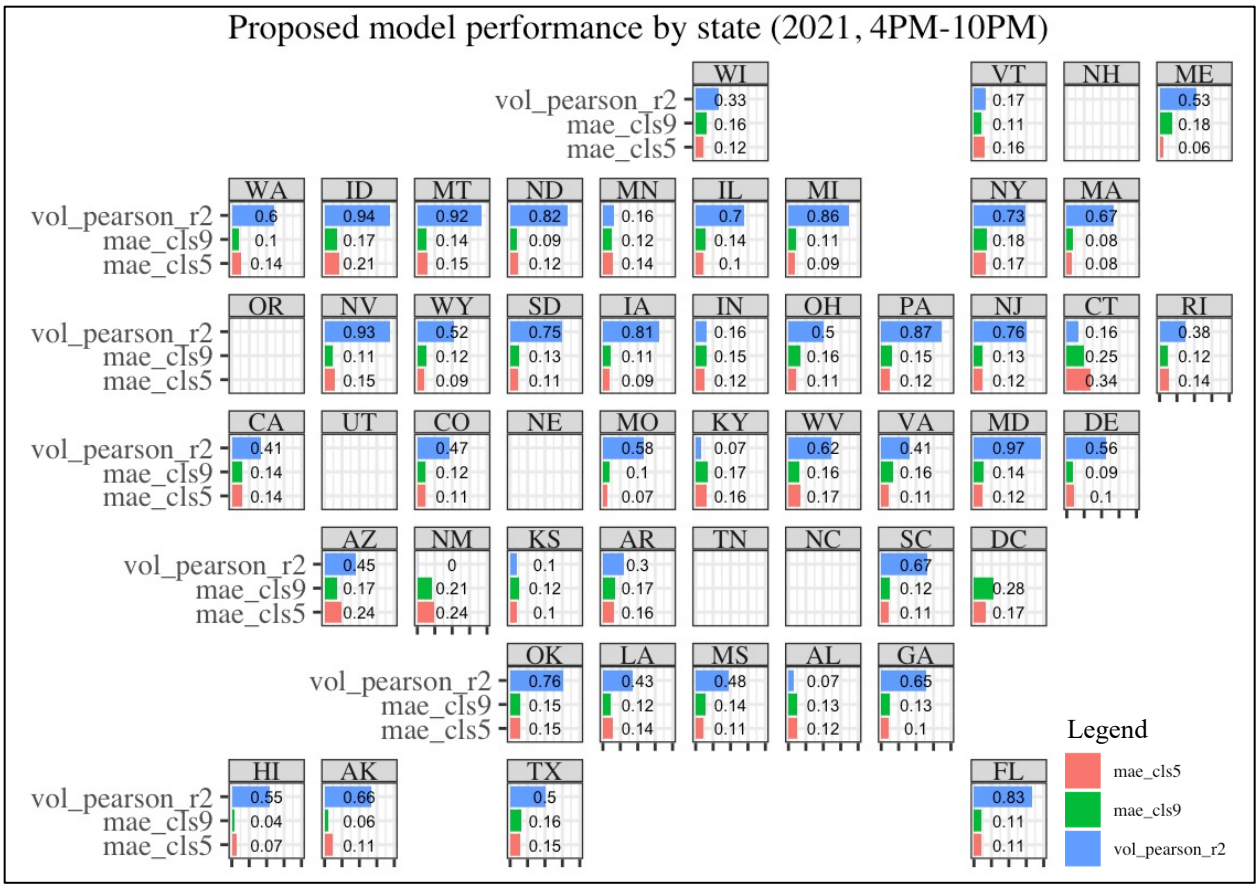


Figure 40. Breakdown analysis for the performance of truck volume prediction for different states.

In addition to the regional analysis, we investigated the performance variation across different times of the day. Table 13 presents a breakdown of performance metrics for truck volume classes across distinct time intervals. Each day was segmented into four periods: T1 (22:00–4:00), T2 (4:00–10:00), T3 (10:00–16:00), and T4 (16:00–22:00). Three metrics were reported: Pearson R^2, MAE, and RMSE. Two other basic benchmark methods, GWR and Kriging, were compared with the proposed method. The results showed that the Kriging method has a poor performance with R^2 ranges from 0.10 to 0.20. GWR and the proposed method had a similar output. For GWR, the performance was good because AADT was well correlated to the imputed traffic volume.

Despite initial concerns that temporal segmentation might adversely affect performance, the Pearson R^2 outcomes exhibited only marginal degradation for the proposed method. Instead of using a statistical regression method, the proposed method gets a similar result solely based on a physical informed algorithm.

Table 14 disaggregates the performance by each state. Five temporal aggregation levels were reported: all time, T1, T2, T3, and T4. The RMSEs for Classes 5 and 9 were examined. Among the distribution of Classes 5–13, the CEL was reported to measure the discrepancies between observed and predicted class distributions.

Model evaluation by visualizing imputation results

Figure 41 illustrates the performance breakdown of traffic volume prediction across individual states, employing cross validation methodology for analysis. The *x*-axis denotes the density of TMAS stations, whereas the *y*-axis represents the Pearson R^2 coefficient. Each data point corresponds to a state's prediction performance, with the size of the marker indicating the number of TMAS stations within the state. Some states are excluded from the graph because they do not have TMAS stations. Additionally, the District of Columbia and Hawaii were excluded from the analysis because they exhibited outlier characteristics owing to unique geometries or a scarcity of detectors. States with higher density seem to have more stable performances.

Figure 42 illustrates a comparative analysis between imputed truck volume and HPMS truck volume. The left subplot displays the forecasted values generated by the proposed algorithm, whereas the right subplot exhibits the HPMS's truck AADT volume. Each blue point on the left subplot corresponds to a TMAS station. As anticipated, the imputation outcomes were similar to those of the historical AADT plot, despite AADT values being solely used as weights for network links within the algorithm. This underscores the efficacy of the algorithm in accurately imputing traffic volume across the network using the TMAS dataset.

Analysis of truck travel pattern along highway: a preliminary case study of Savannah, Georgia

A significant advantage of the proposed method lies in its capacity to extrapolate values across the network, transcending the confines of fixed TMAS stations. Furthermore, in addition to its capability to analyze traffic patterns at national or state levels, the proposed method proves highly valuable for scrutinizing local activities through the strategic placement of detectors along traffic links. This spatial analysis is automatically facilitated by the imputation algorithm, which

Table 13. A comparison of different experiments segmented by time of the day (for weekdays)

Average hourly traffic volume (Classes 5–13)									
Aggregation level	GWR			Kriging			Proposed method		
	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE
All time	0.59	28.48	65.59	0.16	58.73	96.30	0.57	15.67	42.26
T1: (22:00–4:00)	0.49	15.65	42.70	0.13	31.41	57.35	0.52	14.85	39.58
T2: (4:00–10:00)	0.61	31.27	68.80	0.20	61.81	99.85	0.58	32.05	70.93
T3: (10:00–16:00)	0.62	42.10	89.40	0.19	84.93	132.98	0.65	47.35	103.12
T4: (16:00–22:00)	0.59	29.53	65.61	0.19	35.59	58.36	0.52	30.22	71.78
January 2021	0.66	23.62	51.60	0.16	53.53	86.50	0.57	26.18	57.78
February 2021	0.48	27.60	79.23	0.15	57.15	93.78	0.52	28.84	66.17
March 2021	0.54	27.95	73.75	0.17	59.04	98.90	0.55	28.57	67.43
April 2021	0.56	28.71	68.89	0.16	59.95	97.58	0.54	29.63	67.60
May 2021	0.47	29.26	78.41	0.17	56.73	94.34	0.52	28.69	67.62
June 2021	0.48	30.66	81.17	0.18	60.64	97.61	0.51	31.38	70.96
July 2021	0.59	28.65	63.55	0.16	60.92	96.28	0.53	29.36	67.18
August 2021	0.57	29.03	68.83	0.16	61.34	99.51	0.52	30.51	69.96
September 2021	0.71	27.00	52.46	0.18	59.69	93.06	0.58	30.03	62.33
October 2021	0.70	26.57	53.66	0.15	60.92	96.12	0.61	28.58	60.40
November 2021	0.71	26.25	52.20	0.18	59.27	93.98	0.63	28.67	59.12
December 2021	0.67	26.03	54.42	0.20	56.48	89.92	0.63	27.31	57.04

operates seamlessly within the traffic network. To illustrate the efficacy of the tool, a case study was conducted, focusing on Savannah, Georgia. By 2023, the port of Savannah had become the fourth largest port in the United States (Lauriat, 2024), underscoring the importance of assessing truck performance along the major highways in proximity to Savannah.

Figure 43 illustrates the distribution of vehicle classes across varying times of the day and on weekdays versus weekends. Analysis revealed that within the subset of Vehicle Classes 9–13, Class 9 vehicles (i.e., trailer trucks) are the most common vehicles traveling on the eastbound and westbound Interstate 16 highways. In contrast, the use of Interstate 95 by Class 9 vehicles was comparatively lower, with Class 5 (two-axle vehicles, such as delivery trucks) being more prevalent. The figure also depicts the volume of Class 9 vehicles. Integration of this data with speed profiles and emission models facilitated the estimation of emission effects on major roadway segments.

A further visualization of specific stations: analyzing outliers

Assessing performance at the level of individual detectors for comprehensive analysis was imperative. Figure 44 illustrates an interactive map using the Folium Python package, wherein each circle denotes a TMAS station. The size of each circle corresponds to the hourly truck traffic volume at the station, and the angle of each pie chart corresponds to a TMAS station. The angular sectors of the green and red pies represent the shares of Classes 5 and 9, respectively. Smaller angles signify smaller share. Note that a TMAS station could have one or two directions. The figure therefore provides a comprehensive visualization of truck volume and class distributions. The figure also reveals that many TMAS stations were located at connector roads instead of along busy highways.

The map's interactivity enables users to click on circles to access the vehicle class distribution for each station. For example, the bottom row of Figure 44 shows the class distributions of three corresponding TMAS stations. The blue line signifies the observed percentage for each class, whereas the orange line depicts the percentage predicted or imputed by the proposed algorithm. Researchers can use the chart to check the distribution of trucks over places to understand truck traffic patterns over the highway system.

Conclusion

In this study, we introduced a novel iterative method for imputing traffic information across a traffic network. Beginning with a foundational HPMS dataset of AADT values, our approach enabled efficient imputation of observations at fixed locations throughout the traffic network. Unlike conventional interpolation techniques that primarily estimate AADT values, our method extends its capabilities to estimate aggregated traffic information across both temporal and spatial dimensions, accounting for diverse vehicle classes. The method's versatility and underlying analytical framework render it highly applicable in various domains, including traffic analysis, emission modeling, and truck behavior analysis. Notably, our study offers several new perspectives on a national scale, leveraging insights derived from the TMAS dataset, thus advancing the state-of-the-art traffic data analysis based on the TMAS dataset.

Table 14. A summary of cross validation experiment results of the vehicle class shares of each state

State name	Vehicle class probability (proposed method)														
	Class 5 (RMSE)					Class 9 (RMSE)					All classes (CEL)				
	All	T1	T2	T3	T4	All	T1	T2	T3	T4	All	T1	T2	T3	T4
All states	.039	.042	.037	.036	.047	.040	.042	.038	.039	.048	1.442	1.446	1.471	1.444	1.370
Alabama	.119	.026	.020	.017	.024	.137	.032	.020	.020	.026	1.517	1.44	1.470	1.469	1.430
Alaska	.165	.066	.061	.039	.042	.084	.034	.020	.010	.009	1.320	1.575	1.398	1.311	1.212
Arizona	.255	.186	.117	.108	.106	.185	.117	.063	.058	.068	1.323	1.653	1.395	1.370	1.221
Arkansas	.155	.026	.030	.026	.041	.174	.043	.029	.038	.055	1.364	1.433	1.358	1.390	1.355
California	.152	.044	.036	.036	.054	.148	.039	.037	.038	.049	1.201	1.306	1.271	1.205	1.137
Colorado	.123	.022	.027	.020	.030	.137	.029	.029	.029	.032	1.560	1.475	1.572	1.524	1.525
Connecticut	.275	.066	.105	.095	.135	.196	.096	.070	.069	.066	1.816	1.924	1.848	1.799	1.792
Delaware	.180	.034	.028	.058	.054	.092	.032	.017	.013	.022	1.474	1.879	1.471	1.486	1.337
District	.215	.003	.059	.091	.002	.054	.018	.006	.003	.002	1.694	1.963	2.979	1.844	0.872
Florida	.121	.028	.022	.024	.034	.137	.03	.031	.032	.040	1.377	1.342	1.44	1.395	1.266
Georgia	.104	.020	.024	.025	.032	.143	.028	.039	.041	.045	1.366	1.32	1.367	1.343	1.335
Hawaii	.066	.011	.008	.008	.006	.040	.005	.003	.002	.001	0.834	0.877	1.077	0.869	0.505
Idaho	.233	.072	.074	.058	.088	.218	.058	.046	.049	.070	1.843	1.958	1.878	1.823	1.797
Illinois	.118	.017	.020	.020	.033	.159	.032	.033	.036	.050	1.308	1.251	1.316	1.302	1.294
Indiana	.126	.025	.026	.024	.045	.154	.043	.039	.045	.068	1.286	1.217	1.327	1.304	1.253
Iowa	.101	.008	.018	.013	.026	.118	.008	.023	.022	.031	1.239	1.058	1.274	1.241	1.196
Kansas	.104	.009	.018	.017	.026	.110	.010	.021	.019	.024	1.213	1.197	1.259	1.219	1.198
Kentucky	.207	.111	.055	.055	.098	.215	.076	.070	.063	.097	1.469	1.56	1.504	1.502	1.406
Louisiana	.150	.065	.041	.047	.042	.109	.067	.045	.062	.039	1.259	2.126	1.264	1.331	1.238
Maine	.050	.003	.006	.007	.007	.159	.060	.028	.034	.043	1.613	1.45	1.634	1.659	1.544
Maryland	.120	.041	.024	.027	.054	.153	.042	.032	.039	.055	1.424	1.365	1.48	1.485	1.302
Massachusetts	.078	.010	.012	.018	.019	.085	.013	.014	.018	.023	1.540	1.483	1.527	1.522	1.531
Michigan	.097	.012	.013	.010	.023	.133	.020	.022	.024	.036	1.478	1.404	1.543	1.452	1.387
Minnesota	.146	.034	.032	.035	.050	.143	.043	.024	.036	.038	1.589	1.621	1.529	1.565	1.557
Mississippi	.140	.031	.036	.032	.049	.178	.052	.062	.058	.066	1.810	1.599	1.595	1.555	1.581
Missouri	.070	.017	.015	.008	.021	.094	.022	.021	.015	.025	1.215	1.203	1.262	1.237	1.166
Montana	.161	.039	.038	.036	.058	.150	.030	.033	.030	.042	1.658	1.654	1.711	1.666	1.645
Nebraska	Data not available														

Table 14. Continued

State name	Vehicle class probability (proposed method)														
	Class 5 (RMSE)					Class 9 (RMSE)					All classes (CEL)				
	All	T1	T2	T3	T4	All	T1	T2	T3	T4	All	T1	T2	T3	T4
Nevada	.131	.004	.078	.046	.037	.128	.018	.042	.030	.026	1.422	1.263	1.655	1.450	1.306
New Hampshire	Data not available														
New Jersey	.151	.030	.033	.036	.053	.153	.038	.036	.041	.048	1.376	1.413	1.413	1.377	1.228
New Mexico	.321	.145	.155	.136	.166	.265	.121	.121	.106	.142	1.856	1.989	1.872	1.771	1.814
New York	.156	.046	.045	.047	.066	.184	.054	.051	.065	.080	1.640	1.668	1.701	1.686	1.599
North Carolina	Data not available														
North Dakota	.120	.026	.023	.019	.030	.110	.031	.019	.021	.028	1.646	1.759	1.635	1.663	1.631
Ohio	.120	.018	.028	.025	.042	.160	.039	.045	.046	.065	1.425	1.396	1.461	1.402	1.402
Oklahoma	.157	.056	.035	.031	.049	.154	.054	.039	.036	.047	1.304	1.352	1.327	1.351	1.234
Oregon	Data not available														
Pennsylvania	.148	.040	.032	.036	.062	.184	.048	.049	.056	.082	1.465	1.467	1.485	1.491	1.374
Rhode Island	.169	.039	.025	.036	.037	.148	.018	.024	.032	.023	1.246	1.412	1.266	1.251	1.181
South Carolina	.130	.034	.029	.030	.035	.130	.031	.032	.032	.033	1.259	1.238	1.276	1.264	1.157
South Dakota	.108	.008	.021	.019	.022	.143	.010	.029	.023	.026	1.472	1.396	1.531	1.496	1.421
Tennessee	Data not available														
Texas	.146	.044	.042	.040	.049	.158	.049	.046	.042	.055	1.325	1.174	1.365	1.330	1.327
Utah	Data not available														
Vermont	.300	.137	.143	.114	.137	.238	.083	.056	.056	.082	1.763	1.783	1.761	1.672	1.633
Virginia	.118	.021	.024	.029	.036	.170	.035	.042	.048	.051	1.368	1.279	1.382	1.387	1.293
Washington	.148	.056	.049	.049	.060	.110	.031	.018	.023	.029	1.758	1.823	1.827	1.789	1.609
West Virginia	.184	.084	.049	.070	.079	.190	.077	.057	.070	.066	1.437	1.416	1.52	1.504	1.315
Wisconsin	.122	.025	.023	.021	.042	.181	.046	.050	.046	.057	1.490	1.383	1.513	1.474	1.404
Wyoming	.115	.023	.021	.023	.029	.144	.028	.030	.031	.041	1.613	1.681	1.669	1.635	1.586

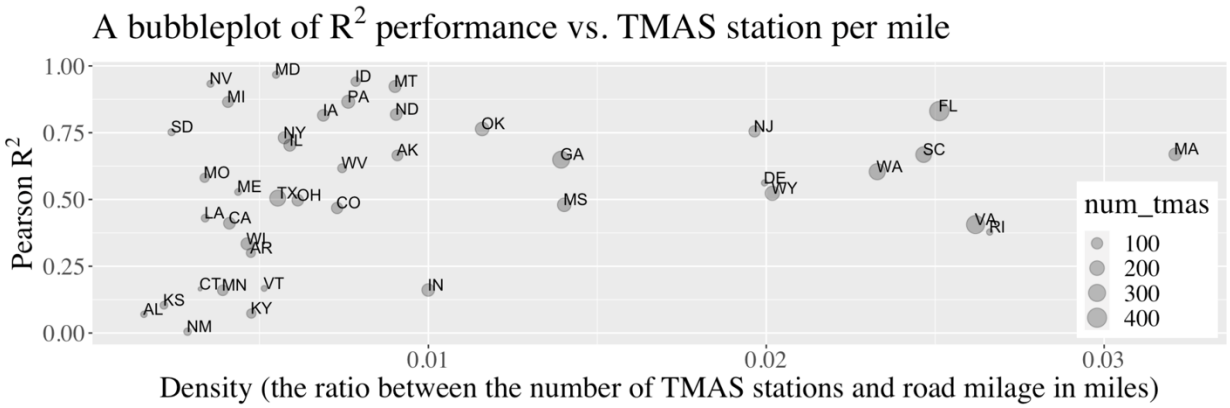


Figure 41. A bubble plot comparing the Pearson R^2 performance for traffic volume with the density of TMAS stations by state.

