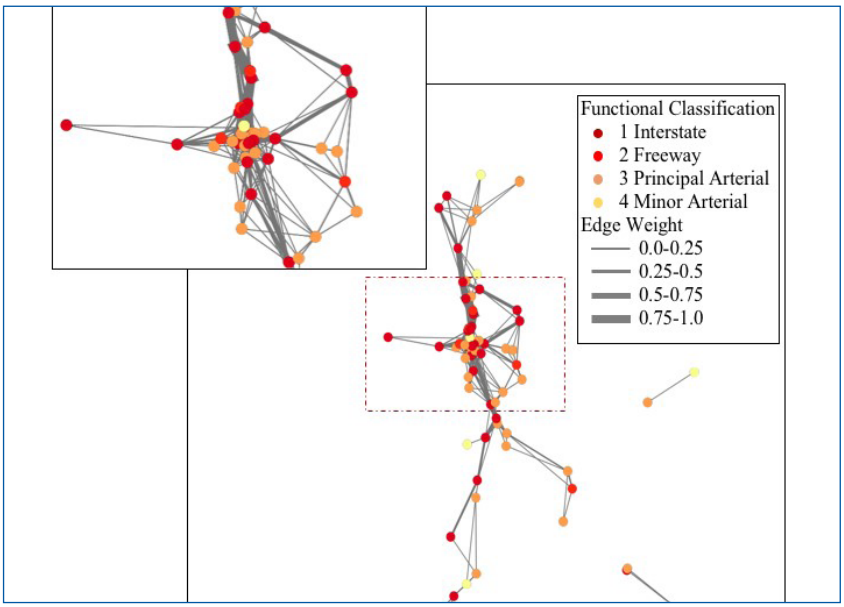
MOUNTAIN-PLAINS CONSORTIUM

MPC 21-428 | X.C. Liu and Z. Yi

Big Transportation Data Analytics



MOUNTAIN-PLAINS CONSORTIUM REGION 8

A University Transportation Center sponsored by the U.S. Department of Transportation serving the Mountain-Plains Region. Consortium members:

Colorado State University
North Dakota State University
South Dakota State University
University of Colorado Denver
University of Denver
University of Utah
Utah State University
University of Wyoming

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
MPC-543		

4. Title and Subtitle	5. Report Date
Big Transportation Data Analytics	March 2021
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
X.C. Liu Z. Yi	MPC 21-428

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
University of Utah Salt Lake City, UT 84112	11. Contract or Grant No.

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Mountain-Plains Consortium North Dakota State University PO Box 6050, Fargo, ND 58108	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Supported by a grant from the US DOT, University Transportation Centers Program

16. Abstract

Traffic volume data is crucial in many applications, including transportation operation analysis, congestion management, accident prevention, etc. Yet an extensive capture of accurate volume information on a large-scale network can be difficult and costly. This research focuses on hourly traffic volume prediction in a statewide network using spatial-temporal features and heterogenous data sources. We present a classic machine learning technique - support vector machine (SVM) and compare its efficiency for traffic volume prediction with traditional estimation method. Further, the study develops an innovative spatial prediction method. The method is built off a state-of-the-art tree ensemble model - extreme gradient boosting tree (XGBoost) - to handle the large-scale features and hourly traffic volume samples. Moreover, spatial dependency among road segments is considered using graph theory. Specifically, we created a traffic network graph leveraging probe trajectory data and implemented a graph-based approach - breadth first search (BFS) - to search neighboring sites in this graph for computing spatial dependency. The proposed spatial dependency feature is subsequently incorporated as a new feature fed into XGBoost. The proposed methods are applied to 101 continuous count station (CCS) sites in the State of Utah. Prediction accuracy and training time are compared across the proposed models.

17. Key Word	18. Distribution Statement
data analysis, estimating, forecasting, information processing, machine learning, traffic counting, traffic volume, vehicle miles of travel	Public distribution

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	41	n/a

BIG TRANSPORTATION DATA ANALYTICS

Xiaoyue Cathy Liu, Ph.D., P.E.
Associate Professor
Department of Civil and Environmental Engineering
University of Utah
Salt Lake City, Utah, 84112
Phone: (801) 587-8858
Email: cathy.liu@utah.edu

Zhiyan Yi
Ph.D. Student
Department of Civil and Environmental Engineering
University of Utah
Salt Lake City, Utah, 84112
Email: zhiyan.yi@utah.edu

March 2021

Acknowledgements

The authors acknowledge the Mountain Plain Consortium (MPC) and the Utah Department of Transportation (UDOT) for funding this research, and the following individuals from UDOT on the Technical Advisory Committee for helping to guide the research:

- Nicholas Black
- Jordan Backman
- Jamie Mackey
- Rukhsana Lindsey

Disclaimer

ABSTRACT

Traffic volume data are crucial in many applications, including transportation operation analysis, congestion management, and accident prevention. Yet an extensive capture of accurate volume information on a large-scale network can be difficult and costly. This research focuses on hourly traffic volume prediction in a statewide network using spatial-temporal features and heterogeneous data sources. We present a classic machine learning technique – support vector machine (SVM) – and compare its efficiency for traffic volume prediction with traditional estimation methods. Further, the study develops an innovative spatial prediction method. The method is built off a state-of-the-art tree ensemble model – extreme gradient boosting tree (XGBoost) – to handle the large-scale features and hourly traffic volume samples. Moreover, spatial dependency among road segments is considered using graph theory. Specifically, we build a traffic network graph using probe trajectory data, and implemented a graph-based approach – breadth first search (BFS) – to search neighboring sites in this graph for computing spatial dependency. The proposed spatial dependency feature is subsequently incorporated as a new feature fed into XGBoost. The proposed methods are applied to 101 continuous count station (CCS) sites in the State of Utah. Prediction accuracy and training time are compared across the proposed models.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Traffic volume is a critical piece of information in many applications, such as transportation long-range planning and traffic operation analysis. Effectively capturing traffic volumes on a network scale is beneficial to transportation systems management & operations (TSM&O). Yet an extensive capture of accurate volume information on a large-scale network can be difficult and costly. Previous literature attempting volume estimation mostly focuses on annual average daily traffic (AADT) prediction using various statistical techniques. On one hand, AADT is not able to reflect traffic flow variation in finer granularity for operational analysis purposes. On the other hand, hourly volume estimation is more challenging due to fluctuations induced by spatial-temporal features. Besides, predicting hourly traffic volumes with high accuracy requires larger datasets and more features that might potentially impact traffic flow. As such, a model's predictive capability and time complexity should be taken into account simultaneously. This research focuses on hourly traffic volume prediction in a statewide network using spatial-temporal features and heterogeneous data sources. We present a classic machine learning technique – support vector machine (SVM) – and compare its efficiency for traffic volume prediction with traditional estimation methods.

Furthermore, in many traffic volume prediction efforts, spatial prediction techniques are widely performed to estimate traffic volumes at sites without sensors. In retrospect, most relevant studies resort to machine learning methods and treat each prediction location independently during the training process, ignoring the potential spatial dependency among them. To address this, we applied a state-of-the-art tree ensemble model – extreme gradient boosting tree (XGBoost) – to handle the large-scale features and hourly traffic volume samples due to the model's powerful scalability. Moreover, spatial dependency among road segments is taken into account in the proposed model using graph theory. Specifically, we build a traffic network graph using probe trajectory data, and implemented a graph-based approach – breadth first search (BFS) – to search neighboring sites in this graph for computing spatial dependency. The proposed spatial dependency feature was subsequently incorporated as a new feature fed into XGBoost. The proposed model was tested on the road network in the state of Utah. Numerical results not only indicated high computational efficiency of the proposed model, but also demonstrated significant improvement in prediction accuracy of hourly traffic volume compared with the benchmarked models.