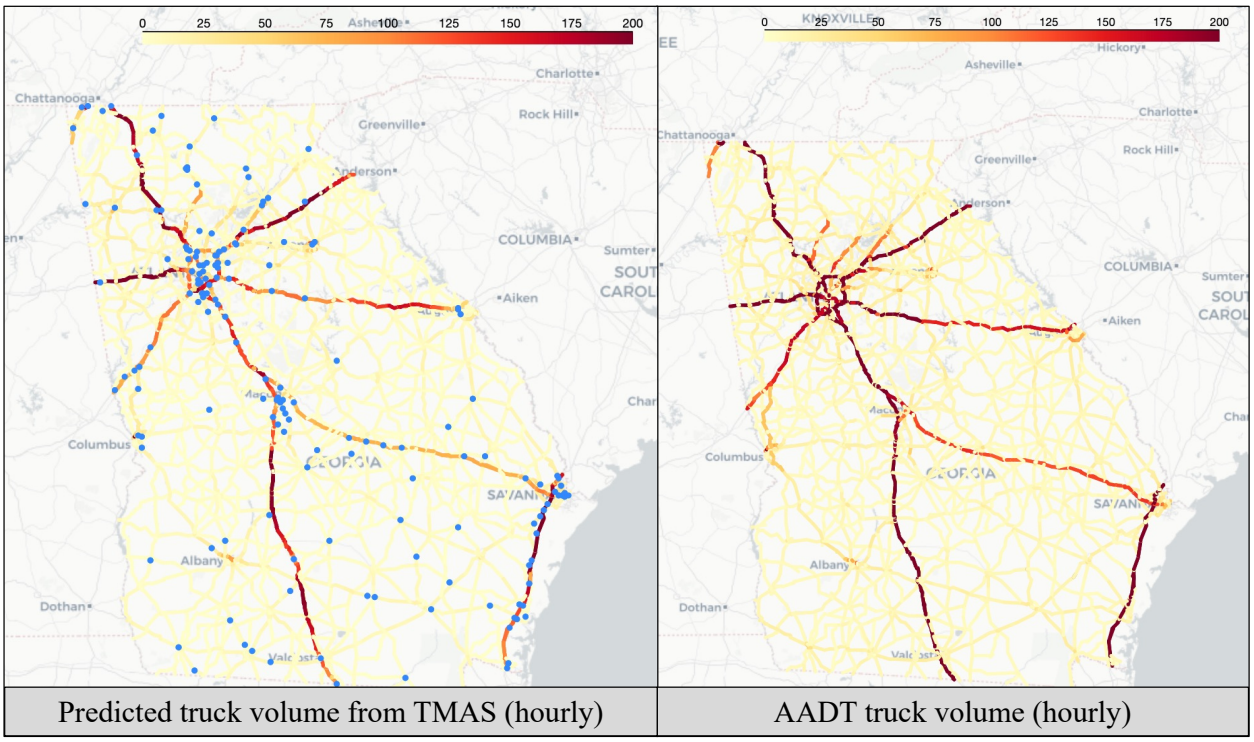


Figure 42. A comparison between imputed truck value per hour and HPMS AADT truck volume (normalized to hourly volume) for Georgia.

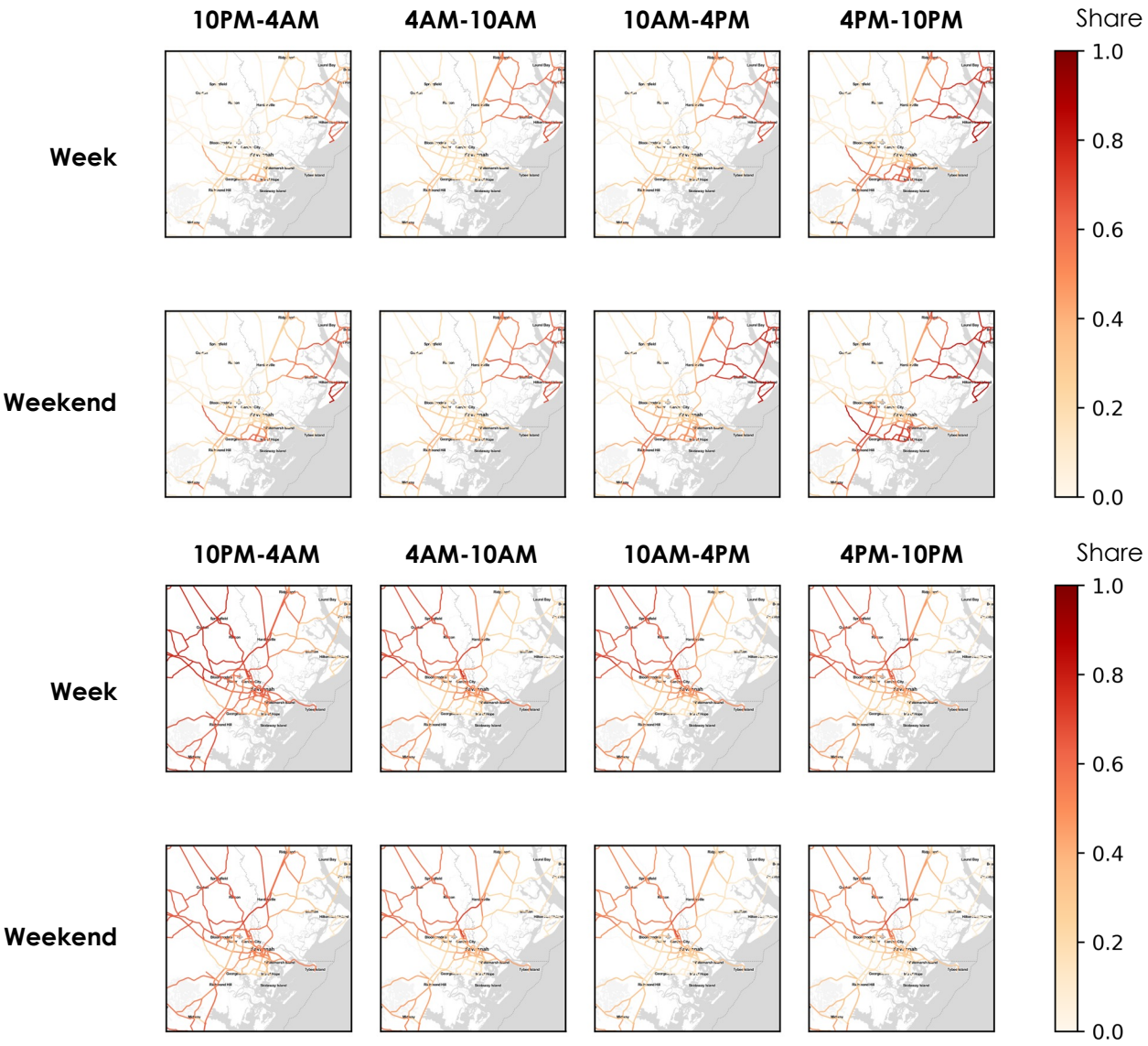


Figure 43. A visualization of the ratio of Classes 5 and 9 for different times of the day for weekdays and weekends in Savannah, Georgia.

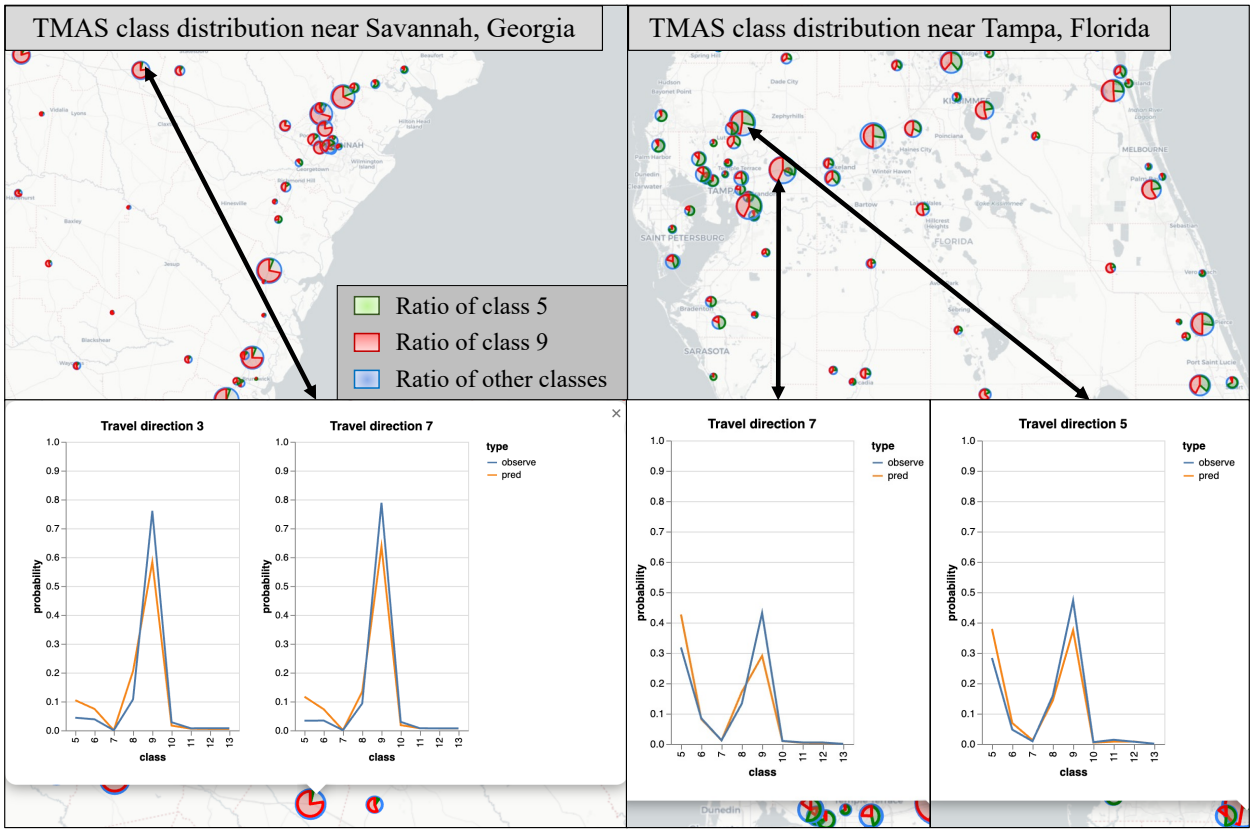


Figure 44. An interactive map of TMAS stations visualizing truck volume and vehicle class distributions near (left) Savannah, Georgia, and (right) near Tampa, Florida.

As a novel approach, the proposed method may have some inherent limitations. Despite enhancements to the quality of the TMAS dataset over the preceding 5 years, significant room for refinement persists. First, the integrity of certain state-contributed data remains suspect, with concerns arising with the potential use of synthesized data for estimating absent observations. Second, the heterogeneity of detector types and data formats across different states presents an obstacle to the conversion of records into the standardized TMAS format. Within the TMAS data schema, ambiguous descriptions remain. Notably, in the context of detector alignment, the dataset exclusively accommodates eight cardinal travel directions (i.e., east, west, north, south, northeast, northwest, southeast, and southwest), complicating station-to-direction alignment procedures. It is plausible, for instance, for a directional designation such as northbound to correspond to eastbound in practice, owing to overarching highway nomenclature, such as *I-40 E*. Furthermore, certain states have not submitted vehicle classification data, although it is likely that many of these states possess the relevant data. The complexity inherent in the TMAS dataset's 13-class "F-scheme" may prove prohibitive, necessitating advocacy for states to furnish a simplified classification system when F-scheme is not possible. Such a simplification could entail the adoption of a truncated F-scheme featuring two to three most dominant vehicle classes, such as Class 5 for delivery trucks, Class 9 for single-trailer trucks, and Class 13 for double-trailer trucks.

Despite data quality considerations, the proposed methodology has several inherent limitations. Each computational step potentially can introduce errors. The estimation of AADT values for individual segments within the FAF network based on the HPMS network may incur inaccuracies because of the intricate geometries of the network. Additionally, in the process of imputation, the determination of the termination criterion necessitates manual intervention to regulate the parameter K, the number of epochs through the graph's edges. Furthermore, the imputation technique relies on the assumption of AADT truck volume, which commonly deviates from reality if investigated for some special time range.

Overall, our study demonstrates the feasibility of estimating freight traffic through the integration of AADT and TMAS observations, employing imputation techniques across the network. The sparse distribution of TMAS stations presents a significant challenge that has not been thoroughly investigated. This concept holds promise for further development and adaptation into alternative methodologies. For instance, there is potential to leverage deep learning techniques such as graph neural networks to facilitate information propagation across the network based on TMAS. Additionally, merging the three TMAS datasets—volume, class, and weight—offers the prospect of establishing a cohesive digital twin for analyzing freight traffic dynamics.

References

2022 Traffic Monitoring Guide, 2022.

AADT/ADT | Federal Highway Administration [WWW Document], 2016. URL https://www.fhwa.dot.gov/policyinformation/travel_monitoring/pubs/aadt/ (accessed 3.4.24).

Anderson, M.D., Sharfi, K., Gholston, S.E., n.d. Direct Demand Forecasting Model for Small Urban Communities Using Multiple Linear Regression.

Bae, B., Kim, H., Lim, H., Liu, Y., Han, L.D., Freeze, P.B., 2018. Missing data imputation for traffic flow speed using spatio-temporal cokriging. Transp. Res. Part C Emerg. Technol. 88, 124–139. https://doi.org/10.1016/j.trc.2018.01.015

Bui, K.-H.N., Cho, J., Yi, H., 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. Appl. Intell. 52, 2763–2774. https://doi.org/10.1007/s10489-021-02587-w

Caceres, N., Romero, L.M., Morales, F.J., Reyes, A., Benitez, F.G., 2018. Estimating traffic volumes on intercity road locations using roadway attributes, socioeconomic features and other work-related activity characteristics. Transportation 45, 1449–1473. https://doi.org/10.1007/s11116-017-9771-5

Chang, H., Cheon, S., 2019. The potential use of big vehicle GPS data for estimations of annual average daily traffic for unmeasured road segments. Transportation 46, 1011–1032. https://doi.org/10.1007/s11116-018-9903-6

Chapter 2. Introduction to Vehicle Classification - Verification, Refinement, and Applicability of Long-Term Pavement Performance Vehicle Classification Rules , November 2014 - FHWA-HRT-13-091 [WWW Document], n.d. URL https://www.fhwa.dot.gov/publications/research/infrastructure/pavements/ltpp/13091/002.cfm (accessed 5.15.24).

Dasgupta, S., Papadimitriou, C.H., 2006. Algorithms.

Gotway, C.A., Ferguson, R.B., Hergert, G.W., Peterson, T.A., 1996. Comparison of Kriging and Inverse-Distance Methods for Mapping Soil Parameters. Soil Sci. Soc. Am. J. 60, 1237–1247. https://doi.org/10.2136/sssaj1996.03615995006000040040x

Gu, Y., Liu, D., Arvin, R., Khattak, A.J., Han, L.D., 2023. Predicting intersection crash frequency using connected vehicle data: A framework for geographical random forest. Accid. Anal. Prev. 179, 106880. https://doi.org/10.1016/j.aap.2022.106880

Hagberg, A., Swart, P.J., Schult, D.A., 2008. Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Hezaveh, A.M., Arvin, R., Cherry, C.R., 2019. A geographically weighted regression to estimate the comprehensive cost of traffic crashes at a zonal level. Accid. Anal. Prev. 131, 15–24. https://doi.org/10.1016/j.aap.2019.05.028

Hwang, H.-L., Hargrove, S., Chin, S.-M., Wilson, D.W., Lim, H., Chen, J., Taylor, R., Peterson, B., Davidson, D., 2016. The Freight Analysis Framework Verson 4 (FAF4) - Building the FAF4 Regional Database: Data Sources and Estimation Methodologies (No. ORNL/TM-2016/489, 1325489). https://doi.org/10.2172/1325489

Hwang, H.-L., Lim, H., Chin, S.-M., Uddin, M., Biehl, A., Xie, F., Hargrove, S., Liu, Y., Wang, R., 2021. Freight Analysis Framework Version 5 (FAF5) Base Year 2017 Data Development Technical Report (No. ORNL/TM-2021/2154, 1844893). https://doi.org/10.2172/1844893