1. INTRODUCTION

1.1 Problem Statement

Traffic volume, or throughput, serves as a crucial indicator in highway performance and transportation operation analysis. Highly granular traffic volume provides key information in identifying congested roadways, assisting traffic redistribution, and implementing accident prevention strategies (Cheng, Lu, Peng, & Wu, 2018; El Esawey, Mosa, & Nasr, 2015; Karlaftis & Golias, 2002; Liebig, Piatkowski, Bockermann, & Morik, 2017). Furthermore, it is the disaggregated source for calculating annual average daily traffic (AADT). AADT at the network level offers a measure of overall utilization of a highway facility, implies the level of service of roads, and can be used for highway planning, trend studies, and project prioritization (Lam & Xu, 2000). Currently, traffic count (volume) is mainly obtained from sensors such as inductive loop detectors, radar detectors, and/or continuous counting stations (CCSs) (Leduc, 2008). Yet installing sensors with a large network coverage can be impractical and expensive given budget constraints, especially in rural areas (Zhan, Zheng, Yi, & Ukkusuri, 2017). As a result, how to spatially estimate/predict traffic volume to substitute massive sensor deployment has been an intriguing topic over the past decade.

Spatial prediction of traffic volume at locations without sensors is usually conducted by utilizing relevant information, including road characteristics, spatial-temporal features, socioeconomic indices and other factors that may affect traffic flow, to build a model. On top of that, spatial prediction can also be performed by interpolating traffic flows from neighboring roads, as spatial dependency exists among neighboring road segments. However, quantifying spatial correlation between roads is difficult because traffic patterns vary by urban topologies and time, making it hard to capture correlations. From the perspective of aggregation level, traffic volume prediction problems can be classified into small-granularity prediction (e.g., 15-minute or hourly traffic volume) and AADT estimation. Small-granularity volume data are more valuable than AADT for micro-level analysis (Castro-Neto, Jeong, Jeong, & Han, 2009; Z. Chen, Liu, & Zhang, 2016; Cheng, Lu, Peng, & Wu, 2018; Gastaldi, Gecchele, & Rossi, 2014; F. Zhao & Chung, 2001). Yet predicting small-granularity traffic volume is more challenging than AADT. First of all, small-granularity traffic volume varies both spatially and temporally. Miscellaneous factors such as spatial-temporal features and traffic flow characteristics with finer granularity need to be considered (Sekuła, Marković, Vander Laan, & Sadabadi, 2018). Second, data size for small-granularity volume prediction is much larger than for AADT prediction (Zhao, Chen, Wu, Chen, & Liu, 2017). It generally requires more observations and features. To summarize, two main challenges exist for spatial small-granularity traffic volume prediction. The first one is accurately capturing spatial correlations; and the second challenge is incorporating a large amount of historical data points and aforementioned features into building a more efficient model. To this end, a model that is capable of accurately capturing neighboring spatial correlation and efficiently handling big data is required for small-granularity traffic volume prediction.

1.2 Objectives

This research project focuses on predicting hourly traffic volume on road segments distributed in the state of Utah. Currently, UDOT deploys over a hundred CCSs to count traffic volumes at different locations in order to achieve geographic distribution of count information. Although it is doable to move the CCSs across different locations, it is still difficult to capture traffic characteristics of the entire Utah road network within a short time period. As a result, computational methods that are easily implementable on a large-scale dataset and are able to estimate traffic volume with satisfactory accuracy present an attractive alternative.

The primary objective of this project is to implement machine learning (ML) techniques to predict hourly traffic volumes using features that are associated with the variation of traffic volume. To achieve this, spatial-temporal features as well as traffic flow characteristics are collected from multiple sources. Then a fraction of road segments with ground-truth volume data are trained through the proposed models in combination with collected features. Lastly, the constructed models are tested on new locations for performance evaluation.

The secondary objective of the project is to enhance the prediction accuracy by exploring methods to quantify the spatial dependency among road segments. To this end, we present a spatial prediction of hourly traffic volume using the road network in the State of Utah. Considering the large-scale features and observed hourly samples, a state-of-the-art tree ensemble model – extreme gradient boosting tree (XGBoost) – is applied to estimate hourly traffic volume. Compared with other ML methods (e.g. support vector machine (SVM) and deep neural networks), XGBoost model runs more than 10 times faster on a single machine and scales to billions of examples in memory-limited settings (Chen & Guestrin, 2016). Apart from its high computational efficiency, XGBoost is also highlighted for its interpretability, which allows it to rank variables' importance (Pourebrahim, Sultana, Niakanlahiji, & Thill, 2019; Tuv, Borisov, & Torkkola, 2006). Alajali et al. (2018) implemented XGBoost to predict traffic volumes at intersections in the city of Melbourne, where it achieved the lowest mean squared error (MSE) among proposed ML models, including gradient boosting regression tree (GBRT), random forest (RF), regression tree (RT), and SVM. Moreover, to explore the aforementioned spatial dependency among road segments, a weighted graph is constructed leveraging probe trajectory data. The graph uses nodes to represent CCS sites, and weighted edges to delineate their spatial correlation intensity. A graph-based theory Breadth First Search (BFS) is implemented to search neighboring CCS sites and compute a spatial dependency feature. This feature is then utilized as a new feature for prediction purposes.

1.3 Outline of Report

The rest of the report is structured as follows. Chapter 2 summarizes the literature on traffic volume prediction and ML methods. The proposed methodology, including the formulation of SVM, XGBoost, and implementation of BFS algorithm, are explained in Chapter 3. Chapter 4 details the description of data sources. And Chapter 5 presents model performance and spatial correlation analysis. Implications and conclusions are presented in Chapter 6.

2. LITERATURE REVIEWS

Traffic volume estimation problems can be mainly branched into two categories: predicting future volumes at locations equipped with traffic sensors and estimating historical traffic volumes at locations without sensors (also referred to as "spatial traffic volume prediction"). In this study, previous literature in terms of future traffic volume prediction is first reviewed and followed by the historical traffic volume prediction reviews. Further, historical traffic volume prediction literature is divided into two sub-categories: historical traffic volume prediction without considering spatial correlation, and historical traffic volume prediction considering spatial correlation. Finally, the SVM and tree ensemble models are introduced.

2.1 Future Traffic Volume Prediction

Future traffic volume prediction generally involves estimation of immediate future volume at the same locations within a short time period based on historical information. One of the most common solutions to future volume prediction is the time-series method, namely auto-regressive integrated moving average (ARIMA) and its variations. Sarby et al. (2007) explored two forecasting techniques, logistic regression and ARIMA, for daily traffic prediction on Egyptian intercity roads. Historical traffic volume data from 1990 to 2001 are used to forecast traffic volumes for years 2002 and 2003. Their analysis indicates that ARIMA outperforms logistic regression, especially for average monthly and average weekly daily traffic volume estimation. Williams and Hoel (2003) implemented seasonal ARIMA (SARIMA) to predict traffic volume on two freeway locations, one in the United States (I-75) and one in the United Kingdom (M25). Predictions were performed every 15 minutes from 5:00 am to 12:00 pm for an arbitrarily selected weekday. Meanwhile, the predictive performance of SARIMA was tested against heuristic forecasting methods (i.e., random walk, historical average, and deviation from historical average). Results indicated that SARIMA obtains the lowest prediction error on both freeways, with mean absolute percentage error (MAPE) being 8.74% for M74 and 8.97% for I-75. Gavirangaswamy et al. (2013) also assessed the effectiveness of ARIMA-based models on traffic volume prediction. Their empirical tests showed that ARIMA-GARCH outperforms ARIMA and SARIMA, with stable model order across different historical volume records.

2.2 Historical/Spatial Traffic Volume Prediction without Considering Spatial Correlation

Spatial prediction usually aims at estimating traffic volumes at locations without sensors. To achieve this goal, traffic flow characteristics, geographic features, economic indices, and other factors that might impact traffic flow are frequently utilized to model the traffic volume at different locations. Xia et al. (1998) constructed a multiple regression model using roadway characteristics (number of lanes, functional classification), socioeconomic indices (population density, dwelling units), and other factors from 450 non-state roads to estimate AADT in Broward County, Florida. The result shows a strong relationship between those factors and AADT, where the adjusted R^2 of the model is 0.607. To further improve model's prediction accuracy, Zhao and Chung (2001) used a larger dataset by incorporating AADT information from state roads, and the best result of their proposed four models achieves an R^2 of 0.82.