Jiang, W., Luo, J., 2022. Graph neural network for traffic forecasting: A survey. Expert Syst. Appl. 207, 117921. https://doi.org/10.1016/j.eswa.2022.117921
Kong, X., Zhou, W., Shen, G., Zhang, W., Liu, N., Yang, Y., 2023. Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data. Knowl.-Based Syst. 261, 110188. https://doi.org/10.1016/j.knosys.2022.110188
Krile, R.S., Jeremy; Jessberger, Steven, 2016. Assessing Roadway Traffic Count Duration and Frequency Impacts on Annual Average Daily Traffic Estimation: Assessing Accuracy Issues Related to Short-Term Count Durations (No. FHWA-PL-16-016).
Li, M., Zhu, Z., 2021. Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. Proc. AAAI Conf. Artif. Intell. 35, 4189–4196. https://doi.org/10.1609/aaai.v35i5.16542
Liang, Y., Zhao, Z., Sun, L., 2022. Dynamic Spatiotemporal Graph Convolutional Neural Networks for Traffic Data Imputation with Complex Missing Patterns. Transp. Res. Part C Emerg. Technol. 143, 103826. https://doi.org/10.1016/j.trc.2022.103826
Liu, H., Lee, M., Khattak, A.J., 2019. Updating Annual Average Daily Traffic Estimates at Highway-Rail Grade Crossings with Geographically Weighted Poisson Regression. Transp. Res. Rec. J. Transp. Res. Board 2673, 105–117. https://doi.org/10.1177/0361198119844976
Lowry, M., 2014. Spatial interpolation of traffic counts based on origin–destination centrality. J. Transp. Geogr. 36, 98–105. https://doi.org/10.1016/j.jtrangeo.2014.03.007
Mohamad, D., Sinha, K.C., Kuczek, T., Scholer, C.F., 1998. Annual Average Daily Traffic Prediction Model for County Roads. Transp. Res. Rec. J. Transp. Res. Board 1617, 69–77. https://doi.org/10.3141/1617-10
Shen, G., Zhou, W., Zhang, W., Liu, N., Liu, Z., Kong, X., 2023. Bidirectional spatial–temporal traffic data imputation via graph attention recurrent neural network. Neurocomputing 531, 151–162. https://doi.org/10.1016/j.neucom.2023.02.017
Shiledar, A., Sujan, V., Siekmann, A., Yuan, J., 2024. Enhanced Safety of Heavy-Duty Vehicles on Highways through Automatic Speed Enforcement–A Simulation Study. SAE Technical Paper.
Top 20 North American ports [WWW Document], n.d. URL https://www.ajot.com/premium/{path= (accessed 5.16.24).
United States. Department of Transportation. Office of the Secretary of Transportation, United States. Department of Transportation. Bureau of Transportation Statistics. Office of Statistical and Economic Analysis, 2017. Freight Analysis Framework (FAF) FAF5: HiLoForecasts [supporting datasets]. Freight Analysis Framework (FAF). https://doi.org/10.21949/1529116
Wang, X., Kockelman, K.M., 2009. Forecasting Network Data: Spatial Interpolation of Traffic Counts from Texas Data. Transp. Res. Rec. J. Transp. Res. Board 2105, 100–108. https://doi.org/10.3141/2105-13
Weiss, M., Cloos, K.C., Helmers, E., 2020. Energy efficiency trade-offs in small to large electric vehicles. Environ. Sci. Eur. 32, 46. https://doi.org/10.1186/s12302-020-00307-8
Xu, C., Wang, Y., Ding, W., Liu, P., 2020. Modeling the Spatial Effects of Land-Use Patterns on Traffic Safety Using Geographically Weighted Poisson Regression. Netw. Spat. Econ. 20, 1015–1028. https://doi.org/10.1007/s11067-020-09509-2

Yang, H., Yang, J., Han, L.D., Liu, X., Pu, L., Chin, S., Hwang, H., 2018. A Kriging based spatiotemporal approach for traffic volume data imputation. PLOS ONE 13, e0195957. https://doi.org/10.1371/journal.pone.0195957

Zhang, Y., Dong, X., Shang, L., Zhang, D., Wang, D., 2020. A Multi-modal Graph Neural Network Approach to Traffic Risk Forecasting in Smart Urban Sensing, in: 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). Presented at the 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, Como, Italy, pp. 1–9. https://doi.org/10.1109/SECON48991.2020.9158447

Zhao, F., Chung, S., 2001. Contributing Factors of Annual Average Daily Traffic in a Florida County: Exploration with Geographic Information System and Regression Models. Transp. Res. Rec. J. Transp. Res. Board 1769, 113–122. https://doi.org/10.3141/1769-14

Zhao, F., Park, N., 2004. Using Geographically Weighted Regression Models to Estimate Annual Average Daily Traffic. Transp. Res. Rec. J. Transp. Res. Board 1879, 99–107. https://doi.org/10.3141/1879-12

CHAPTER IV. IMPROVING THE ACCURACY OF FREIGHT MODE CHOICE MODELS: A CASE STUDY USING THE 2017 CFS PUF DATA SET AND ENSEMBLE LEARNING TECHNIQUES

A version of the chapter IV was originally published by Diyi Liu, Hyeonsup Lim, Majah Uddin, Yuandong Liu, Lee D. Han, Ho-long Hwang, and Shih-Miao Chin in *Expert Systems with Application, 240, 122478.* Most of the original paper's content is kept in this paper. Besides, the discussion of interpreting the model is slightly modified.

Abstract

The US Census Bureau has collected two rounds of experimental data from the Commodity Flow Survey, providing shipment-level characteristics of nationwide commodity movements, published in 2012 (i.e., Public Use Microdata) and in 2017 (i.e., Public Use File). With this information, data-driven methods have become increasingly valuable for understanding detailed patterns in freight logistics. In this study, we used the 2017 Commodity Flow Survey Public Use File data set to explore building a high-performance freight mode choice model, considering three main improvements: (1) constructing local models for each separate commodity/industry category; (2) extracting useful geographical features, particularly the derived distance of each freight mode between origin/destination zones; and (3) applying additional ensemble learning methods such as stacking or voting to combine results from local and unified models for improved performance. The proposed method achieved over 92% accuracy without incorporating external information, an over 19% increase compared to directly fitting Random Forests models over 10,000 samples. Furthermore, SHAP (Shapely Additive Explanations) values were computed to explain the outputs and major patterns obtained from the proposed model. The model framework could enhance the performance and interpretability of existing freight mode choice models.

Keywords: Commodity Flow Survey, freight mode choice, machine learning, stacking method, interpretable machine learning, behavior analysis

Introduction

Understanding national-level freight mode choice is a vital research area regarding sustainability, planning, policymaking, and economic analysis. The share of freight modes fluctuates over time because of the ever-changing operational characteristics and environment. Therefore, a comprehensive grasp of the underlying determinants that influence mode choice is crucial for optimizing freight transportation systems.

Recently, the landscape of freight logistics has experienced rapid changes under the influence of multiple factors, including technological advancements, infrastructure deterioration, labor shortages, and the fast growth of electronic commerce. For instance, the adoption of autonomous and electric vehicles holds great potential in reducing truck operating expenses (Monios & Bergqvist, 2019), leading to a possible shift from rail to road transport for certain cargo types. The surge in online shopping has placed immense strain on parcel delivery services worldwide (Zenezini et al., 2018), particularly when freight capacity is constrained by factors such as a shortage of truck drivers (Burks & Monaco, 2019; Hobbs, 2020). Furthermore, new operational practices in rail systems such as shift scheduling have been linked to fatigue, which could result in accidents (Rudin-Brown et al., 2019; Zhang et al., 2021). Mitigating these challenges requires a thorough understanding of the flow of goods across various transportation modes.

The Commodity Flow Survey (CFS), a collaborative effort of the US Census Bureau and the Bureau of Transportation Statistics, provides valuable insights into the characteristics of freight shipments in the United States. This shipper-based survey has been conducted every 5 years since the early 1990s and covers a broad spectrum of industries, encompassing approximately 100,000 establishments. The CFS data have been widely used by various federal, state, and local and regional agencies to understand freight commodity flows, evaluate transportation demand and infrastructures, and facilitate informed decision-making processes.

The CFS provides valuable data for analyzing freight transportation mode choices. Furthermore, experimental data products—Public Use Microdata in 2012 and Public Use File (PUF) in 2017—have allowed for more detailed analyses at the shipment level (Arencibia et al., 2015; Larranaga et al., 2017; Samimi et al., 2011). These surveys present a unique opportunity to employ a data-oriented, disaggregate-level approach to better understand freight mode selection.

This study focuses solely on the 2017 CFS PUF data and proposes a framework to enhance model performance by considering three key improvements: (1) constructing local models for each individual commodity/industry category; (2) extracting useful geographical features, particularly the derived distance of each freight mode between origin and destination zones; and (3) implementing additional ensemble learning methods, such as stacking or voting, to combine results from local and unified models for superior performance.

This article is organized as follows. In **Section 2**, prior practices of fitting machine learning models for both general classification tasks and transportation/freight mode choice tasks are discussed. **Section 3** provides a definition of the mode classification used in this study, along with a summary of the CFS PUF data set. **Section 4** describes the data processing steps, the model selection procedures, and the machine learning algorithms adopted. **Section 5** presents model results through both performance metrics and a receiver operating characteristic curve. The model is also explained using different model interpretation methods including variable importance, SHAP values, etc. Finally, **Section 6** concludes with an overview of lessons learned from this study, limitations, and future research directions.

Literature Review

The complexity of the freight system, with its multiple modes of transportation, operators, and routes, makes it difficult to obtain all operational data directly from the field. Consequently, surveys are one of the most viable methods for understanding freight choices for most applications. Two types of surveys are commonly used: stated preference (SP) and revealed preference (RP). SP surveys allow responders to indicate their preferred choice given specific scenarios and choices, whereas RP surveys generate reports that reflect actual choices made for their existing operations. SP surveys are often designed to gain insights into adopting new technologies, infrastructures, and policies. For example, Larranaga et al. (2017) designed an SP survey for six scenarios targeting policies that can encourage intramodality in Rio Grande do Sul, Brazil. RP surveys have become increasingly popular in recent years because of the lowered cost in collecting large data sets. Rich et al. (2009) in Denmark used an RP survey, the FEMEX/COMVIC RP data set, to build a multinomial logistic regression (MNL) model considering the effects of free flow time, congestion time, and waiting time for each traffic mode.

In the United States, numerous SP surveys have been established to understand freight choices within specific regions, industries, and commodities. For example, Samimi et al. (2011) built a behavior model of freight choice based on a survey conducted for 316 firms. Conducting such surveys incurs high labor and monetary costs. The recent establishment of the CFS PUF data set, as an RP data set, has presented an opportunity for researchers to access millions of freight shipment records. This public data set provides a nationwide scope for understanding freight choice in the United States using millions of records revealing freight shipments. Two shipment-level microdata sets were released for 2012 and 2017. The CFS PUF data set contains a wide range of industry and commodity types, as described in **Section 3**. For the 2017 survey, approximately 60,000 establishments were involved, and 5,987,535 shipments were recorded (2017 Commodity Flow Survey (CFS) Public Use File (PUF) Data Users Guide, 2020). With such scope and complexity, data-oriented methods such as machine learning may be preferred over traditional methods to attain high prediction performance in accuracies.

In the field of behavior modeling, two primary approaches are used: regression-based methods and machine learning methods. The regression-based methods, such as MNL, have proven useful in identifying statistical significance surveying transportation mode choice (Arencibia et al., 2015; Holguín-Veras et al., 2021; Keya et al., 2018; Uddin et al., 2021). For models with only two classes, MNL reduces to binary logit regression (LR). Hierarchical classification structures, such as those found in freight modes where intermodal modes like water-truck mode and rail-truck mode fall under intermodal mode, can be modeled with nested logistic regression (NL), which nests multiple MNLs together. These three models (i.e., MNL, LR, and NL) are known as *random utility models* because they assume an individual's preference can be described by a utility function. This approach is commonly employed in traditional freight choice modeling (Arencibia et al., 2015; Holguín-Veras et al., 2021; Larranaga et al., 2017; Moschovou & Giannopoulos, 2012; Norojono & Young, 2003; Rich et al., 2009; Samimi et al., 2011; Shen & Wang, 2012; Uddin et al., 2021; Wang et al., 2013). These models are augmented by various explanatory variables that influence the utilities or attractiveness of each choice. Sensitivity analysis can be used to evaluate the effect of different influencing factors (Arencibia et al., 2015; Larranaga et al., 2017; Samimi et al., 2011). Another method, random regret minimization, employs a similar structure but uses different formulations that focus on the compromise effect of alternatives for modeling freight mode choice using the CFS PUF data set (Keya et al., 2018, 2019). Although these parametric models are valuable for analyses such as sensitivity analysis and econometric analysis, the lack of flexibilities of traditional models often limit their overall performance when applied on millions of data records.

For the classification of CFS PUF, which has thousands of subcategories and millions of records, machine learning methods may be preferred over traditional methods for their superior performance. To evaluate the effectiveness of various machine learning approaches, such as Naïve Bayes (NB), support vector machine (SVM), artificial neural network, *k*-nearest neighbors (KNN), decision tree (DT), random forest (RF), boosting DT, and bagging DT, Uddin et al. (2021) conducted a series of experiments with different numbers of training samples. Although the performance of DT and RF methods reached around 75.4% accuracy, which is significantly higher than utility-based MNL performance of 42.0% accuracy, further improvements can be made using ensemble techniques. Moreover, Uddin et al. (2021) and Keya et al. (2019) augmented the CFS Public Use Microdata data set by adding several secondary information such as local weather, logistics information, and origin and destination facility densities, which could

limit the applicability of these models to public users. Models with accuracies below 80% may still not fully exploit the provided shipment information in terms of predicting freight mode. To further enhance model performance, other machine learning practices must be adopted that utilize big data from other disciplines.

The topic of transportation mode choice has been the focus of recent studies recently. A comprehensive synthesis of these studies during the last decade is presented in Table 15. Notably, with the exception of Uddin et al. (2021), all other studies have primarily focused on individual trip mode rather than freight mode choice, and they are arranged in ascending chronological order of publication. The observations from these studies are as follows: 1) Conventional methods such as MNL/LR do not perform well; 2) Machine learning methods such as NB, SVM, DT, MLP show better performance; 3) Tree-based models like RF and boosting decision trees (BOOST) are likely to achieve higher accuracies, probably due to the tabular nature of mode choice data, as indicated by these comparison studies. However, a limitation inherent in these methodologies is their tendency to prioritize the identification of fundamental input variables, often neglecting the derived explanatory features. Additionally, machine learning offers other advanced techniques to further enhance model's performance.

Over the past two decades, the field of machine learning has seen significant advancements, leading to the emergence of several new techniques that enhance model performance. One noteworthy development in the data science and machine learning community was the Netflix Prize, held from 2006 to 2009 to predict Netflix user ratings on various movies (Bennett & Lanning, 2007). The winning team and other top-performing teams gained valuable insights into developing high-performance machine learning models. According to descriptions provided by the two top teams (Koren, 2009; Töscher et al., 2009), these breakthroughs were mainly attributed to three significant aspects: discovering new features underlying the data; employing different types of base learners; and blending predictions from different algorithms through ensemble methods such as gradient-boosted DT. Following the Netflix Prize, numerous competitions have been conducted over the past decade on platforms such as Kaggle (Bojer & Meldgaard, 2021) for various data sets and tasks. One popular category of ensemble methods is to blend different algorithms together, known as the stacking method (Hastie et al., 2009a), which is frequently employed in diverse classification tasks such as detecting phishing websites (Li et al., 2019), identifying defective products on production lines (Mangal & Kumar, 2016), and predicting antigenic variants of H1N1 influenza (Yin et al., 2018).

Given the constraints of prior freight choice models and the recent advances in developing high-performance models, this study aims to construct a high-performance model using solely the CFS PUF data set without any external data sources. The objective is to build a model framework that achieves superior performance by incorporating the three aspects gleaned from the Netflix Prize teams and other recent machine learning modeling practices into the development process.

In this study, we considered two distinct ensemble learning techniques: the weighted voting method and the stacking method. For feature extraction, we identified routed distances (estimated given the combination of origin, destination, and mode) as a crucial factor. Additionally, other features such as value-to-weight ratio and origin and destination metropolitan

Table 15. Summary of recent comparative study comparing between machine learning algorithms and traditional methods in the application of transportation mode choice.

Paper	Data sources & sizes	Study scope	Mode	Method tested	Training and testing sets	Best performed model reported
Uddin et al. (2021)	2012 CFS data; 4,547,661 shipments from approximately 60,000 responding establishments;	United States	For-hire truck, private truck, parcel service, air, other mode	MNL; NB; SVM; ANN; CART; RF; BOOST; KNN	Performance tested under different sample sizes (136,073-2,267,842); tested training set (10%-60%).	Among all investigated classifiers, the highest model, a RF model, reached about 82% accuracy.
Omrani (2015)	Luxemburg PSELL survey; 9,500 individuals	Luxemburg	Car, PT, walk/bike	MNL; ANN-MLP; ANN-BLF; SVM	60% training set; 40% testing set	Accuracy reported: MNL (0.65); SVM (0.68); ANN-RBF (0.80); ANN-MLP (0.82)
Nam et al. (2017)	Switzerland mode choice survey for long-distance travel; 6,768 individual observations	Switzerland	Car, rail, Metro	NL; CNL; ANN; ANN-MLP; DNN	Best outcomes correspond to 70% training set; 30% testing set	Accuracy reported: NL (0.64); CNL (0.64); ANN-MLP (0.63); DNN (0.67)
Hagenauer & Helbich (2017)	Dutch National Travel Survey (NTS); 230,608 trips reported from 69,918 individuals	Netherlands	car, walk, bike, PT	MNL; NB; SVM; ANN; RF; BOOST; BAG	69,918 individuals with 230,608 trips reported	MNL (0.561), ANN (0.606), NB (0.602), BOOST (0.801), SVM (0.825), BAG (0.906), RF (0.914)
Cheng et al. (2019)	Household survey;	Nanjing, China	Walk, bike, EM, PT, car	MNL; RF; SVM; BOOST	80% training set; 20% testing set	RF and SVM provides the best performance
Zhao et al. (2020)	State Preference Survey at University of Michigan, Ann Arbor campus; 8,141 observations from 1,163 individuals	Ann Arbor, Michigan, United States	car, walk, bike, PT	MNL; NB; NN; CART; SVM; RF; BOOST; BAG	10-fold cross validation for tunning.	MNL(0.647); mixed logit (0.631); RF (0.856)
Kim (2021)	2016 NHTS. 172,889 trips	Seoul, Korea	Car, bike, PT, walk	ANN, RF, XGBoost	75% for training 25% for testing	ANN (0.882); RF (0.909); XGBoost (0.923)
Tamim Kashifi et al. (2022)	Dutch national travel survey (NTS); 230,608 trips reported from 69,918 individuals	Netherlands	car, walk, bike, PT	LR; MLP; DT; RF; LightGBDT	10% testing set & 90% training set; Ten-fold cross validation for tunning;	Reported accuracies: LR (0.58), MLP (0.62), DT (0.80), RF (0.87), LightGBDT (0.89)

Notes for abbreviation and acronyms: LR: Logic Regression; MNL: Multinominal Logistic Regression; NL: Nested Logit Model; CNL: Cross Nested Logit Model; NB: Naïve Bayes; SVM: Support Vector Machine; CART: Classification and Regression Tree; RF: Random Forest; BOOST: Gradient Boosting Decision Tree; BAG: Bagging Decision Tree; NN: Neural Networks; ANN: Artificial Neural Networks; MLP: Multi-layer Perceptron; PT: public transits; EM: e-motorcycle.

type (i.e., combined statistical areas, metropolitan statistical areas, and other regions) were also found useful. To construct base learners of the stacking and voting methods, we developed local models for various Standard Classification of Transported Goods types (SCTG code) and North American Industry Classification System types (NAICS code). Finally, stacking and voting methods were applied to ensemble the base learners from different commodity and industry genres and data scopes. Given these countermeasures, our findings indicate that all three countermeasures were effective in improving model performance. To the best of the authors knowledge, this is the first machine learning model that achieved accuracies above 90% using the CFS Public Use Microdata data set on freight mode choice predictions. This powerful model not only generates the most probable mode of transport but also estimates the probability of selecting each mode. In summary, this predictive model driven by machine learning can assist various stakeholders such as courier agencies, shippers, and transportation agencies in analyzing freight-related issues across different geographic scales (e.g., state-level, national-level) and product types (e.g., commodity type, industry type).

Explorative Data Analysis

Mode Classification and Variable Definitions

To compare the model performance of this study to the results of the previous work, this study used the five categories of freight modes that were adopted by two previous studies (Keya et al., 2021; Uddin et al., 2021). Table 16 describes the five modes used in this study, along with the original 13 types of major freight modes defined by the CFS. In this classification, rail, water, and pipeline modes are combined into a single mode referred to as other modes. Thus, the 13 modes from CFS are compressed to 5 freight modes: for-hire truck, privately owned truck, parcel service, air, and other modes. Privately owned trucks include company-owned trucks used for logistics activities, and for-hire trucks include self-employed drivers transporting goods over long distances. In total, 18,539 (0.31%) out of 5,978,523 records are provided at a more aggregated level (e.g., single mode) and are not matched to a specific mode in Table 16. Therefore, this study excluded such records and analyzed 5,959,984 (99.69%) records of the 2017 CFS PUF data.

Table 17 presents all information considered from the CPS PUF data set. There are three categories of information: basic shipment information, spatial information, and supplementary information. The basic shipment information includes weight (in US pounds), value (in US dollars), commodity type (SCTG), and industry type (NAICS). The spatial information is based on origin, destination, great circle distance, and the routed distance for each specific freight mode. The 2017 CFS PUF data set reports the routed distance estimated for only the chosen mode. Because of this dependency on the chosen mode, directly incorporating the routed distance as model inputs results in a loss of information, which may cause modeling bias and overfitting. To address this, this study utilized derived routed distance estimates by origin and destination for each of the five modes, including both chosen and unchosen modes (as detailed in **Section 4**). This approach allowed for constructing a geographical profile of various freight modes between states or metropolitan regions. The group of variables of all derived distances for all available modes were employed as model inputs. Supplemental information pertains to detailed data regarding shipments of specialized commodities that require specific handling conditions because of their hazardous nature (e.g., chemical compounds) or temperature

sensitivity (e.g., fresh meat). Such information proves valuable in forecasting the appropriate shipping mode for such commodities. Notably, our machine learning model does not rely on external data sets, thereby easing the efforts of data collection for researchers in their analyses. For each piece of information, there exists one or multiple variables to describe it. In the context of machine learning, these input variables are referred to as input features.

Analyzing Freight Mode Choice Patterns

Figure 45 shows the distribution of the five freight modes across multiple metrics for the 2012 and 2017 CFS data sets. The four plots, from left to right, correspond to metrics based on shipment count, total weight (in US tons), total value (in US dollars), and total ton-miles traveled (in US ton-mile). Across metrics, the share of different modes varies significantly. For instance, although most shipments correspond to the parcel service parcel in terms of shipment numbers, parcel service only represents a very small share in terms of the total weight. Besides, a side-by-side comparison between the 2012 and 2017 data sets provides insight into trends observed over 5 years. Between these two years, the shares of other modes remained relatively stable except for an increase in parcel service and a decrease in privately owned trucks. For analysis in this study, only the 2017 data set was used.

Through the use of total values in US dollars as a metric, Figure 46 and Figure 47 illustrate the percentage distribution (in terms of value) of various modes of shipment origins and destinations in different US states. The results demonstrate that different states exhibit varying mode distributions. For instance, privately owned trucks are more prevalent in the Northeast, whereas for-hire trucks dominate many Midwest states. Furthermore, this study also used spatial information inferred from the derived distance between modes with respect to origin and destination states and metropolitan regions (Section 4).

In addition to the spatial distribution of freight modes, the modal share of different product types must also be understood. To this end, we present the results of our distribution analysis in Figure 48, which displays the modal split by each commodity type according to the SCTG code and by each industry type according to the NAICS code. It is found that certain commodities present distinct mode preferences. For instance, alcoholic beverages (SCTG #08) are predominantly transported by privately owned trucks because of their requirement for special permits (Alcohol Transport Permit Requirements by State | Brew Movers, 2022). Meanwhile, parcel services are commonly used for transporting electronic devices (SCTG #35) or precision instruments (SCTG #38), likely because of their small size/weight and urgent delivery requirements.

In summary, this section presents the results of an exploratory data analysis aimed at identifying key factors and trends to guide the modeling step in freight mode selection. First, the analysis reveals that different locations exhibit varying proportions of shipments. Therefore, incorporating both geographical traits and travel distances between origin and destination is crucial in narrowing the feasible freight modes. Second, the SCTG and NAICS codes characterizing commodity types play a significant role in selecting the preferred freight mode. Given the existence of more than 40 categories for SCTG and NAICS types, careful consideration must be given to exploit such categorical heterogeneity across different cases. Finally, the distribution of different freight modes varies significantly under different metrics. Consistent with previous studies (Keya et al., 2018; Rich et al., 2009; Uddin et al., 2021), we employed the number of shipments (i.e., records) for model evaluation.

Table 16. Five aggregated freight mode classifications and more specific CFS PUF data set mode classifications

Mode classification	Code defined by the CFS PUF data set
For-hire truck (Mode 1)	
For-hire truck	04
Privately owned truck (Mode 2)	
Privately owned truck	05
Parcel service (Mode 3)	
Parcel service (parcel, USPS, or courier)	14
Air (Mode 4)	
Air (including truck and air)	11
Other modes (Mode 5)	
Rail	06
Water	07
Inland water (water)	08
Great lakes (water)	09
Deep sea (water)	10
Multiple waterways (water)	101
Pipeline	12
Truck and rail (multiple modes)	15
Truck and water (multiple modes)	16
Rail and water (multiple modes)	17

Table 17. Description of all considered information from the CFS PUF data set

Column name	Description
Basic shipment information	
Mode (target variable)	Shipment mode (for-hire truck, privately owned truck, parcel, air, other modes)
Weight	Weight of the shipment (in US pounds)
Value	Value of the shipment (in US dollars)
SCTG	Two-digit SCTG commodity code of the shipment (43 categories)
NAICS	Industry classification of the shipper (45 categories)
Spatial information	
Origin	Origin code (state code and metropolitan area code)
Destination	Destination code (state code and metropolitan area code)
Distance	Great circle distance between origin and destination (in miles)
Routed distance	For the chosen mode, the actual routed traveling distance of the shipment (in miles)
Supplemental information	
HAZMAT	Hazardous material (Class 3.0 hazard, other hazard, or not hazard)
Temperature controlled	Temperature-controlled shipment

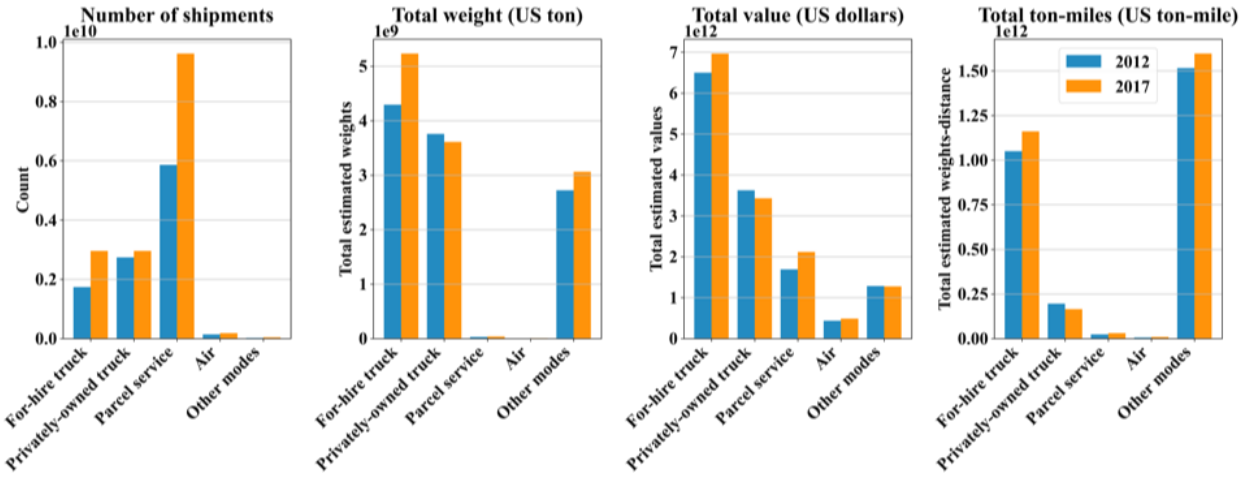


Figure 45. Share of different modes with respect to the number of shipments, total weight, total value, and total ton-miles traveled

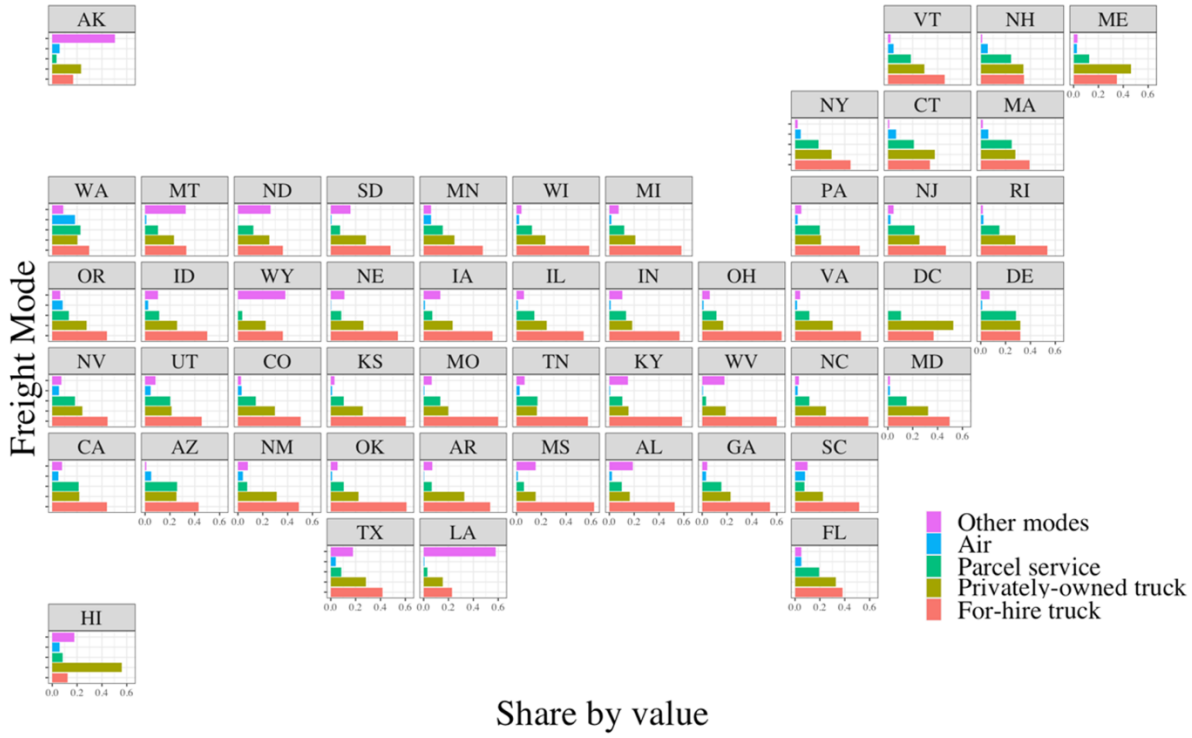


Figure 46. Percentage value share (in US dollars) of different freight modes by origin state

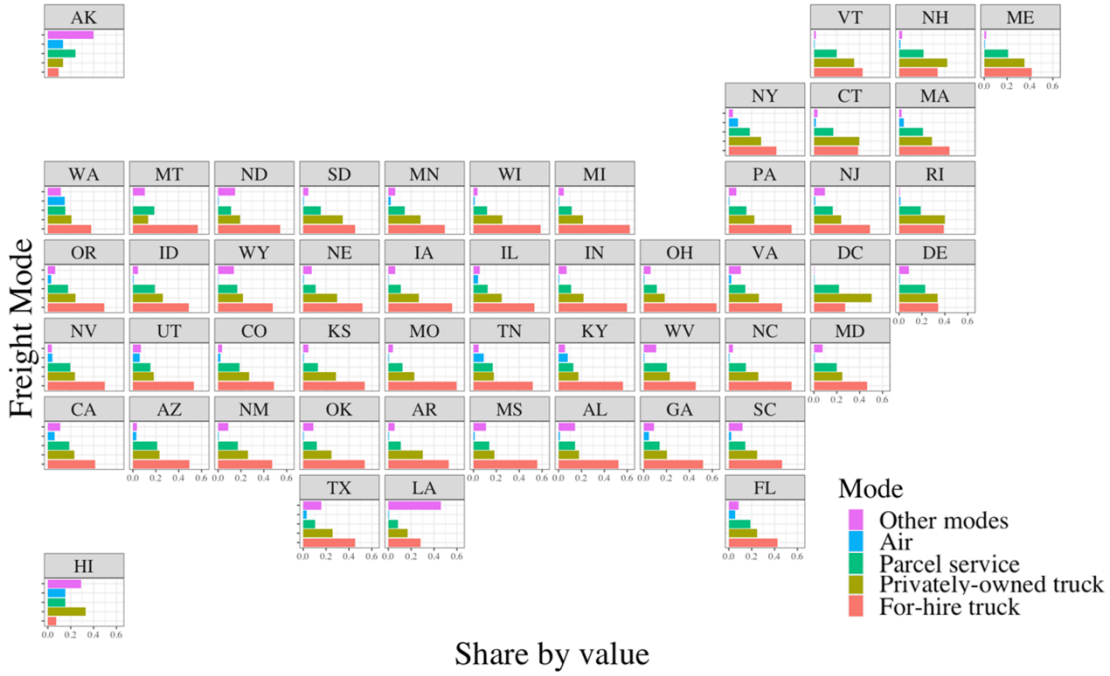


Figure 47. Percentage value share (in US dollars) of different freight modes by destination state

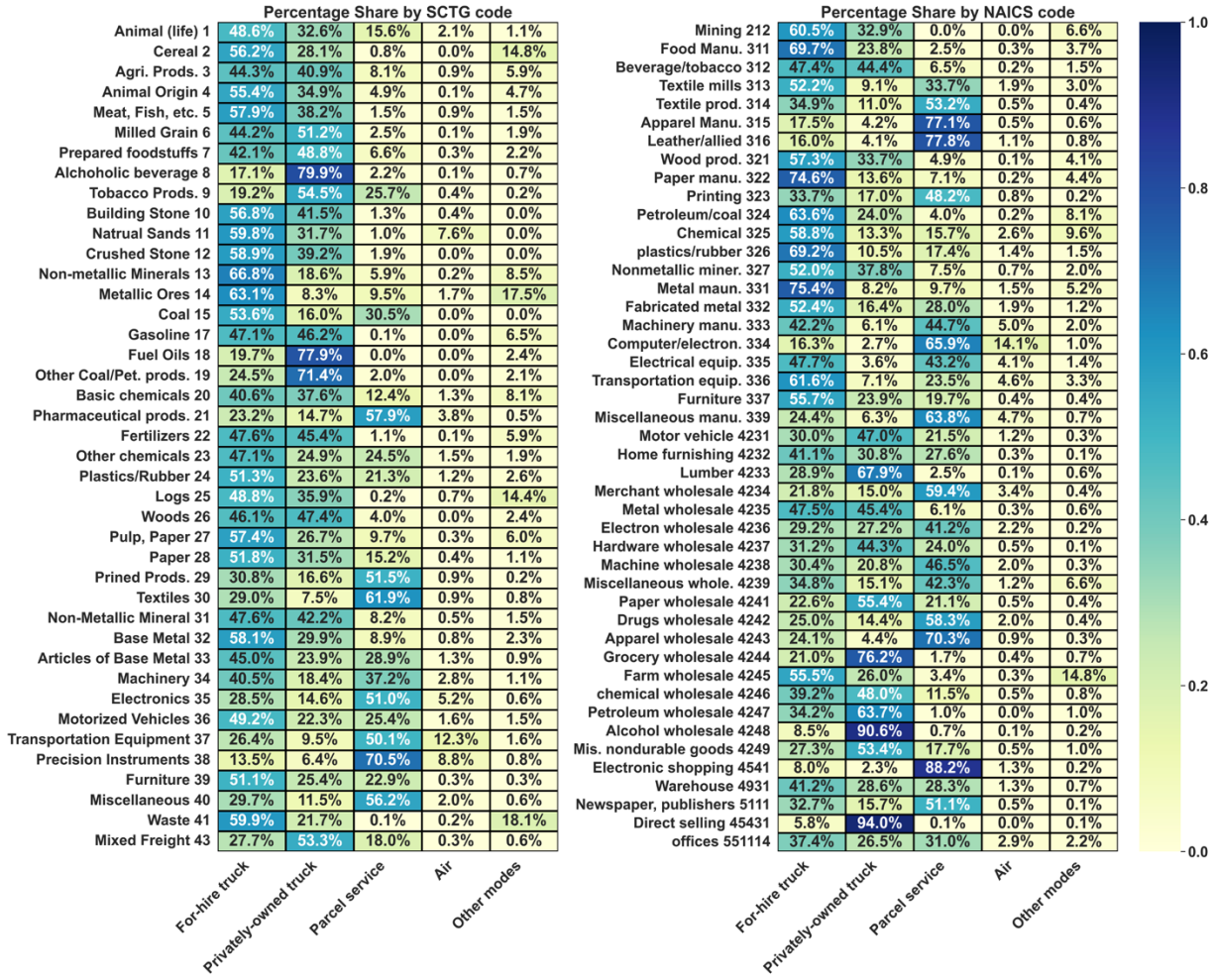


Figure 48. Distribution of different freight mode choices by different SCTG (commodity) codes and NAICS (industry) codes (measured in the number of shipments)

Methodology

The structure of the methodology subsections is delineated as follows: (1) **Section 4.1** outlines the general data processing pipeline employed for constructing the model; (2) **Section 4.2** develops the augmented data features used for machine learning models; (3) **Section 4.3** shows different types of level-1 model as well as the benefits of developing local/separate models to exploit the heterogeneous nature of different commodities/industries types of goods; (4) **Section 4.4** shows the level-2 model blending different genres of level-1 model via model stacking/voting techniques; Furthermore, more detailed steps for training level-2 stacking methods are provided in this section.

An Overview of the Proposed Methodology

To develop a high-performance predictive model, it is essential to construct an efficient data pipeline through an analysis framework. This study focuses on building pipelines to prevent overfitting, data leakage issue, and to compare against different machine learning models and

layouts. Figure 49 illustrates the data processing pipeline and analysis framework employed in this study. Step 1 in the pipeline is to split the data into a training set and a testing set. The testing set comprises 20% of the data, using stratified random sampling, grouped by SCTG code and NAICS code. In Step 2, useful geographical features are extracted, including great circle distance reported from the 2017 CFS PUF data and the derived distances between each origin and destination CFS metropolitan area (i.e., CFA area) for each of the five modes of transportation. At this point, the training and testing data sets are cleaned and formed. In Step 3, multiple machine learning models are trained and tested using different input features. Global base learners are trained using randomly drawn samples from all training data sets while separate models (base learners) for each commodity and industry type are trained and concatenated as one predicting model (i.e., base learner). In Step 4, different types of base learners are combined using either voting averages or stacking methods to create a final estimator (i.e., level-2 model or meta-learner). Each step of the data pipeline is discussed in further detail in the following sections.