Although the aforementioned parametric methods (i.e., ARIMA-based models and linear regression) are relatively easy to implement and capable of explaining the relationship between independent and dependent variables, those models usually simplify the intricate relationship between traffic volume and potential variables, resulting in lower prediction accuracy. To improve model performance, a plethora of non-parametric approaches have been applied for traffic volume estimation. Castro-Neto et al. (2009)

used SVM to estimate future-year AADT. In Tennessee, 25 counties were tested utilizing AADT values from 1985 to 1999 to estimate AADT in the next five years. The analysis indicated that the average MAPE for SVM is only 2.14% for rural roads, and 2.26% for urban roads, which demonstrates an excellent prediction result. Gastaldi el al. (2014) used artificial neural network (ANN) to estimate AADT from one-week traffic counts. Similarly, in a case study of historical hourly volume prediction in Maryland, Sekuła et al. (2018) estimated hourly volume over 45 CCSs with ANN. The features used for volume prediction consist of temporal-spatial features and probe vehicle data for 45 road segments. Results show that the average R^2 is 0.85 for ANN model. Xu, et al. (2013) implemented a decision tree model – classification and regression tree (CART) – to estimate short-term traffic flow. The proposed model is tested on five freeways in Portland every 15 minutes in one day, and achieves 8.53% MAPE on test set, showing fairly good prediction performance.

2.3 Historical/Spatial Traffic Volume Prediction Considering Spatial Correlation

As mentioned earlier, non-parametric approaches can generate fewer prediction errors for estimation. Yet none of the aforementioned literature utilizes spatial dependency for spatial prediction. Instead, all road segments are treated independently for model construction. In fact, traffic flows are spatially correlated (Kerkman, Martens, & Meurs, 2017; Shi, Gong, Deng, Yang, & Xu, 2018). Ignoring spatial dependency can result in misspecification of models. As a result, spatial prediction via geographical dependency has been investigated as a sub-branch of traffic volume prediction. One of the applicable approaches for exploring spatial correlation is K nearest neighbors (K-NN). Smith and Demetsky (1996) introduced the idea of using K-NN for short-term traffic flow prediction on freeways, and claimed that this non-parametric regression model can provide robust and accurate prediction results. Habtemichael and Cetin (2016) implemented an enhanced K-NN, which applies weighted Euclidean distance and winsorization skill to augment short-term traffic flow forecasting accuracy. Results imply that the proposed method reduces MAPE by more than 25%.

Another stream of research for spatial prediction often resorts to the Kriging-based method. Wang and Kockelman (2009) used ordinary Kriging to forecast AADT across the entire Texas road network. However, ordinary Kriging does not allow analysts to control for point-specific characteristics. In a continued research, the authors (Selby & Kockelman, 2013) applied a more complex model than ordinary Kriging – universal Kriging (UK) – to lower the prediction errors. Yet both ordinary Kriging and UK use point-based interpolation, which may lead to inaccurate estimation for a road segment. To address this issue, Song et al. (2019) performed segment-based regression Kriging (SRK) to assess volume of heavy vehicles in western Australia. However, results from the aforementioned Kriging-based approaches show that prediction error remains high in certain locations, particularly those with low volume and less populated areas. It could be explained by two reasons. First, spatial-based methods usually ignore differences of road characteristics and other factors by locations. Traffic volumes on road segments with different functional classifications (e.g., freeways vs. local streets) and road characteristics can vary significantly. This issue can be simply solved by incorporating additional features to reflect road discrepancies. Another reason is that most Kriging-based approaches use Euclidean distance for interpolation.

Nevertheless, Euclidean distance sometimes cannot reflect correct spatial correlation between two sites (e.g., impedance over two locations). To this end, Euclidean distance can be replaced by other metrics to examine spatial relationships. In a case study of transit ridership estimation for subway stations at New York City (Zhang & Wang, 2014), the authors applied the Kriging method with network distance – a graph-based distance – to resemble the fact that subway stations are connected by tunnels. Similarly, Lowry (2014) used centrality to interpolate AADT spatially in a community's street network, and demonstrated the advantages of applying this graph-based metric over Euclidean distance. In fact, such

graph theory-based concepts have been frequently applied to model social networks, transportation networks, and other networks (Leskovec & Mcauley, 2012). For instance, graphs can dynamically simulate the variation of traffic flow between different zones (Du, Song, Wang, Huang, Yu, & Ruan, 2018), explore the shortest path given two locations (Sun, Yu, Bie, & Song, 2017), and optimize freight transportation in service networks (Kelley, Kuby, & Sierra, 2013). Compared with Kriging-based methods, a weighted network graph can better explain traffic flow patterns. Zhang et al. (2019) leveraged high-resolution bike trajectory data to construct a biking traffic network graph with weights, and identified clusters with high concentration of bike usage via percolation theory. Salamanis et al. (2016) built a large-scale traffic network graph, where each node represents a road and each edge a straight connection between any two nodes, to predict future travel-time. In order to explore potential spatial correlation between roads, a modified BFS is implemented to find the most correlated roads in neighboring regions. This searching technique is further incorporated with Graph Based Lag-STARIMA (GBLS) for travel-time estimation. Compared with other non-parametric models, such as k-NN, RF, and SVM, their proposed method achieves the lowest root mean square error (RMSE) on the two datasets in Berlin and Thessaloniki, respectively. Moreover, the modified BFS method significantly reduced computational complexity. Due to BFS's successful applications in exploring spatial correlation in graph, we therefore adopted this algorithm in this study.

2.4 SVM Model

SVM was originally developed to classify observations given a set of attributes (Cortes and Vapnik, 1995). In other words, the SVM classifier divides different categories in a feature space with a gap as wide as possible. Drucker et al. (1996) introduced SVM for regression – also referred to as SVR. SVR follows the same concept as the SVM classifier, with only minor differences.

A few studies have used SVR in predicting AADT (Castro-Neto et al., 2009; and Khan et al, 2017). Castro-Neto et al. (2009) used SVR with data-dependent techniques, where SVR parameters are computed based on the distribution of a training dataset to estimate AADT on Tennessee highways with short-term automatic traffic recorders. They compared the SVR results with the Holt-Winters exponential smoothing technique and with OLS. It is reported that SVR outperforms both methods. Khan et al. (2017) trained SVM and ANN to estimate AADT from short-duration traffic counts in South Carolina. The result showed that SVR outperforms ANN, traditional factor-methods (currently used by DOTs), and regression models. However, the major weakness of SVR is the dependency of its accuracy on the selection of a kernel function, a cost function, and a maximum error allowed. These parameters are chosen based on experiences and application-specific knowledge. In addition, the SVR is not robust to outliers.

2.5 Tree Ensemble Model

The tree-ensemble model combines a collection of base learners, specifically decision trees, to form a stronger model. It has been frequently and successfully applied to prediction problems due to its capability in lowering variance (e.g., RF, Breiman, 2001), mitigating bias (e.g., GBDT, J. H. Friedman, 2002), and improving computational efficiency. Among tree ensembles, XGBoost is a novel boosting tree recently proposed by Chen and Guestrin (2016). It has received wide popularity among ML challenges. In comparison with traditional tree ensembles (e.g., RF and GBDT), a series of techniques, such as split finding algorithms, data compression, and column block for parallel learning, are applied to make XGBoost a scalable ML system for tree boosting. This trait highlights XGBoost by allowing training on a large dataset efficiently. Meanwhile, as a member of tree ensembles, it enables the evaluation of variable importance based on its interior tree structure. Important features can be consequently identified to offer insights on their impacts to traffic volumes.

3. METHODOLOGY

3.1 SVM Method

SVM is a supervised learning algorithm used for classification and regression analysis, and it can be further divided into linear SVM and non-linear SVM models. For classification problems, given the training samples, $\{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)\}$, where $x_i \in R^m$ is a feature vector with m features, $y_i \in R^1$ is the target value, n is the size of training dataset and m is the size of features, linear SVM constructs a hyperplane (or set of hyperplanes) with the largest margin to separate the samples into different classes. This hyperplane is:

$$\langle w^*, x \rangle + b^* = 0 \tag{1}$$

where $\langle w, x \rangle$ is the dot product of w and x; $w^* \in R^m$ and b^* are the parameters of the hyperplane with the largest soft margin. The corresponding classification function is:

$$f(x_i) = sign(\langle w^*, x_i \rangle + b^*) \tag{2}$$

where $f(x_i)$ is the predicting value for the i^{th} sample. One can derive parameters w^* and b^* by solving the following convex optimization problem:

$$min_{(w,b,\xi)} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{3}$$

$$\text{s.t.} \qquad y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1,2,\dots,n \tag{4}$$

$$\xi_i \geq 0, \qquad i = 1,2,\dots,n \tag{5}$$

where C is the penalty hyperparameter, controlling the trade-off between soft margin and classifying training samples correctly; $\xi_i \in R^m$ are slack variables for the misclassified samples. One can transform and solve the optimization problem through its dual. The above formulation illustrates the linear classification case, where one can use the parameter w^* to interpret the importance of features, similar to linear regression model.

Meanwhile, by conducting kernel trick with the dot product in Eq.(1) to map samples into higher dimensional space, linear SVM can be transformed into non-linear SVM model. This is able to benefit model when the samples are non-linear separable. In this study, we use Gaussian kernel as the kernel function, which is expressed as:

$$\mathrm{K}(x_i, x_j) = e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2} \tag{6}$$

where x_i and x_j correspond to w^* and x in Eq.(1). Non-linear SVM has kernel trick to power its prediction performance. Nevertheless, non-linear SVM requires longer training time to obtain the optimized hyperparameters. In this project, both linear SVM and non-linear SVM are trained to predict hourly traffic counts.

3.2 Graph Theory-Based Spatial Correlation Method

Figure 3.1 shows the methodological framework for predicting hourly traffic volumes on road segments using a proposed graph-based method. A traffic network graph is built on the basis of probe trajectory data, and a graph-based approach – BFS – is applied to extract spatial dependency between CCS sites from the proposed graph. This spatial dependency feature, together with other characteristics collected from heterogeneous sources, is fed into this framework to train the XGBoost. The model is then tested on new locations for performance evaluation. In this section, the graph representation, BFS method, spatial dependency feature computation, and XGBoost model are introduced in detail.

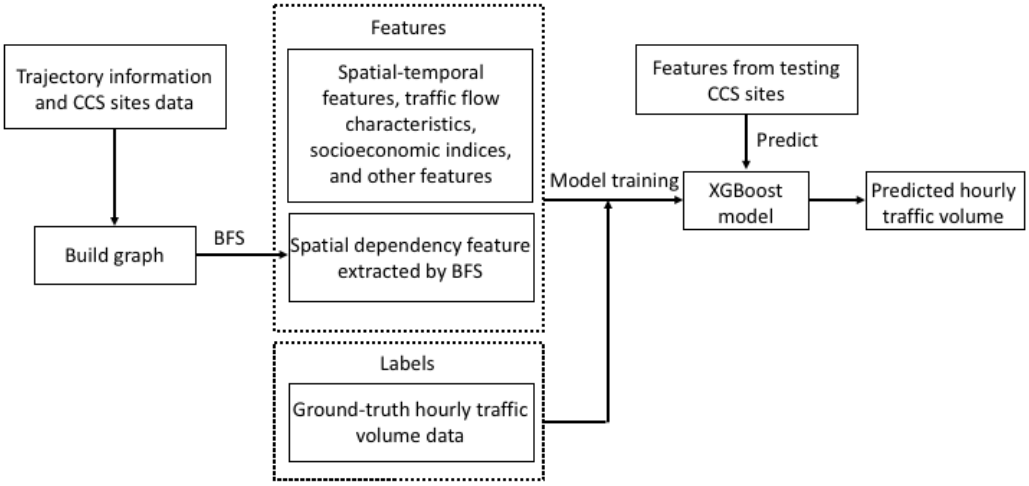


Figure 3.1 Methodological framework for hourly traffic volume prediction

3.2.1 Graph Representation

In order to explore spatial correlations among CCS sites, traffic network is abstracted as a graph first, and then BFS is implemented to find highly correlated neighboring CCS sites. In general, a weighted graph consists of a set of nodes, a set of edges, and weights associated with the edges. Formally, given a graph G, it is defined as:

$$\mathrm{G} = (V, E, w) \tag{7}$$

where V represents a set of vertices; E represents a set of edges, and w denotes the weights assigned to the edges.

Based on the construct of traffic network and purpose of the study, a node in the graph represents a CCS site, and an edge indicates the existence of spatial connection between two CCS sites. Adjacency matrix M is used to measure the intensity of those connections (i.e., the weight). Figure 3.2 illustrates an example of adjacency matrix. In this matrix, the elements in row i and column j represent the correlation intensity between CCS sites indexed by i and j. Probe trajectory data are used to measure connections between nodes. Specifically, an edge exists between node i and node j if there is more than one trajectory passing through i to j directly:

$$E_{i,j} = \begin{cases} 0 & if\ T_{i,j} = 0 \\ 1 & otherwise \end{cases} \tag{8}$$

where $T_{i,j}$ denotes the number of probe trajectories between node *i* and *j*. Then, the correlation intensity between node *i* and *j* is defined as follows:

$$M_{i,j} = \frac{T_{i,j}}{T_{max}} \tag{9}$$

where T_{max} represents the maximum number of probe trajectories across all edges in this entire traffic network. If $M_{i,j}$ is 0, it simply denotes that no edge exists between the two nodes. Since CCS captures the total traffic counts (non-directional), it is difficult to delineate the traffic volume on each direction. Thus, the proposed graph is a weighted undirected graph. Consequently, the corresponding adjacency matrix is symmetric (i.e., $M_{i,j} = M_{j,i}$).

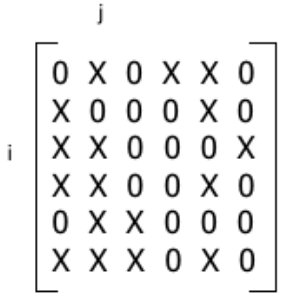


Figure 3.2 A 6 by 6 adjacency matrix of a graph

3.2.2 Breadth First Search (BFS)

BFS is a graph-based algorithm for traversing or searching purposes. It is commonly applied to examine connectivity or compute the shortest path from a single source node to other nodes of unweighted graphs (Kurant, Markopoulou, & Thiran, 2010). BFS is capable of identifying neighboring nodes within a certain depth. Specifically, given a source node, BFS first detects all neighboring nodes that are directly connected with the source node, marks them as visited nodes, and defines them as nodes in layer 1 (depth=1). In a similar fashion, each vertex in layer 1 checks all of the directly connected nodes to see if they are unvisited. Those unvisited nodes will be marked as visited and labeled as nodes in layer 2 (depth=2). This process repeats iteratively until all connected nodes are visited or the depth of layer reaches the maximum allowed depth.

One important feature of BFS is that, during each iteration, it will only visit all vertices at the same depth before visiting any at further depths. This enables finding neighboring nodes of source node by accumulating visited nodes at each layer. It is noted that the depth of layer denotes the minimum number of edges to the source node, instead of the geographical distance between nodes.