Developing Useful Features (Data Augmentation)

Routed distance is the actual distance traveled by a specific freight mode. To obtain useful geographical and distance information (i.e., step 2 of the framework), regional definitions must be well understood. The CFS PUF data set classifies the United States into more than 123 regions, including combined statistical areas, metropolitan statistical areas, and the remainder of each state (2017 Commodity Flow Survey (CFS) Public Use File (PUF) Data Users Guide, 2020). These regions divide the United States into distinct parts (i.e., CFS areas). Since the CFS records provide the estimated routed distances only for the selected mode, directly using the derived distance information in the model may lead to a loss of information. However, since the data set has millions of records, between a specific pair of origins and destinations, it is plausible to assume that there would be many records for each freight mode. Thus, all five modes' routed distances can be used jointly as input features in the model. For example, as input features, given a specific pair of origin and destination, these derived distances by each mode can be computed as a statistic (e.g., median) of observed routed distances.

In Figure 50, the map displayed on the left illustrates the identified CFS regions and their corresponding codes. Each yellow region represents a CFS area, and the remaining regions of each state are depicted in white and have a different code. The routed distance for various freight modes can vary significantly between an origin–destination pair. For instance, Figure 50 shows that the routed distance by air between Los Angeles, California, and New York, New York, is shorter than that of for-hire trucks, which is still shorter than that of other modes (e.g., water, rail, pipes). Comparing the routed distances of various freight modes (top right of Figure 50) provides valuable information regarding mode choice. Notably, using routed/derived distance entails a potential risk of overfitting due to data leakage. In this case, the data leakages refer to the misbehavior of disclosing the predicting mode by providing the routed distance for the selected freight mode. Consequently, all routed distances must be generated as input features from the training data set. Moreover, in this study, only the median distances for each mode between origin–destination pairs are employed to describe a general derived distance given the candidate mode, origin, and destination. Also, it is likely that some modes are not available between some origins and destinations. For example, although privately owned trucks mode is theoretically possible between Los Angeles and New York, no such records are found in the dataset because the trip distance is too long (the privately owned trucks rarely ship beyond 500 miles). For such cases, we impute that derived distances of the missing modes by fitting a regression model to estimate the routed distance given the great circle distance.

All explanatory input variables adopted in the model are described in Table 18, which were further derived from the data set inputs presented in Table 17. Compared with Table 17, the derived variables (marked by the asterisks in Table 18) are explained as follows. The value to weight variable, by measuring the products value given weights, is a useful derived feature. Origin and destination types are referred to by CFS area classes. Three classes are available: combined statistical area, metropolitan statistical area, and remaining part of each state. Although listed in Table 17, origin and destination CFS area is not directly fed into any predicting model. The origins and destinations are applied in an indirect way by aggregating derived distances between origin–destination pairs. Specifically, “M1”–“M5” correspond to the routed distance from each of the five modes. Missing values in routed distance for a specific mode (e.g., “M5”) between an origin–destination pair can arise from limited coverage of the data records, and it is imperative to impute these values for model execution. To this end, a linear regression model was constructed between the great circle distance and derived distance to impute those missing derived distances. Since imputation of missing values may affect model performance, we also propose a set of features to describe whether each derived distance has been imputed or not using Boolean values (true or false), denoted as “I1”–“I5” for five specific modes.

Building Base-learner (Level-1 Model)

In addition to expanding the dataset by deriving more features from existing information, performance enhancement can also be achieved by constructing distinct sub-models tailored to specific data categories. Observed from Figure 48, different commodity or industry types exhibit different preferences and patterns in freight modes. Moreover, dedicated models are frequently fitted to model a specific type of commodity (e.g., hazardous commodity) in some previous studies (Nowlan et al., 2021; Xie et al., 2012). Those practical instances imply that pursuing a single for all product categories may not lead to an optimal performance. Given the limited

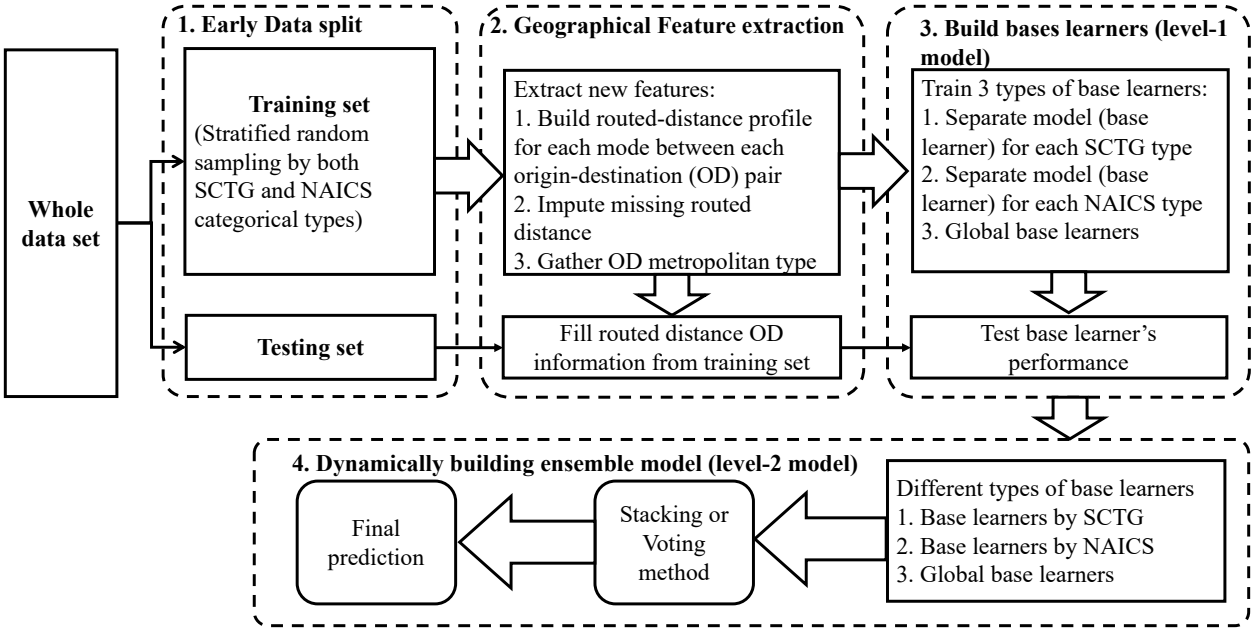


Figure 49. Proposed data processing pipelines and analysis framework

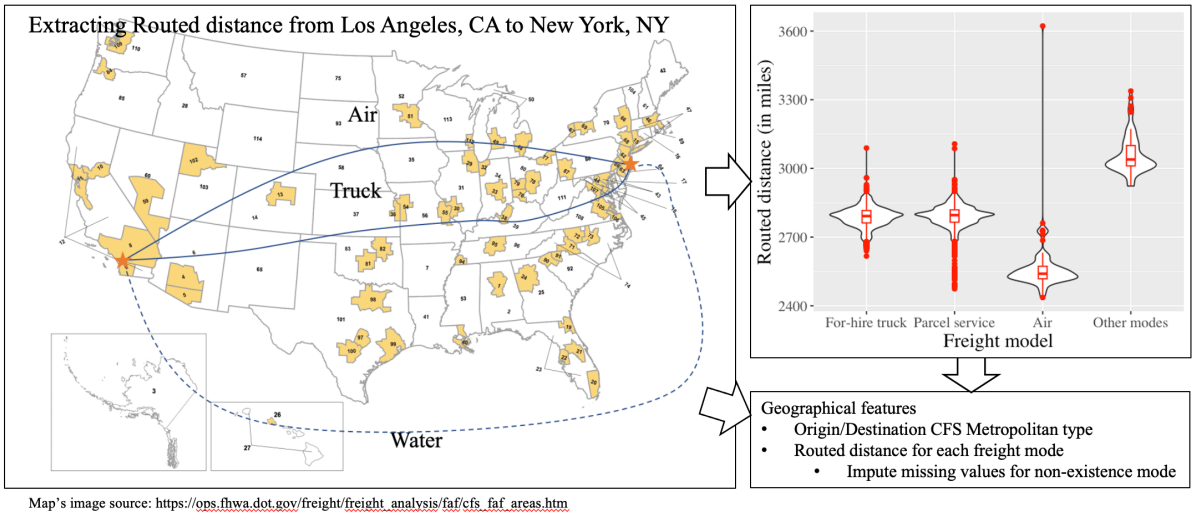


Figure 50. Collected derived distance and spatial information for each mode between each origin–destination pair of CFS areas

Table 18. All explanatory input variables and features employed in the model

Variable	Feature	Description
Basic shipment information		
Single weight	"sw"	Weight of the shipment (in US pounds)
Single Value	"sv"	Value of the shipment (in US dollars)
SCTG	"sctg"	Two-digit SCTG commodity code of the shipment (43 categories)
"sctg_n"	"sctg_n"	Nine types of aggregated SCTG commodity codes
NAICS	"naics"	Industry classification of the shipper (45 categories)
*Value to weight ratio	"v2w"	Value to weight ratio for the shipment (US dollars per pound)
Spatial information		
Origin	"orig"	Origin CFS area code
*Origin area type	"orig_type"	Three classes of CFS areas (C: combined statistical area; M: metropolitan statistical area; R: remaining part of each state)
Destination	"dest"	Destination CFS area code
*Destination area type	"dest_type"	Three classes of CFS areas (C: combined statistical area; M: metropolitan statistical area; R: remaining part of each state)
Great Circle Distance	"dc_dist"	Great circle distance between origin and destination (in miles)
*Routed Distance	"M1," "M2," "M3," "M4," "M5"	For each freight mode, the median routed traveling distance between origin and destination (in miles)
*Imputed Distance	"I1," "I2," "I3," "I4," "I5"	For each freight mode, whether the routed traveling distance was imputed since no records are recorded between origin and destination (in miles)
Supplemental information		
HAZMAT	"haz"	Hazardous material (Class 3.0 hazard, other hazard, not hazard)
Temperature controlled	"temp_cntl"	Temperature-controlled shipment

*Derived information

number of 43 SCTG types and 45 NAICS types, it is entirely feasible to build a dedicated model by each SCTG and each NAICS category. Even for the category with fewest records, there are still enough number of records (e.g., over 2,000) to fit base learners.

In conclusion, three distinct modeling approaches can be considered: (1) a unified base learner applied on all record types; (2) a dedicated model for each SCTG type; (3) a dedicated model for each NAICS type. Fitting dedicated models for different types are not equivalent to the act of adding NAICS/SCTG features (e.g., one-hot encoding variables for SCTG type or NAICS type) for a unified model. Our base learners are trained utilizing a diverse set of machine learning algorithms, encompassing LR, DT, SVM, MLP, KNN, NB, RF, Gradient Boosted Decision Tree (GB), AdaBoost (ADA), and Extra Trees (Extra). Importantly, for any machine learning models, all three types of models can be built as base learners. For any trip to predict, one can dynamically get results from three types of models given its SCTG/NAICS type. Although any available machine learning models can be used as base-learners, only those best performed models are suitable as candidates for level-2 model, as elaborated in the following section.

Dynamically Building Ensemble Models (Level-2 model)

Voting mechanisms are commonly employed ensemble techniques for generating a final prediction by aggregating the outputs of multiple models (Hastie et al., 2009b). Moreover, a diverse range of local models must be considered that specialize in modeling different commodities and industry types. To consider such models, we propose a dynamic voting scheme, as illustrated in Figure 53. Specifically, all relevant base learners/models are selected for each record based on their commodity and industry types. These local models, together with a global base learner, constitute a pool of models. Each model generates five probabilities for selecting five freight modes. A voting classifier then computes the average probability of selecting each mode across all models. The predicted freight mode with the highest probability is considered the final outcome.

Figure 51 illustrates the stacking methodology employed in this study. Analogous to the voting approach, high-performing tree-based models are employed as level-1 base learners. To obtain predictions for the level-2 model, an XGBoost model is trained on the probability predictions and a subset of input features from the level-1 models (Chen & Guestrin, 2016). Compared to other models like RF, XGBoost model has the benefits of fitting on large scale data more efficiently. In contrast, XGBoost model is not used for level-1 model in this study because the model takes too much memory space compared to other machine learning models. The purpose of the level-2 model is to determine optimal weights for the different combinations of NAICS and SCTG types, utilizing a DT model to achieve this objective. Finally, five probabilities are generated as predictions corresponds to each of the five different freight modes. Although the final mode prediction is determined by selecting the prediction with the highest probability, these five probabilities can also provide useful information since products are often distributed using more than one freight modes.

The precise steps to generate the training and testing data sets considering 2-level stacking methods are summarized in Figure 52. As previously mentioned, the first data preparation step is to do a stratified sampling approach to reserve 20% of the data as the testing data set. To prevent data leakage of derived features, the derived features for the testing set were only derived from

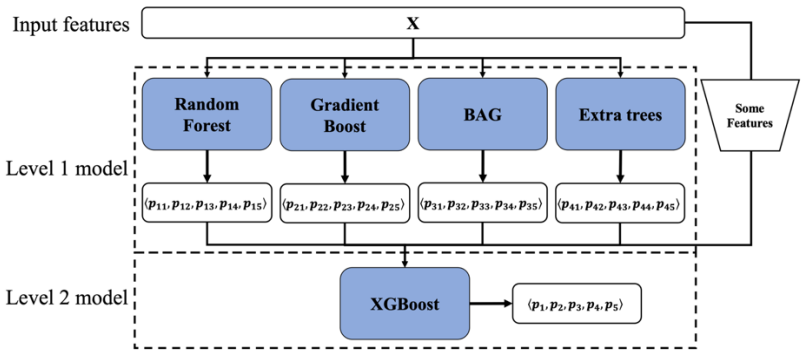


Figure 51. The basic structure of stacking different types of tree-based methods using a two-level hierarchy to obtain final ensembled predictions

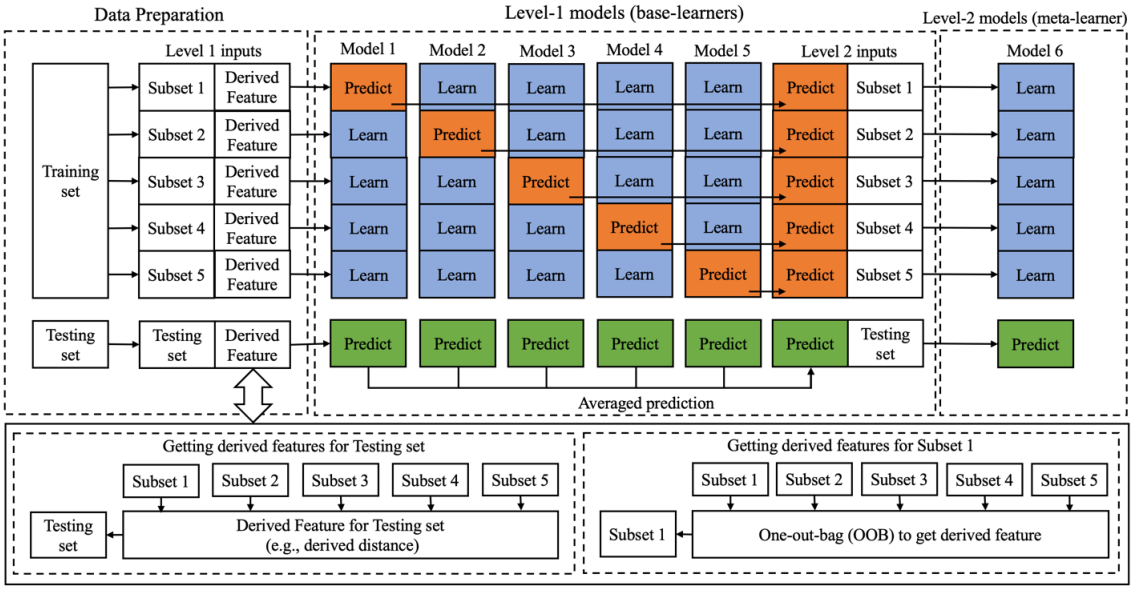


Figure 52. Data flow of the analysis framework: data preparation and cross-validation for the two-level stacking method

Algorithm 1. Dynamically combining different base learners using a voting mechanism

```
procedure Collect_Models_For_One_Record(record, model_types)
    S_M1 = Get_Base_Learners_By_SCTG(record.sctg_id, model_types);
    S_M2 = Get_Base_Learners_By_NAICS(record.naics_id, model_types);
    S_M3 = Get_Global_Learners(model_types);
    S_pool = {s: s ∈ S_M1 ∪ S_M2 ∪ S_M3};
    V_pool = Compute_Predict_Vectors_For_Each_Learner(S_pool);
    V_final = Get_Average_Voting_By_Probabilities(V_pool);
```

Figure 53. One possible pseudocode of steps applying a voting mechanism to obtain ensembled predictions of base learners for each record.

the training set instead of testing set (Figure 52 (bottom left)). Furthermore, to avoid data leakage during the data preparation step before training, a one out-of-bag method was used to generate derived features (Figure 52 (bottom right)). This method ensures that no data records used routed distances from their own group to obtain derived distance.

Notably, derived features only needed to be computed once during the entire data pipeline. When using the stacking method, a level-2 meta-learner is required (Figure 52 (top right)). The inputs for the meta-learner were generated from the validation set of each *k*-fold model to prevent overfitting (orange blocks in Figure 52). During level-1 model training, a five-fold cross-validation approach was applied to determine the best-performing base learners with respect to their meta-parameters. For level-2 models, all training data sets were used to train a meta-learner and were then tested using the reserved testing data set. This means that final model delivered was tested on the testing data set.

Experiment Results

Model Implementation

To assess the efficacy of various methodologies, an extensive range of machine learning models were employed in this study. Similar to some of the previous studies in Table 15, the models used encompassed Logistics Regression (LR), DT, SVM, MLP, KNN, NB, RF, GB, ADA, and extra trees (Extra). In most cases, the Scikit-learn package in Python was used to execute the models (Pedregosa et al., 2011). For the stacking model, the XGBoost package in Python was preferred for analysis because of its greater computational efficiency compared with the Scikit-learn package (Chen & Guestrin, 2016).