3.2.3 Spatial Correlation Calculation

After extracting all neighboring CCS sites within a given depth by BFS, the spatial dependency feature is created by aggregating the hourly traffic volumes from neighboring CCS sites. In fact, not all CCS sites captured by BFS algorithm can be utilized in the calculation. In this study, CCS sites are divided into training set, validation set, and test sets, where training set is used for model training, validation set for model calibration, and testing set for evaluating model performance. As a result, only information from CCS sites in training set can be extracted by BFS and be deemed known. Besides, traffic volumes vary greatly across road functional classifications (Malenkovska Todorova, Donceva, & Bunevska, 2009). Therefore, including CCS sites with different functional classifications may largely deviate from the ground truth values. To this end, neighboring CCS sites that are neither in training set nor in the same

functional classification as the source node are excluded. Then, hourly volumes from qualified CCS sites are averaged by corresponding correlation weights. Specifically, the calculation of neighboring hourly volumes for a source node *s* (i.e., prediction site) at a specific time *t* is defined as follows:

$$v_{s,t} = \frac{\sum_i^N w_{s,i} * V_{i,t}}{\sum_i^N w_{s,i}} \tag{10}$$

where *N* is the total number of qualified neighboring CCS sites; $v_{i,t}$ is the hourly volume of CCS *i* at time *t*; $w_{s,i}$ is the correlation weight from source node *s* to the node *i*. $w_{s,i}$ is calculated as:

$$w_{s,i} = \min\left(M_{s,j_1}, M_{j_1,j_2}, \dots, M_{j_k,j_{k+1}}, \dots, M_{j_{k-1},j_k}, M_{j_k,i}\right) \tag{11}$$

where j_k and j_{k+1} are two consecutive intermediate nodes along the path *s* to *i*. The correlation weight calculation borrows from the concept of capacity constraints in the maximum flow problem (Yuan, Bae, & Tai, 2010), where the total amount of traffic flow on a single path is dictated by capacity of the edge with minimum capacity along that path. Similarly, it makes sense that the spatial correlation between two CCS sites connected via a path is constrained by the edge with minimum weight along that path. It is worth mentioning that in case there is no qualified CCS site within the maximum allowed depth, the hourly volume information from the same functional classification with the closest Euclidean distance will be assigned to $v_{s,t}$ directly.

3.2.4 XGBoost Model

XGBoost is a highly effective tree boosting system that achieves state-of-the-art results on many challenging problems. For example, among the 29 challenge winning solutions published in Kaggle competition during 2015, 17 solutions used XGBoost (Chen & Guestrin, 2016). The tree boosting model forms a stronger learner by combining weak base learning models. XGBoost uses CART as the base learner. It is developed from GBDT, a classic boosting tree proposed by Friedman (2002). For a given dataset with n samples and m features $D = \{(x_i, y_i)\}$ ($|D| = n, x_i \in \mathbb{R}^m,\ y_i \in \mathbb{R}$), GBDT predicts the output $\hat{y}_i$ using *K* additive functions:

$$\hat{y}_i = \Phi(x_i) = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}, \quad (for\ i = 1, \dots, n) \tag{12}$$

where $\mathcal{F} = \{f(X) = w_{q(X)}\}$ ($q: \mathbb{R}^m \to T, w \in \mathbb{R}^T$) is the space of CART, q represents the structure of each CART that maps an example to the corresponding leaf index, and T is the number of leaves in the tree. Each f_k corresponds to an independent tree q and leaf weight w. For prediction, each tree contains a continuous score on each of the leaf, where the score on i^{th} leaf is denoted by α_i. GBDT updates the new tree by minimizing a specified loss function. However, traditional optimization method in Euclidean space cannot be used to optimize parameters of a new tree. Instead, GBDT greedily builds the model in a forward stagewise fashion (J. Friedman, 2001):

$$\hat{y}_i^t = \hat{y}_i^{t-1} + f_k(x_i) \tag{13}$$

where $\hat{y}_i^{t-1}$ is the sum of predictions on previous $t-1$ optimized trees. Mean square error function is applied as the loss function of GBDT for regression problem (Friedman, 2001), which is defined as:

$$L(y, \hat{y}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{14}$$

The final prediction is derived by summing up the scores in the corresponding leaves in the decision trees.

As an improvement, XGBoost adds a regularization term to the loss function to help smooth the final weights α to avoid overfitting. The objective function for XGBoost can be written as:

$$Obj(y, \hat{y}) = L(y, \hat{y}) + \sum_{k=1}^{K} \Omega(f_k) \tag{15}$$

$$\text{where } \Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \alpha_j^2 \tag{16}$$

where γ and λ are hyperparameters to penalize model's complexity. XGBoost applies second-order Taylor expansion (GBDT uses first-order) to quickly optimize the objective function as follows:

$$Obj^{(t)}(y, \hat{y}) \simeq \sum_{i=1}^{n} [L(y_i, \hat{y}^{t-1}) + g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i)] + \Omega(f_k) \tag{17}$$

where g_i and h_i are the first and second gradient statistics of loss function, and $L(y_i, \hat{y}^{t-1})$ is a constant term that can be removed. Because each sample only corresponds to one leaf node in a decision tree, the loss function can be reformulated as the sum of loss for each leaf node:

$$Obj^{(t)}(y, \hat{y}) \simeq \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i)\alpha_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\alpha_j^2 \right] + \gamma T \tag{18}$$

where I_j is the sample set of leaf j. Eventually, the optimal weight α_j^* of leaf j can be derived by minimizing the objective function:

$$\alpha_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{19}$$

and consequently, the corresponding optimal value of objective function is:

$$\widetilde{Obj}^{*(t)}(y, \hat{y}) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{20}$$

The optimal weight α_j^* can be calculated once a tree structure q is fixed. CART is a binary tree model, which grows recursively by splitting from a father node to left and right children nodes using one of the features. In fact, it is intractable to enumerate all possible tree structures. A greedy algorithm is therefore applied to find the optimal split point with the largest loss reduction for each split:

$$L_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{21}$$

where I, I_L, and I_R are samples of father node, left children, and right children, respectively. A tree structure will be determined once L_{split} does not improve significantly.

In order to achieve scalability, XGBoost adopts a series of strategies to accelerate the training process, such as the greedy approach for feature split and a column block technique. Hourly volume prediction involves a large number of features, making XGBoost well suited for handling the problem efficiently. In addition, tree-based models are capable of interpreting feature importance (Tuv, Borisov, & Torkkola, 2006). In CART, every node split uses a single feature with the largest loss function (Equation 21). One can compute the loss reduction accordingly, and rank features according to the average loss reductions across all trees in the model (also called mean decrease impurity). The larger the average loss reduction, the more important this feature is. This method enables us to identify critical components in affecting traffic volume variation.

3.3 Performance Assessment

Model performance is mainly evaluated based on prediction accuracy. In this study, *coefficient of determination* (R^2), MAE, and MAPE are used for performance assessment. Mathematical formulations and brief descriptions are shown in Table 3.1.

Table 3.1 Measurements of model performance for hourly volume prediction

Name of measurement	Mathematical formulation	Brief description
R^2	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	R^2 is the proportion of traffic volume variance that is explained by predicting models, and it provides a measure of how well observed outcomes are replicated by the model.
MAE	$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i \rvert$	MAE is a measure of difference between actual values and predicted values, which gives a clear interpretation of average magnitude of the errors for predictions.
MAPE	$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left\lvert \frac{y_i - \hat{y}_i}{y_i} \right\rvert$	MAPE is a statistical measure of prediction accuracy, where the prediction error is presented as a percentage. Smaller values indicate better prediction power.

In Table 3.1, y_i is actual traffic volume of the i^{th} sample, $\hat{y}_i$ is estimated hourly traffic volume of the i^{th} sample, $\bar{y}$ is sample average, and n is the number of total sample size. Note that R^2 can vary from $-\infty$ to 1, and values closer to 1 represent better predictive capability. Conversely, MAE and MAPE closer to 0 indicate better model performance.

4. DATA DESCRIPTION

Here we introduce the datasets used in this study. Probe trajectory data are exclusively utilized to construct the traffic network graph for the graph-theory based method, and the rest datasets are used for model building.

4.1 Probe Trajectory Data

To create the traffic network graph, probe vehicle data provided by INRIX are utilized in this study (*Inrix Data*, 2019). Probe data refers to the information extracted from a portion of a vehicle stream using probe sensors (e.g., cell phone and automated vehicle location [AVL]). The raw data include 2.5 million trajectory records with 130 million GPS points in the state of Utah during September 2018. A trajectory is defined as a trace consisting of a set of GPS points from origin to the destination of a trip. Each GPS point contains basic information of a unique trajectory ID, time stamp, and geographical coordinates. The median sampling rate is 50 seconds.

Using this dataset, we will be able to capture the number of trajectories traversing a roadway segment equipped with CCS. In order to do that, the trajectory data need to be preprocessed. Considering that GPS points may deviate from the actual locations due to measurement errors inherent to GPS technology, and trajectories with low frequency of sampling rate are difficult to capture, a computationally intense map matching technique is carried out with the OpenStreetMap tool, which applies a hidden Markov model to reconstruct the most likely road-based route from a time-stamped sequence of latitude/longitude pairs. As a result, the raw GPS points are mapped on the nearest streets, and the number of GPS points grows from 130 million to 970 million by interpolating artificial points along the trajectories. However, due to signal loss and other GPS errors, a portion of trajectories are either broken into two parts, or represented by sparse GPS points far apart from each other even after interpolation. These anomalous trajectories are further removed due to quality control. Note that anomalous trajectory in this study is defined as the trajectory that has missing values, or has two consecutive GPS points that either the time span is beyond 10 minutes, or the Euclidean distance is longer than 2.5 km. Finally, 1.5 million trajectories are obtained for analysis.

4.2 Inputs and Outputs for Prediction

Prediction is a subcategory of the supervised ML problem. Supervised ML uses algorithms to learn the mapping function from input to output (Alpaydin, 2009). In supervised learning, each sample in the dataset consists of input variables (typically a feature vector) and an output variable. The mapping function trains the algorithm so that it can use new input variables to predict the output variable with good accuracy. In this study, the output consists of ground truth hourly traffic counts (*a*), which are used to train and evaluate model performance. Apart from our proposed spatial dependency feature, input features such as spatial, temporal, and traffic flow characteristics are subcategorized into six parts (*b*-*g*). Data for each subcategory are briefly explained in the following.

(a) CCS data

Throughout the year, CCS records continuous counts of traffic on road segments, which can be translated into hourly volume. In this study, we use historical hourly traffic counts from 94 CCSs from May to December 2017 provided by the Utah Department of Transportation (UDOT). These CCSs are located in different geographical regions in the state of Utah with records ranging from 0 to 12,000 veh/hr.

(b) Probe data

Other than the trajectory data, iPeMS, an online database owned by the UDOT, provides access to aggregated probe data collected by HERE in the state of Utah (*HERE Probe Data*, 2019). Measured average speed and estimated free-flow speed from May to December 2017 are retrieved from HERE (all data are collected every five minutes and aggregated by the hour).

(c) Temporal features

To delineate temporal variation of hourly volumes, hour-of-day (1:00 to 24:00), time-of-day (morning vs. midday vs. afternoon vs. evening vs. midnight), day-of-week (Monday to Sunday), and season (spring to winter) are added as temporal features. Meanwhile, federal holidays during the observed period (Memorial Day, Independence Day, Labor Day, Columbus Day, Veterans Day, Thanksgiving Day, and Christmas Day) are also accounted for.

(d) Infrastructure characteristics

UDOT provides infrastructure information for each CCS site. Variables, including road functional classification (interstate, freeway, major/minor arterial), segment speed limit, number of through lanes and number of high-occupancy vehicle (HOV) lanes, are extracted.

(e) Surrounding road network

We hypothesize that the complexity of a surrounding road network may affect the traffic volume of a segment. As a result, the length of roads, classified by functional classification, in a one-mile radius of a CCS site is captured using ArcGIS. Specifically, these features are the total length of interstate, expressway, principal arterial, minor arterial, major collector, minor collector, and local streets, respectively.

(f) Weather

MesoWest, a program started at the University of Utah, provides current and historically archived weather observation (*MesoWest*, 2019). Weather information recorded every five minutes is retrieved from 24 weather stations across Utah and then aggregated by hour. Each CCS site is associated with the weather information of its closest weather station. The information contains weather conditions (clear, cloudy, rainy, snowy and others), distance to the closest weather station, air temperature, dew point temperature, wind speed, and range of visibility.

(g) Socioeconomic factors

UDOT also provides socioeconomic features, which contain average income, employment density, school enrollment rate, number of households, household size, and population density. All socioeconomic features are considered within a three-mile radius of each CCS site. Those features are used to represent potential socioeconomic impacts on traffic flow.

In sum, continuous historical hourly volumes of 94 CCS sites were collected from May 17 to December 31, 2017. However, due to malfunction and/or other unknown reasons, some CCSs failed to record the complete data across the entire period, where the missing values accounted for 22% of the total observations. Feature values associated with missing labels are therefore removed. Among the aforementioned 33 features (including spatial dependency feature) and categorical variables (i.e., hour-of-day, time-of-day, day-of-week, season, and weather conditions) are converted into dummy variables using One-Hot Encoding. Finally, our dataset contains 384,879 data points with 73 features.

5. RESULTS AND ANALYSIS

5.1 Overview

In this section, the proposed ML approaches are implemented for hourly volume prediction. This analysis explores models' performance in response to the size of input features. A computer with an Intel i5 8400 processor clocked at 2.81 GHz is used to conduct numerical analysis. In this study, model training is implemented in Python 3.7 and R. Open source packages Scikit-learn and XGBoost API are used to train SVM and XGBoost, separately.

5.2 SVM Modeling Result

In this subsection, both linear SVM and non-linear SVM are performed on the dataset. Model calibration and prediction are conducted subsequently, followed by the result analysis and comparison.

5.2.1 Model Calibration

Linear SVM is capable of training datasets with faster speed than non-linear SVM. In linear SVM, parameter *C* needs to be adjusted to optimize model performance, where *C* is the regularization parameter controlling the trade-off between low training error and low testing error. The values of parameter *C* are selected as [0.001, 0.01, 0.05, 0.1], respectively, to test on the training set, and prediction accuracy for the linear SVM is displayed in Figure 5.1.

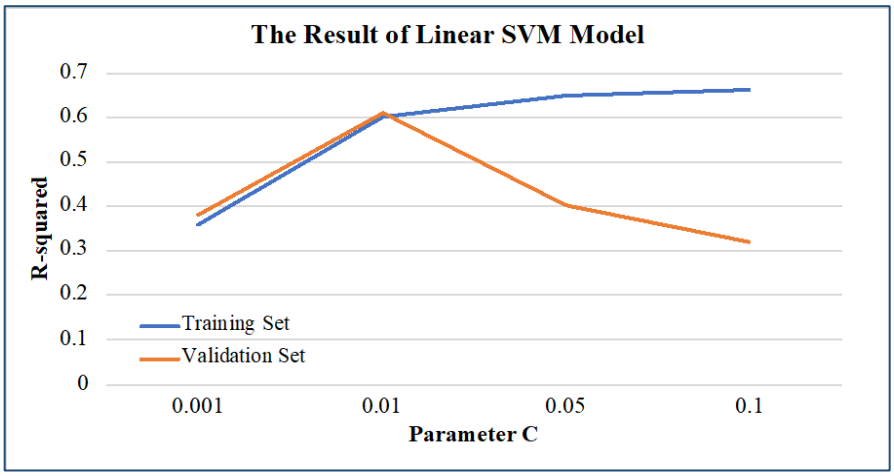
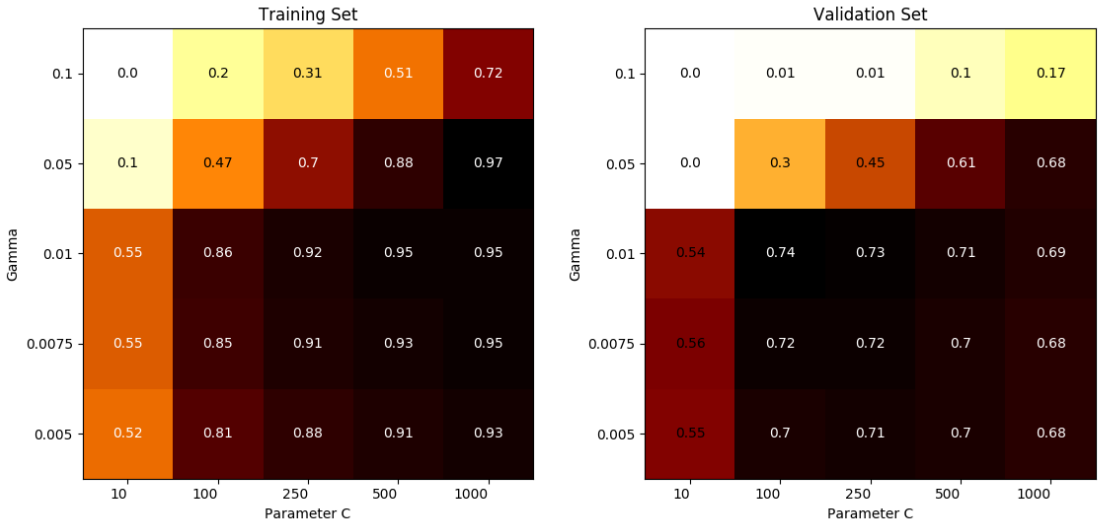


Figure 5.1 Model calibration result from linear SVM model

From Figure 5.1, it is observed that the highest prediction accuracy on validation set only reaches 0.61 when C is set as 0.01. As the value of C continues increasing, prediction accuracy on validation set starts to decrease, which indicates overfitting.