Experiment Design and Model Evaluation

In addition to the model performance, the input features and data fitting scope of the model were also evaluated. To enhance the model efficacy, this study proposes three countermeasures: (1) use derived distance as input features; (2) build dedicated models for each SCTG/NAICS type; (3) use ensemble methods to combine different models. The experimental controls for each proposed countermeasure are detailed in Table 19. More specifically, to compare the effectiveness of each of the three countermeasures, seven scenarios were developed and are presented in Table 20. The training set for all data sets comprises 80% of the total samples, and the remaining 20% of samples are reserved as testing set. Testing samples were selected randomly within each NAICS and SCTG category.

Table 19. Experiment design aspects to measure the effectiveness of three countermeasures

Input features	**Description**
"without derived distance"	"M1"–"M5" and "I1"–"I5" are not considered in the model
"with derived distance"	"M1"–"M5" and "I1"–"I5" are considered in the model
Modeling scope	
"unified model"	Only one model is built across the entire data set
"local (concatenated) model"	One separate model is built for each NAICS and SCTG category
Stacking method	
"voting model"	The average of output probabilities of other models is used as the final prediction
"stacking model"	An ensemble structure is applied (**Figure 51**) based on other models

Table 20. Performance of unified models, SCTG models, NAICS models, and the overall voting model on the testing data set

(a) Unified models (without derived distance, trained on 10,000 training records)

	LR	DT	SVM	MLP	KNN	NB	RF	GB	BAG	ADA	Extra	XGBoost	Vote	Stack
Accuracy	0.559	0.653	0.168	0.709	0.630	0.439	0.729	0.670	0.710	0.610	0.668	0.713	**0.731**	0.726
BA	0.395	0.579	0.283	0.583	0.490	0.238	0.623	0.591	0.612	0.508	0.444	0.600	0.617	**0.627**
Precision	0.553	0.653	0.437	0.717	0.628	0.468	0.731	0.676	0.708	0.622	0.686	0.713	**0.733**	0.726
Recall	0.559	0.653	0.168	0.709	0.630	0.439	0.729	0.670	0.710	0.610	0.668	0.713	**0.731**	0.726
F1	0.541	0.653	0.181	0.701	0.627	0.356	0.725	0.664	0.708	0.598	0.659	0.709	**0.728**	0.720
$\hat{\sigma}$(Accuracy)	0.000	0.070	0.046	0.052	0.047	0.060	0.000	0.004	0.009	0.027	0.026	0.001	0.000	0.000

(b) Unified models (with derived distance, trained on 10,000 training records)

	LR	DT	SVM	MLP	KNN	NB	RF	GB	BAG	ADA	Extra	XGBoost	Vote	Stack
Accuracy	0.563	0.779	0.374	0.787	0.681	0.511	0.819	0.716	0.824	0.595	0.796	0.821	0.837	**0.841**
BA	0.469	0.696	0.210	0.709	0.555	0.333	0.699	0.561	0.709	0.491	0.671	0.724	0.715	**0.752**
Precision	0.560	0.779	0.366	0.789	0.682	0.510	0.818	0.724	0.822	0.615	0.797	0.821	0.837	**0.841**
Recall	0.563	0.779	0.374	0.787	0.681	0.511	0.819	0.716	0.824	0.595	0.796	0.822	0.837	**0.840**
F1	0.547	0.779	0.303	0.784	0.680	0.499	0.817	0.708	0.821	0.580	0.794	0.820	0.834	**0.839**
$\hat{\sigma}$(Accuracy)	0.000	0.095	0.074	0.089	0.078	0.074	0.000	0.058	0.055	0.103	0.095	0.000	0.000	0.000

(c) Concatenate local models for each NAICS code (without derived distance, trained on all training data)

	LR	DT	SVM	MLP	KNN	NB	RF	GB	BAG	ADA	Extra	XGBoost	Vote	Stack
Accuracy	0.676	0.752	0.351	0.759	0.774	0.602	0.797	0.759	0.797	0.715	0.756	0.786	**0.806**	0.785
BA	0.499	0.715	0.345	0.649	0.716	0.398	0.723	0.650	0.739	0.529	0.624	0.681	**0.735**	0.722
Precision	0.671	0.752	0.507	0.759	0.773	0.611	0.796	0.758	0.796	0.711	0.756	0.787	**0.806**	0.786
Recall	0.676	0.752	0.351	0.759	0.774	0.602	0.797	0.759	0.797	0.715	0.756	0.786	**0.806**	0.785
F1	0.670	0.752	0.387	0.755	0.773	0.594	0.796	0.756	0.796	0.709	0.753	0.783	**0.805**	0.783
$\hat{\sigma}$(Accuracy)	0.001	0.070	0.164	0.001	0.008	0.077	0.000	0.019	0.018	0.034	0.030	0.000	0.000	0.000

(d) Concatenate local models for each NAICS code (with derived distance, trained on all training data)

	LR	DT	SVM	MLP	KNN	NB	RF	GB	BAG	ADA	Extra	XGBoost	Vote	Stack
Accuracy	0.681	0.884	0.441	0.843	0.873	0.636	0.916	0.812	0.913	0.712	0.902	0.868	**0.922**	0.919
BA	0.552	0.834	0.325	0.752	0.818	0.482	0.849	0.696	0.848	0.530	0.837	0.775	**0.859**	**0.859**
Precision	0.679	0.884	0.476	0.842	0.873	0.634	0.916	0.812	0.913	0.710	0.903	0.870	**0.923**	0.920
Recall	0.681	0.884	0.441	0.843	0.873	0.636	0.916	0.812	0.913	0.712	0.902	0.868	**0.922**	0.919
F1	0.678	0.884	0.447	0.841	0.873	0.632	0.915	0.809	0.913	0.706	0.902	0.866	**0.922**	0.918
$\hat{\sigma}$(Accuracy)	0.000	0.101	0.073	0.000	0.015	0.105	0.000	0.052	0.048	0.084	0.080	0.001	0.000	0.000

(e) Concatenate local models for each SCTG code (without derived distance, trained on all training data)

	LR	DT	SVM	MLP	KNN	NB	RF	GB	BAG	ADA	Extra	XGBoost	Vote	Stack
Accuracy	0.661	0.724	0.280	0.742	0.755	0.554	**0.773**	0.750	0.769	0.706	0.724	0.766	0.759	0.751
BA	0.505	0.689	0.294	0.639	0.685	0.363	**0.700**	0.641	0.713	0.545	0.582	0.659	0.488	0.486
Precision	0.658	0.724	0.415	0.742	0.755	0.572	**0.773**	0.750	0.768	0.704	0.729	0.768	0.751	0.746
Recall	0.661	0.724	0.280	0.742	0.755	0.554	**0.773**	0.750	0.769	0.706	0.724	0.766	0.759	0.751
F1	0.655	0.724	0.306	0.739	0.754	0.545	**0.771**	0.746	0.768	0.701	0.72	0.762	0.754	0.744
$\hat{\sigma}$(Accuracy)	0.000	0.081	0.190	0.000	0.006	0.092	0.000	0.012	0.010	0.027	0.026	0.001	0.000	0.000

(f) Concatenate local models for each SCTG code (with derived distance, trained on all training data)

	LR	DT	SVM	MLP	KNN	NB	RF	GB	BAG	ADA	Extra	XGBoost	Vote	Stack
Accuracy	0.662	0.869	0.423	0.867	0.867	0.586	0.902	0.806	0.901	0.706	0.887	0.853	**0.908**	**0.908**
BA	0.563	0.820	0.340	0.814	0.814	0.430	0.836	0.691	**0.838**	0.546	0.819	0.759	0.834	0.833
Precision	0.662	0.869	0.467	0.866	0.866	0.585	0.902	0.807	0.901	0.707	0.888	0.855	0.908	**0.910**
Recall	0.662	0.869	0.423	0.867	0.867	0.586	0.902	0.806	0.901	0.706	0.887	0.853	0.908	**0.909**
F1	0.659	0.869	0.423	0.866	0.866	0.581	0.901	0.803	0.900	0.701	0.886	0.851	0.907	**0.908**
$\hat{\sigma}$(Accuracy)	0.001	0.119	0.126	0.001	0.010	0.128	0.000	0.048	0.045	0.081	0.076	0.001	0.000	0.000

(g) Ensemble model combining (b)–(d) (with and without derived distance, trained on all training data)

	Vote		Stack	
	Without derived distance	With derived distance	Without derived distance	With derived distance
Accuracy	0.851	0.923	0.862	**0.928**
BA	0.755	0.851	0.798	**0.871**
Precision	0.854	0.924	0.861	**0.928**
Recall	0.851	0.923	0.862	**0.928**
F1	0.849	0.922	0.861	**0.927**
$\hat{\sigma}$(Accuracy)	0.000	0.000	0.000	0.000

To effectively compare different models, various evaluation metrics were computed. These metrics include accuracy, balanced accuracy (BA), precision, recall, and F1 score. Simply measuring accuracy alone may not accurately represent the distribution of data between the air and other modes because they constitute only a small portion of the overall data set (as illustrated in Figure 54). To address this issue, the BA considered the accuracy by averaging them across all five modes to mitigate the impacts of data imbalance issue. All metrics were measured based on the number of records without any further weighting (e.g., by values) to ensure comparability with previous studies.

Based on the analysis, the stacking model was selected as the final model. To further examine the stacking model performance, Figure 54 displays the results by SCTG code (left) and NAICS code (right). Overall, the final model proved effective across most SCTG and NAICS types, except for the air mode for some SCTG and NAICS types because of a severe data imbalance.

Furthermore, by adjusting the probability threshold for each mode in decision-making, a receiver operating characteristic curve was generated with the corresponding Area Under the Curve (AUC) score reported in Figure 55. For the AUC score, the closer to 1, the better the model performed. By comparison, ensemble methods like stacking or voting model (the third row of subplots) can improve the AUC score by combining results from different base learners (the first two rows of subplots). Figure 55 will be discussed in more detailed way in the following section.

Model Results

Table 20 presents the results of evaluating various models under different specifications. Our findings demonstrate that the performance of these models can be significantly enhanced by implementing three key improvements: incorporating the distance between origin and destination for each freight mode; developing local models for each category; and adopting a voting/ensemble approach to combine different types of local and unified models.

All three suggested improvements are helpful in enhancing the performance. Through the use of derived distance, the accuracy of our model on the testing data set improved significantly, from 73% accuracy (best of part (a)) to 84% accuracy (best of part (b)). Furthermore, when dedicated models were constructed by different SCTG and NAICS categories, the performance increased even further from 84.1% (best of part (a)-(b)) to 90.8% (best of part (c)-(d)) and 92.2% (best of part (e)-(f)) for NAICS and SCTG, respectively. Finally, by using stacking method to combine all models, we achieved an accuracy of 92.8% and a BA of 87.1%. Moreover, the standard errors of accuracies are also estimated by bootstrapping over the testing dataset, as displayed in the final row of each section in Table 20. Ensembled model (e.g., stacking or voting method) also shown the advantage of stabilizing performance over testing sets. Most ensembled models exhibited a standard deviation of accuracies that was below 0.005%. Consequently, these values were rounded down to 0.000 in Table 20. Note that all results in Table 20 are run on over 1 million testing cases, and the standard deviation will be larger as the number

testing samples gets lower. These results demonstrate that the model framework is reliable and suitable for use in various applications, including freight modal split analysis, traffic assignment studies, and sustainability assessments.

Other metrics to compare different predicting models were also applied. As mentioned, tree-based models provide the probabilities of selecting each mode. Thus, a threshold probability was used as a decision boundary to select a specific mode for each freight mode. By varying the decision boundary from 0 to 1 for each mode, a receiver operating characteristic curve was plotted, as depicted in Figure 55. The *x*-axis represents the false positive rate, and the *y*-axis represents the true positive rate. The first and second rows correspond to RF and BAG models, respectively. The third row shows the results of two ensemble methods. Each plot displays five lines that represent the receiver operating characteristic curve for each of the five modes indicated by the lower right legends. The AUC was used to evaluate model performance, where a value closer to 1 indicates better performance. The average AUC for all five plots is displayed in each plot. Using the RF or bagging method, the models built by NAICS and SCTG codes outperformed the unified ("one model for all") models. Notably, the voting and stacking models (the third row) yielded significantly better performance than applying the base learners alone (the first and second row).

The behavior of the high-performance models was examined. The left and right subplots in Figure 56 correspond to the variable importance of the RF model and the level-2 stacking model (i.e., the meta-learner), respectively. One RF model was applied to the overall data set for Figure 56 (left). The routed distances of different modes ("M1"–"M5" and "I1"–"I5") accounted for a significant portion of variations in the model. Conversely, a different pattern is discernible in Figure 56 (right), where five out of top six most "expressive" variables are probabilities of choosing different modes derived from base learners/models. This demonstrates that the meta-learner, an XGBoost model, begins to make predictions with the results of base learners before further polishing predictions using other variables (e.g., "v2w," "sw"). Also, the effects of those variables were already considered by the base learners (level-1 model), as shown by Figure 56 (left).

Recently, many methods have been developed for interpreting machine learning models. Among these, the LIME (Local Interpretable Model-Agnostic Explanations) approach is widely used in agnostic-model interpretation by fitting local linear additive models to approximate the model to be explained for individual inputs (Ribeiro et al., 2016). Furthermore, the SHAP (Shapely Additive Explanations) method (Lundberg et al., 2019; Uddin et al., 2023), which is based on game theory and focuses on global interpretation, was developed to evaluate the effects of different model features and has some theoretical and practical benefits. Recently, specialized algorithms are available for generating and efficiently analyzing SHAP values (Lundberg et al., 2019). Although the computational details of SHAP values can be tedious, the SHAP value only describes the "marginal impacts" of a specific feature on the final outputs. For this purpose, Figure 57 shows a force plot for one randomly chosen sample, where the probability of selecting each mode is interpreted by the contribution of each feature. In Figure 57 (a), value to weight

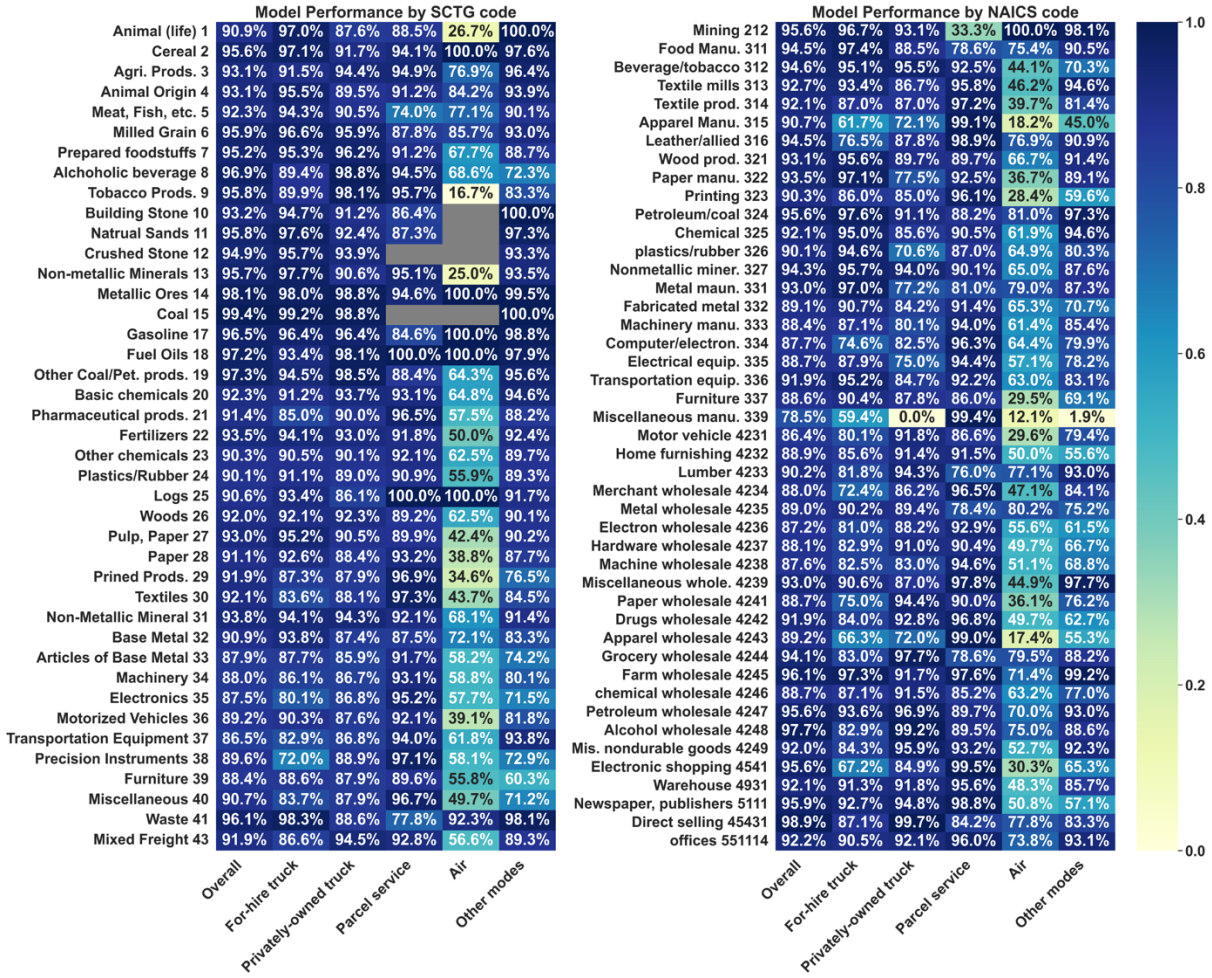


Figure 54. Accuracy of the final model for each freight mode by SCTG code (the left column) and by NAICS code (the right column)