Meanwhile, parameters in non-linear SVM are calibrated. As mentioned earlier, non-linear SVM is a powerful ML technique due to kernel tricks. However, the model's time complexity is $O(n^3)$, where n is the sample size. This implies that non-linear SVM can be computationally expensive when the sample size is large. To avoid excessive training time, only 30,000 samples are randomly chosen from the training set. In non-linear SVM model, two parameters *C* and σ need to be adjusted, where *C* is the same parameter as in linear SVM and σ denotes the parameter from Gaussian kernel equation. Grid search method is used to adjust the two parameters simultaneously. For parameter *C* and σ, five numbers (i.e.,

C ∈ [10, 100, 250, 500, 1000], and σ∈ [0.005, 0.0075, 0.01, 0.05, 0.1]) are chosen, respectively, to perform in CV. Figure 5.2 shows the prediction accuracy (R^2) on training set and validation set.



(a) (b)

Figure 5.2 The predicting accuracy from the grid search method on training set (a) and validation set (b)

In Figure 5.2(a), the prediction accuracy will increase as *C* gets larger and σ gets smaller. However, corresponding to the same values of *C* and σ in training set, the prediction accuracy in validation set will increase first and then start to drop as shown in Figure 5.2(b). High prediction accuracy on training set and low prediction accuracy on validation set denotes overfitting. Consequently, *C*=100 and σ=0.01 are chosen as the optimized parameters due to the highest prediction accuracy on validation set (0.74).

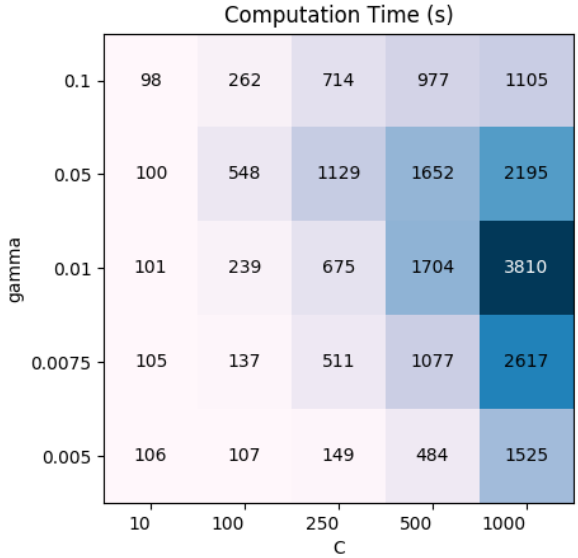


Figure 5.3 The computation time for each trial from the grid search method

Figure 5.3 shows the model's computation time for each trial using grid search method. When training with optimized parameters, it takes approximately 239 seconds to form the model. Nevertheless, as the

model tends to overfit the training data, the training process of non-linear SVM model becomes more sophisticated, which results in longer training time. It is noted that the longest training process in grid search takes 3,810 seconds.

5.2.2 Prediction Accuracy on Test Dataset

Once parameters for SVM are calibrated, two models are performed on the test, and R^2 is used to measure the models' prediction effectiveness. The results show that linear SVM achieves an R^2 of 0.52, while R^2 on non-linear SVM is 0.76. This indicates that non-linear SVM outperforms linear-SVM despite the fact that non-linear SVM is trained with fewer samples.

In the next step, to explore the difference between predicted and ground-truth values explicitly, two CCS sites (i.e., CCS 351-negative direction and CCS 416-positive direction) are selected to visualize the prediction results from non-linear SVM in a continuous 14-day period. The results are illustrated in Figure 5.4.

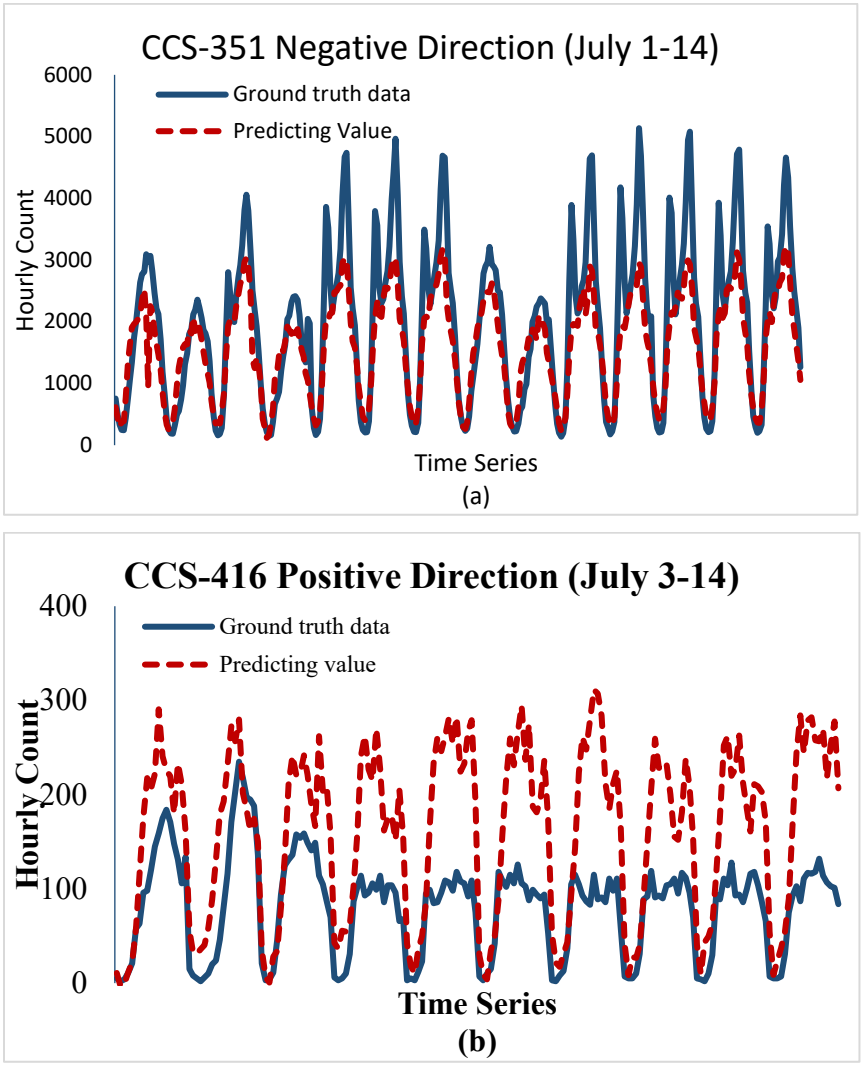


Figure 5.4 A portion of ground-truth data and predicting result for (a) CCS 351 (negative direction); and (b) CCS 416 (positive direction)

In Figure 5.4, it is found that the CCS 351 and CCS 416 sites show different levels of traffic flow and varying traffic patterns. For the CCS 351 site, hourly traffic volume ranges from 0 to 5,000 veh/h. In contrast, for the site of CCS 416, hourly volume is below 250 veh/h in most times of day. Note that in general, the prediction curve fits well when the traffic volumes are relatively low. Yet, prediction error increases when the traffic volume suddenly spikes. It is also noticed that non-linear SVM underestimates hourly volumes during peak hours for the CCS 351 site; whereas, the model overestimates traffic counts during peak hours for the CCS 416 site.