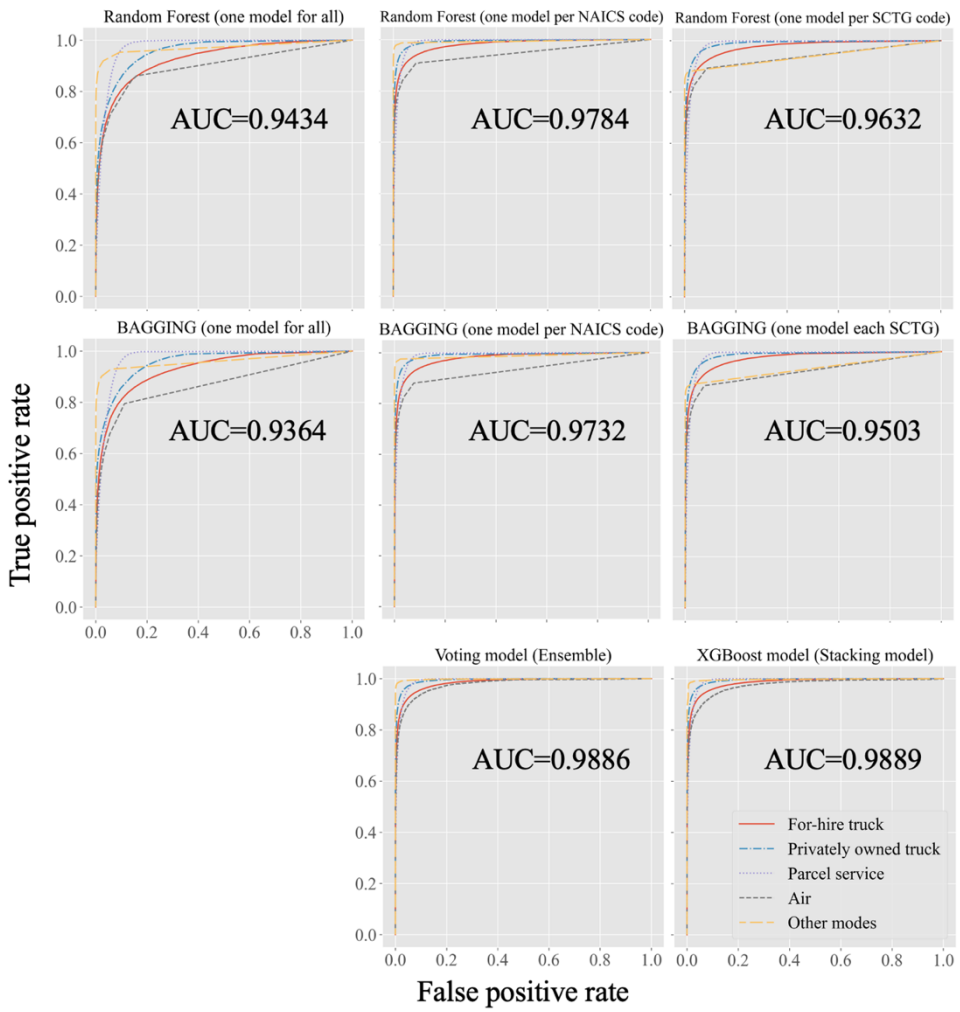


Figure 55. Evaluating performance by receiver operating characteristic's AUC for various models

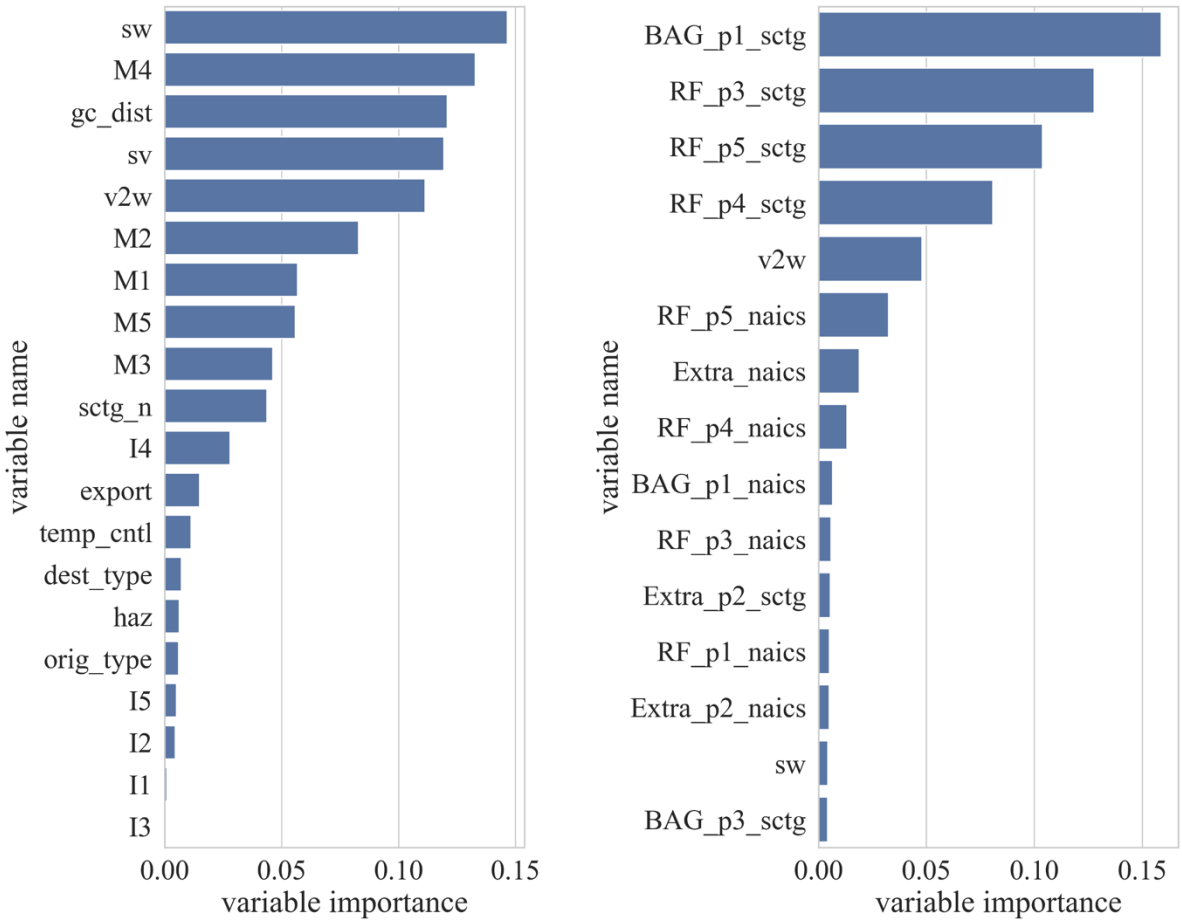


Figure 56. Variable importance for the (left) global RF model and (right) XGBoost stacking model

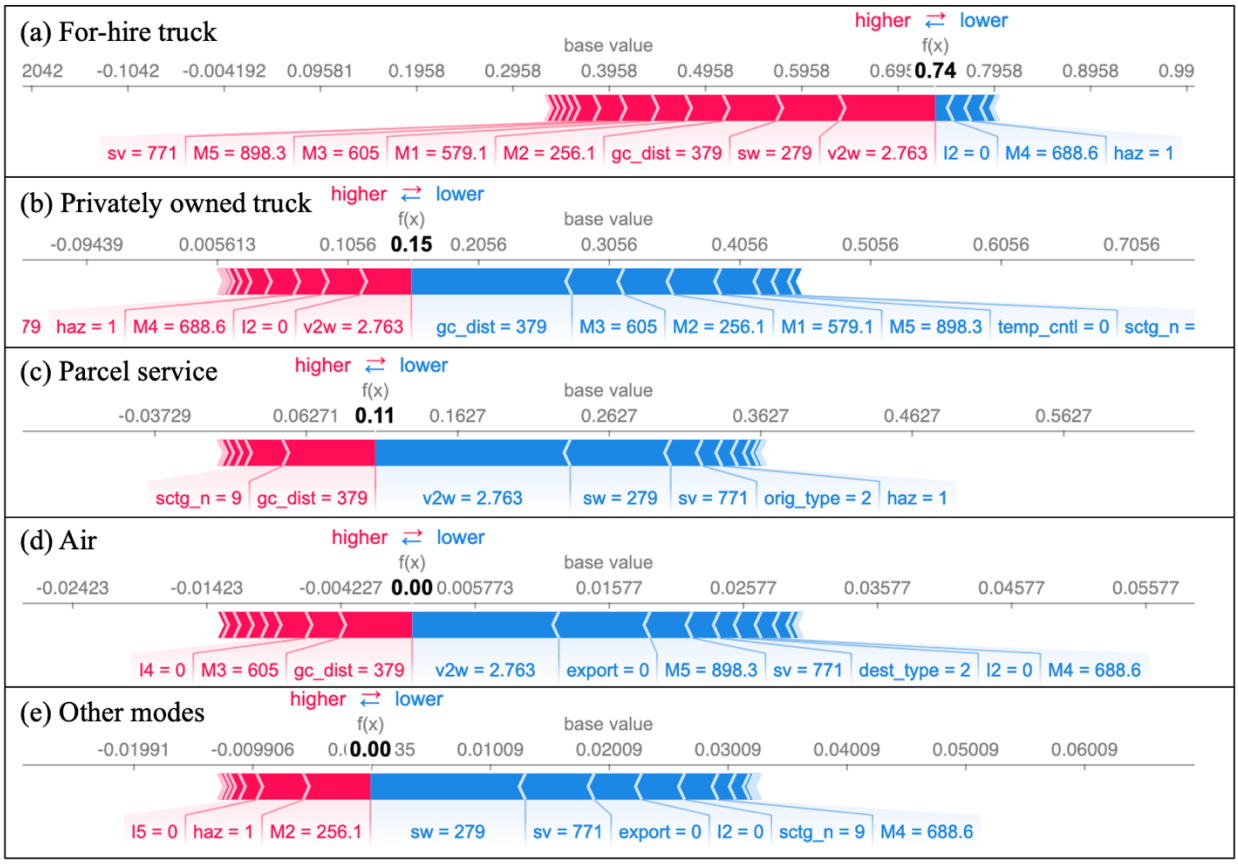


Figure 57. A force diagram of SHAP values of one DT model over one testing data sample

("v2w"), single weight ("sw"), and great circle distance ("gc_dist") are the top three reasons to predict the for-hire truck mode, whereas the same variables on other modes have different magnitudes of influence.

The SHAP method can also be applied to RF models, as shown in Figure 58. Figure 58 (a) displays the ratio of SHAP values across all the five modes by measuring the mean absolute SHAP values for each feature over 500 randomly selected samples. The remaining five plots (Figure 58 (b)–(f)) describe the SHAP values correspond to each of the five modes' output probabilities. The *x*-axis shows the SHAP values. The feature value of each dot is denoted from high to low using a red to blue scale. The features are listed by their total SHAP impacts from top to bottom. A dot with a negative effect falls at the left side of the *x*-axis, and one with a positive effect falls at the right side of the *x*-axis. As shown, different freight modes have different influential features. A single weight ("sw") had the highest SHAP value for for-hire trucks and parcel service, whereas the great circle distance ("gc_dist") had the highest SHAP value for privately owned trucks.

Figure 58 reveals several nontrivial patterns that offer significant insights. For instance, exported products are more likely to be transported via air or other modes. Furthermore, the great circle distance ("gc_dist") has a strong influence on privately owned trucks' mode choice. Also, privately owned trucks rarely travel for distances greater than 500 miles in great circle distance. Notably, several features with high SHAP values for privately owned trucks include routed distances for different modes (e.g., "M4"). In contrast, single weight ("sw"), single value ("sv"), and value-to-weight ("v2w") ratios are the most influential features for for-hire trucks. This could be due to shippers operating for-hire trucks within an open market and competing with other long-distance freight modes who occasionally prefer transporting products with high single weight and single value but low value to weight. Notably, low value to weight is a relative term compared to commodities shipped via parcel service.

In addition to interpreting SHAP values on a feature-by-feature basis, we assessed the interaction values between two variables using the SHAP dependency plot, as illustrated in Figure 59. Each plot displays single weight ("sw") and the corresponding SHAP value. The dots denote another selected feature that has a significant interaction effect for the given freight mode. The red to blue legend indicates the feature values from highest to lowest. The results show that there is an interactive relationship between single weight ("sw") and distance metrics (e.g., "gc_dist"). For-hire truck and parcel service are sensitive to traveling distance given a fixed single weight, and privately owned trucks have reduced SHAP values for single weights when traveling long distances. Moreover, for air and other modes, the variable for indicating export shipment (0-domestic and 1-export) may have minor interaction effects on fixed single weight, as indicated by the SHAP values.

The previous SHAP plots illustrate the patterns of level-1 models, such as a global RF model. Analyzing the SHAP effects on level-2 of the stacking model or on concatenated

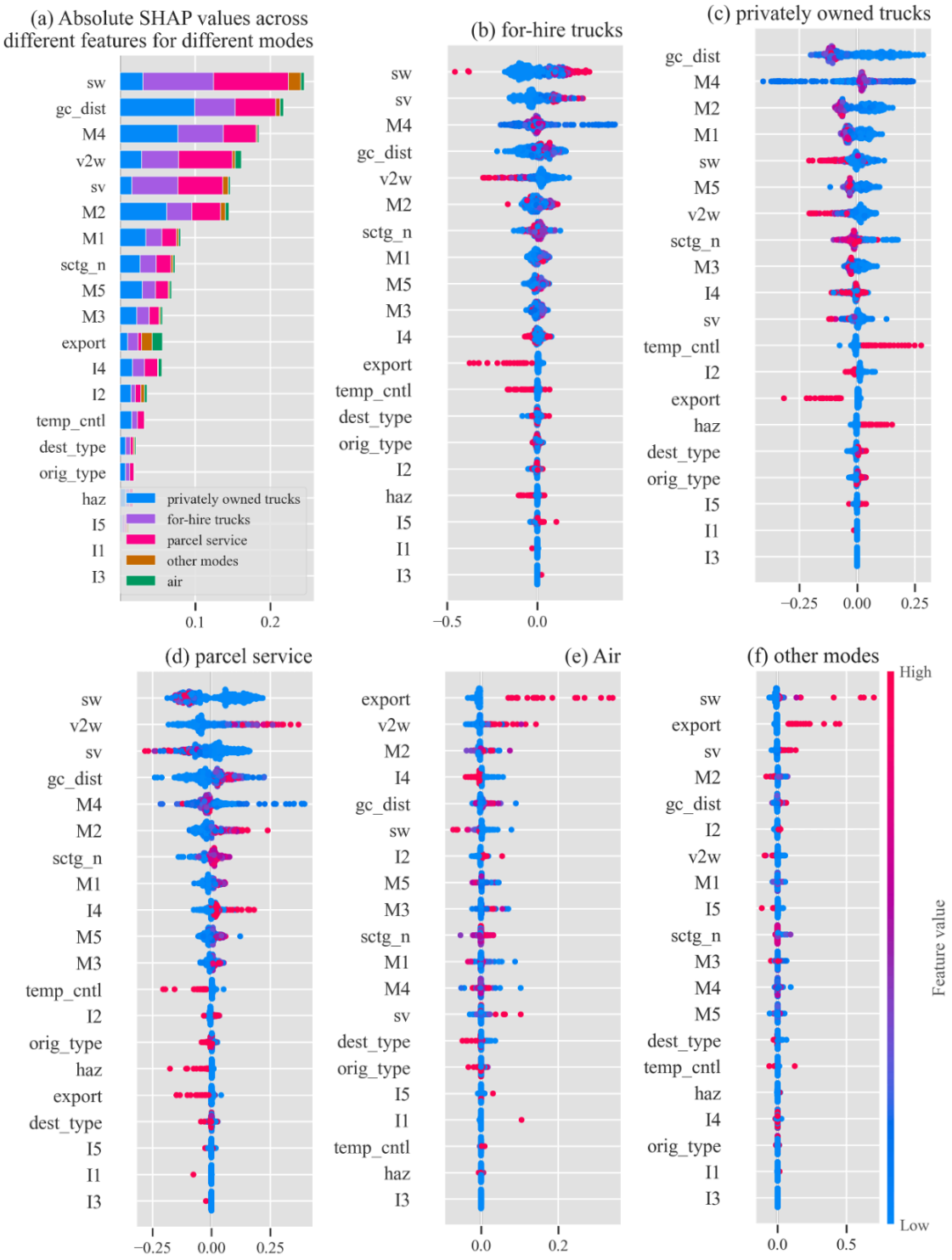


Figure 58. (a) Summary plot of absolute SHAP values and (b–f) swarm plots of each mode choice using the SHAP game theory approach to explain feature influences on the overall RF model

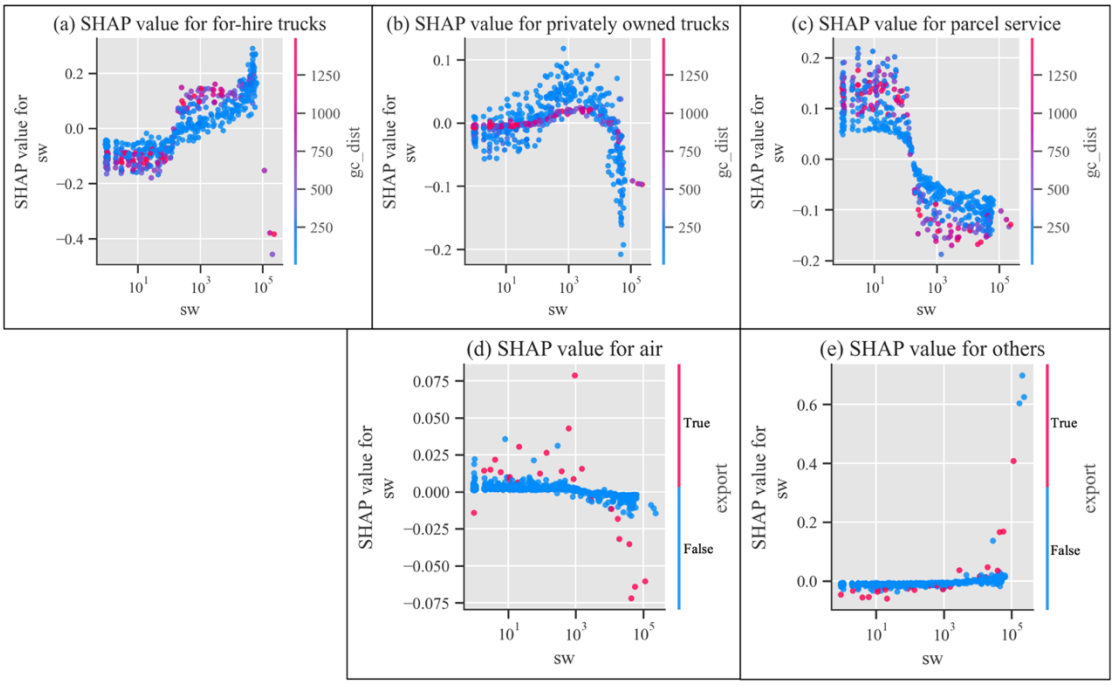


Figure 59. SHAP dependence plot of the single weight ("sw") feature and some interaction features for each freight mode

models split by different commodity or industrial types can offer further insights into interpreting models. Figure 60 presents the results of such analyses, revealing that the level-2 meta-learner simply averaged the probabilities of each model class outputted by the level-1 models. Many features are probability outputs from level-1 models, including "BAG_p1_sctg," which denotes the probabilities of selecting for-hire trucks for the "sctg" segmented bagging tree model. Furthermore, information such as single weight ("sw") and value to weight ("v2w") still contribute to model interpretation based on their SHAP values. Although binary dummy variables are used for constructing level-1 models based on SCTG and NAICS types and their effects are summed, their SHAP values do not rank among the top 10 features for all freight modes.

In summary, the proposed modeling framework and methodologies can significantly improve the performance of mode choice modeling. Besides using basic performance metrics, receiver operating characteristic can also be helpful in visualizing the model's performance. The machine learning model results were also interpreted, generating useful insights about feature importance with the help of the SHAP method.

Conclusions

In this study, we developed a high-performance freight modal choice prediction model using the 2017 CFS PUF data set. The potential applications of such a model are wide-ranging and significant. Using some basic shipment and geographical information such as value, weight, commodity type, origin, and destination, our model predicted the probabilities of choosing available freight modes. Moreover, using ensemble techniques (e.g., stacking) for machine learning models by combining models built upon different sets of data records from different aspects, we significantly improved the model performance for freight mode prediction (or more generally, transportation mode prediction).

During the course of model development, numerous lessons were learned. First, although ensemble methods are advantageous for enhancing the overall model performance, they may not be effective when solely applied to a specific type of model (e.g., local models by SCTG). Furthermore, the performance of non-tree–based models was comparatively lower than that of tree-based models. Consequently, their inclusion in the ensemble did not lead to any significant improvement. Second, we attempted to identify different base learners from various categories dynamically to generate a dynamic model that combines different kinds of machine learning models. We hypothesized that certain commodities or industries may possess unique decision-making logic and so, a simple model could outperform tree-based models for some commodities. However, in practice, tree-based methods like RF and BAG demonstrated superior performance across all categories, including those with few data records. Finally, the use of neural networks as meta-learners in stacking methods was attempted without success.

Some limitations of this study stem from the CFS PUF data set. First, because it is a survey sampling among shippers, the data set does not provide uniform coverage of all

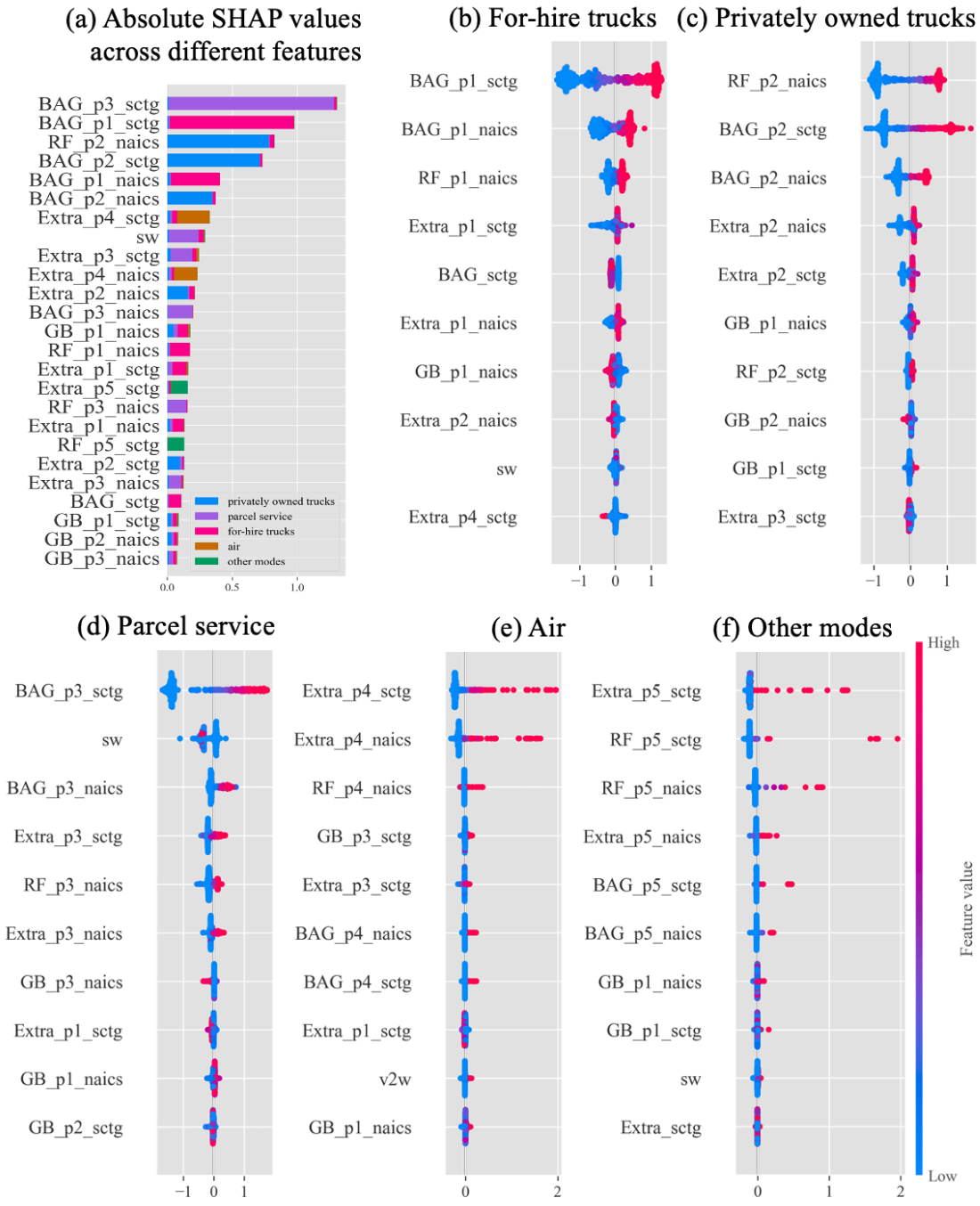


Figure 60. (a) Summary plot and (b–f) swarm plots of the level-2 stacking model

freight movements (Comparison of Commodity Flow Survey (CFS) and Freight Analysis Framework (FAF) for 2017 Domestic Freight Flows, 2022). Another constraint of this research is the lack of consideration given to shipping time and its variability. Although logistics travel time could be estimated by reconstructing travel networks for various modes of transportation, this task is not straightforward because much of the logistics time is spent waiting at fixed locations such as warehouses, in addition to the travel time between locations. Furthermore, pricing or cost bids across different freight modes are essential factors when providing freight service to customers but are not incorporated in the data set. The only reported value is the value of a single product.

In this study, freight modes were classified into five categories to facilitate comparison with previous research. Water-related modes, train-related modes, and pipelines were all grouped under other modes because of their relatively low percentage of shipments. A nested model can be developed to further differentiate those modes. Also, one can apply oversampling methods to enhance the results for the air mode, which composed of less than 2% of trips in most commodity categories. The most challenging aspect of the study was predicting air mode shipments because of a data imbalance resulting from their small proportion in the data set. To address this issue in future work, we will focus on improving predictions for infrequent modes (e.g., air, other modes) through the use of different mode weights.

Statements and Acknowledgements

Statement: During the preparation of this work, the authors used gpt-3.5-turbo in order to improve the fluency and clarity of the writing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgement: The authors appreciate the funding and research opportunities provided by the Graduate Advancement Training and Education for collaborate research between the University of Tennessee, Knoxville, and the US Department of Energy's Oak Ridge National Laboratory.

References

2017 Commodity Flow Survey (CFS) Public Use File (PUF) Data Users Guide. (n.d.). www.census.gov/programs-surveys/cfs.html.

Alcohol Transport Permit Requirements by State | Brew Movers. (n.d.). Retrieved May 3, 2023, from https://brewmovers.com/articles/guide-to-alcohol-transportation-permits-by-state/

Arencibia, A. I., Feo-Valero, M., García-Menéndez, L., & Román, C. (2015). Modelling mode choice for freight transport using advanced choice experiments. *Transportation Research Part A: Policy and Practice*, *75*, 252–267. https://doi.org/10.1016/j.tra.2015.03.027

Bennett, J., & Lanning, S. (2007). *The Netflix Prize*.

Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, *37*(2), 587–603. https://doi.org/10.1016/j.ijforecast.2020.07.007

Burks, S. v, & Monaco, K. (2019). *Monthly La bor BLS Review BLS Is the U.S. labor market for truck drivers broken?*

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.

Cheng, L., Chen, X., De Vos, J., Lai, X., & Witlox, F. (2019). Applying a random forest method approach to model travel mode choice behavior. *Travel Behaviour and Society*, *14*, 1–10. https://doi.org/10.1016/j.tbs.2018.09.002

Comparison of Commodity Flow Survey (CFS) and Freight Analysis Framework (FAF) for 2017 Domestic Freight Flows. (n.d.). https://doi.org/10.21949/1526410

Hagenauer, J., & Helbich, M. (2017). A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Systems with Applications*, *78*, 273–282. https://doi.org/10.1016/j.eswa.2017.01.057

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009a). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009b). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

Hobbs, J. E. (2020). Food supply chains during the COVID-19 pandemic. *Canadian Journal of Agricultural Economics*, *68*(2), 171–176. https://doi.org/10.1111/cjag.12237

Holguín-Veras, J., Kalahasthi, L., Campbell, S., González-Calderón, C. A., & (Cara) Wang, X. (2021). Freight mode choice: Results from a nationwide qualitative and quantitative research effort. *Transportation Research Part A: Policy and Practice*, *143*, 78–120. https://doi.org/10.1016/j.tra.2020.11.016

Keya, N., Anowar, S., Bhowmik, T., & Eluru, N. (2021). A joint framework for modeling freight mode and destination choice: Application to the US commodity flow survey data. *Transportation Research Part E: Logistics and Transportation Review*, *146*. https://doi.org/10.1016/j.tre.2020.102208

Keya, N., Anowar, S., & Eluru, N. (2018). Freight Mode Choice: A Regret Minimization and Utility Maximization Based Hybrid Model. *Transportation Research Record*, *2672*(9), 107–119. https://doi.org/10.1177/0361198118782256

Keya, N., Anowar, S., & Eluru, N. (2019). Joint model of freight mode choice and shipment size: A copula-based random regret minimization framework. *Transportation Research Part E: Logistics and Transportation Review*, *125*, 97–115. https://doi.org/10.1016/j.tre.2019.03.007

Kim, E. J. (2021). Analysis of Travel Mode Choice in Seoul Using an Interpretable Machine Learning Approach. *Journal of Advanced Transportation*, *2021*. https://doi.org/10.1155/2021/6685004

Koren, Y. (2009). *The BellKor Solution to the Netflix Grand Prize*. www.netflixprize.com/leaderboard

Larranaga, A. M., Arellana, J., & Senna, L. A. (2017). Encouraging intermodality: A stated preference analysis of freight mode choice in Rio Grande do Sul. *Transportation Research Part A: Policy and Practice*, *102*, 202–211. https://doi.org/10.1016/j.tra.2016.10.028

Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems*, *94*, 27–39. https://doi.org/10.1016/j.future.2018.11.004

Lundberg, S. M., Allen, P. G., & Lee, S.-I. (n.d.). *A Unified Approach to Interpreting Model Predictions*. https://github.com/slundberg/shap

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2019). *Explainable AI for Trees: From Local Explanations to Global Understanding*. http://arxiv.org/abs/1905.04610

Mangal, A., & Kumar, N. (2016). Using big data to enhance the bosch production line performance: A Kaggle challenge. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, 2029–2035. https://doi.org/10.1109/BigData.2016.7840826

Monios, J., & Bergqvist, R. (2019). The transport geography of electric and autonomous vehicles in road freight networks. *Journal of Transport Geography*, *80*. https://doi.org/10.1016/j.jtrangeo.2019.102500

Moschovou, T. P., & Giannopoulos, G. A. (2012). Modeling Freight Mode Choice in Greece. *Procedia - Social and Behavioral Sciences*, *48*, 597–611. https://doi.org/10.1016/j.sbspro.2012.06.1038

Norojono, O., & Young, W. (2003). A stated preference freight mode choice model. *Transportation Planning and Technology*, *26*(2), 1. https://doi.org/10.1080/715020600

Nowlan, A., Fine, J., O'connor, T., & Burget, S. (2021). Pollution accounting for corporate actions: Quantifying the air emissions and impacts of transportation system choices case study: Food freight and the grocery industry in los angeles. *Sustainability (Switzerland)*, *13*(18). https://doi.org/10.3390/su131810194

Omrani, H. (2015). Predicting travel mode of individuals by machine learning. *Transportation Research Procedia*, *10*, 840–849. https://doi.org/10.1016/j.trpro.2015.09.037

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel V. and Thirion, B., Grisel, O., Blondel, M., Prettenhofer P. and Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *Model-Agnostic Interpretability of Machine Learning*. http://arxiv.org/abs/1606.05386

Rich, J., Holmblad, P. M., & Hansen, C. O. (2009). A weighted logit freight mode-choice model. *Transportation Research Part E: Logistics and Transportation Review*, *45*(6), 1006–1019. https://doi.org/10.1016/j.tre.2009.02.001

Rudin-Brown, C. M., Harris, S., & Rosberg, A. (2019). How shift scheduling practices contribute to fatigue amongst freight rail operating employees: Findings from Canadian accident investigation. *Accident Analysis and Prevention*, *126*, 64–69. https://doi.org/10.1016/j.aap.2018.01.027

Samimi, A., Kawamura, K., & Mohammadian, A. (2011). A behavioral analysis of freight mode choice decisions. *Transportation Planning and Technology*, *34*(8), 857–869. https://doi.org/10.1080/03081060.2011.600092

Shen, G., & Wang, J. (2012). A Freight Mode Choice Analysis Using a Binary Logit Model and GIS: The Case of Cereal Grains Transportation in the United States. *Journal of Transportation Technologies*, *02*(02), 175–188. https://doi.org/10.4236/jtts.2012.22019

Tamim Kashifi, M., Jamal, A., Samim Kashefi, M., Almoshaogeh, M., & Masiur Rahman, S. (2022). Predicting the travel mode choice with interpretable machine learning techniques: A comparative study. *Travel Behaviour and Society*, *29*, 279–296. https://doi.org/10.1016/j.tbs.2022.07.003

Töscher, A., Jahrer, M., & Bell, R. M. (2009). *The BigChaos Solution to the Netflix Grand Prize*.

Uddin, M., Anowar, S., & Eluru, N. (2021). Modeling freight mode choice using machine learning classifiers: a comparative study using Commodity Flow Survey (CFS) data. *Transportation Planning and Technology*, *44*(5), 543–559. https://doi.org/10.1080/03081060.2021.1927306

Uddin, M., Hwang, H. L., & Hasnine, M. S. (2023). An interpretable machine learning framework to understand bikeshare demand before and during the COVID-19 pandemic in New York City. *Transportation Planning and Technology*. https://doi.org/10.1080/03081060.2023.2201280

Wang, Y., Ding, C., Liu, C., & Xie, B. (2013). An Analysis of Interstate Freight Mode Choice between Truck and Rail: A Case Study of Maryland, United States. *Procedia - Social and Behavioral Sciences*, *96*, 1239–1249. https://doi.org/10.1016/j.sbspro.2013.08.141

Xie, Y., Lu, W., Wang, W., & Quadrifoglio, L. (2012). A multimodal location and routing model for hazardous materials transportation. *Journal of Hazardous Materials*, *227–228*, 135–141. https://doi.org/10.1016/j.jhazmat.2012.05.028

Yin, R., Tran, V. H., Zhou, X., Zheng, J., & Kwoh, C. K. (2018). Predicting antigenic variants of H1N1 influenza virus based on epidemics and pandemics using a stacking model. *PLoS ONE*, *13*(12). https://doi.org/10.1371/journal.pone.0207777

Zenezini, G., Lagorio, A., Pinto, R., Marco, A. de, & Golini, R. (2018). *The Collection-And-Delivery Points Implementation Process from the Courier, Express and Parcel Operator's Perspective*. *51*(11), 594–599. https://doi.org/10.1016/j.ifacol.2018.08.383

Zhang, Z., Turla, T., & Liu, X. (2021). Analysis of human-factor-caused freight train accidents in the United States. *Journal of Transportation Safety and Security*, *13*(10), 1157–1186. https://doi.org/10.1080/19439962.2019.1697774

Zhao, X., Yan, X., Yu, A., & Van Hentenryck, P. (2020). Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. *Travel Behaviour and Society*, *20*, 22–35. https://doi.org/10.1016/j.tbs.2020.02.003

CONCLUSION

In conclusion, traffic flow and freight behaviors can be estimated using different data sets from different perspectives. To estimate truck flows, the first two chapters gather traffic information at fixed locations using radar detectors and CCTV cameras. The first chapter is based on Linear Programming to estimate the optimal number of large trucks given occupancies and volumes. The second chapter focuses on extracting traffic information from low-resolution camera data by designing a framework to exploit multiple deep learning algorithms. Both methods are based on detectors that do not generate truck information directly. Besides proposing two innovative methods, many detectors do classify vehicles by their type, namely double loop detectors and Weigh-in-Motion detectors. The methods proposed in the first two chapters can extend the ability to identify truck flows to many more cases when the detectors are not designed for identifying trucks directly.

After getting truck flow information at fixed locations, it is also possible to impute information over the traffic network. The third chapter tries to impute the information on traffic volume by class over the national highway network using an iterative algorithm. The detectors in this case can generate traffic flow information for different vehicle types. Finally, the fourth chapter explored the possibility of getting data from a different angle giving the Commodity Flow Survey, a nationwide survey revealing shipment-level travel records. There are three important freight modes in this chapter related to truck traffic: (1) for-hire trucks; (2) privately-owned trucks; and (3) parcel service. Note that it is possible to identify three kinds of trucks by their appearance (e.g., colors and shapes of a vehicle). In other words, the technique proposed in Chapter II can be used to infer a vehicle's class using CCTV cameras.

In summary, each chapter investigates one type of data set to extract truck and freight information. Although multiple types of detectors are investigated, other data sources are not mentioned or investigated in this dissertation. One type is the telematic data recording GPS trajectories of operating trucks. While they are very useful for optimizing operations at a company or institution level, it is less effective in generating traffic information for a metropolitan region as the penetration level is low or biased among the population of trucks. The dissertation mainly focuses on learning the general impacts of all trucks involved instead of the impacts of a small fleet of trucks. The dissertation mainly focuses on identifying the impacts of trucks on the skeleton networks. The fourth chapter hints that although truck traffic is very important, there is a superset of different freight modes that shippers can potentially choose from. By comparing truck flow changes over time, useful insights can be extracted for transportation planning, construction, repairing, sustainability analysis, etc.

VITA

Diyi Liu was born in Beijing, China. He received his bachelor's degree from Southeast University in Nanjing, China in 2016. From 2016-2020, he received some other master's degrees at UC Berkeley, Columbia University, and Georgia Institute of Technology. He joined the University of Tennessee, Knoxville in Spring 2021. In October 2023, he became a Ph.D. candidate at the University of Tennessee, Knoxville (UTK). During his stay at UT, Knoxville, he is taking an internship at the Oak Ridge National Laboratory (ORNL) under the GATE fellowship program, a joint program between UTK and ORNL. His research focuses on tackling transportation problems using machine learning algorithms.