5.2.3 AADT Estimation

In this subsection, AADT from roads in the test dataset is calculated via ground-truth data and estimated values from non-linear SVM. Calculating AADT requires continuous daily traffic count over a year. However, the dataset used for this project only spans from May to December 2017. To address this issue, two methods for AADT estimation, namely simple average (Figliozzi et al., 2014) and factoring method (Gadda et al., 2007), are implemented to produce AADT. Intuitively, simple average method uses the average daily volume as the estimation of AADT, which is calculated as:

$$AADT_{estimated} = \sum_{t=1}^{N} Vol_{obervation\_t} \tag{22}$$

where $Vol_{obervation\_t}$ is the daily traffic volume at day *t* either from ground-truth data or predicting values, and *N* is the number of days in observation period. Meanwhile, the calculation of factoring method is expressed as follows:

$$AADT_{estimated} = \sum_{t=1}^{P} Vol_{obervation\_t} * M_i * DOW_{i\_t} \tag{23}$$

where M_i is the monthly factor for the functional classification group *i*; and DOW_i is the day-of-week factor for the functional classification group *i*; and *P* is the number of continuous testing days. Empirically, P is chosen as 3 for prediction purpose. The calculated results for M_i and DOW_i are appended in Appendix A. Figure 5.5 shows the predicted AADT values for CCS sites in the test dataset.

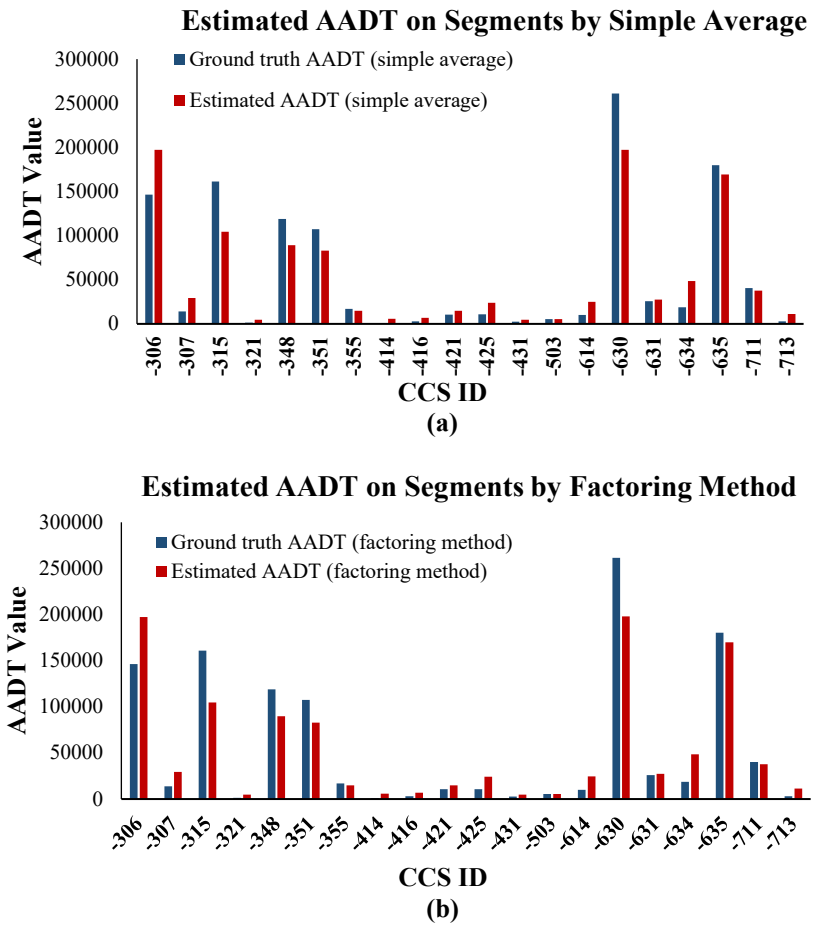


Figure 5.5 Predicted AADT from (a) simple average method; and (b) factoring method

Figure 5.5 shows that the estimated AADT from simple average and factoring method for each site are very close. R^2 values from the proposed two methods are 0.883 and 0.884, respectively. The result indicates that SVM is able to achieve satisfactory prediction accuracy for AADT prediction, although the performance on hourly volume prediction is relatively low.

5.3 Graph Theory Based Spatial Correlation Modeling Result

5.3.1 Graph construction and visualization

As mentioned in the Methodology section, the proposed traffic network is an undirected weighted graph, where V denotes CCS sites, E represents edges for those CCS sites, and w represents the spatial correlation intensity of those edges. An edge exists between two CCS sites only if the two sites are "connected." The probe vehicle trajectory data are utilized to calculate such connectivity and their intensity in this paper. Although these 94 CCS sites are widely distributed in Utah, not all trajectories pass through those sites. As a result, trajectories that do not traverse any CCS sites are filtered out using ArcGIS.

In the next step, the CCS locations that each trajectory traverses are extracted and sequenced by time. Trajectories that pass only one CCS are further removed since an edge requires at least two nodes. A sample of trajectory profiles is shown in Table 5.1.

Table 5.1 The sequence of traversing CCS sites for a sample of trajectories

Encoded unique trajectory ID	[Timestamp, CCS ID]
8cad1f9b0a90151761822b2b9037ff76	[[1535825719, -402]---> [1535826252, -621]---> [1535826644, -400]]
9c3a8b03acfea8474f9e8f24a9d47596	[[1536435629, -402]---> [1536436037, -621]---> [1536436417, -400]]
81f9fcea8615fa175ec2191c71ce565f	[[1537414744, -402]---> [1537415262, -621]]
8207bb6bb6ea69b84dfeed1c3ee9c792	[[1537396287, -411]---> [1537397073, -412]---> [1537401267, -402]---> [1537401695, -621]--->[1537402034, -400]]

Note that an edge exists between any two CCS sites if there is at least one trajectory record. The total number of trajectories for each edge is subsequently counted. For example, the first trajectory in Table 2 traverses CCS sites 402, 621, and 400 sequentially. Correspondingly, there is an edge between CCS sites 402 and 621, and another edge between CCS sites 621 and 400. Meanwhile, three out of four trajectories in the table traverse between CCS sites 621 and 400. Correspondingly, the total count of trajectories for this edge is three. In this study, 686 edges are formed by approximately 1.5 million trajectories among these 94 CCS sites. The edge with the maximum number of trajectories has a count of 44,160 trajectories. The number of trajectories on each edge is divided by the maximum number of trajectories (44,160) to represent the weight *w* of that edge (ranging from 0 to 1). The distribution of edges by the number of trajectories is shown in Figure 5.6.

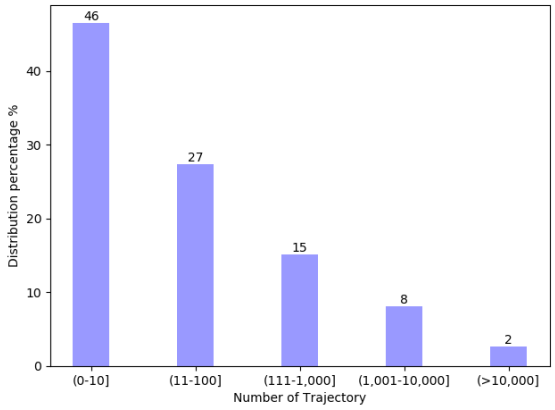


Figure 5.6 Distribution of edges by the number of trajectories

Figure 5.6 shows an uneven distribution of the weighted edges, with approximately half of the edges having a number of trajectories less than 10. Edges with a trajectory count over 1,000 only account for 10% of the total edges. Note that although some edges demonstrate high connectivity, they are geographically distant from each other. When computing the spatial dependency feature introduced in the Methodology section, it is calculated by weighting hourly traffic volume at neighboring CCS sites (same BFS depth) at the same time of day. If two CCS sites are too far away, it would be infeasible to justify their spatial correlation. The graph is thus pruned by cutting edges whose Euclidean distance is larger than 80 kilometers – an empirical value set based on a one-hour driving distance on highway. The modified graph eventually contains 328 edges.

CCSs are distributed on road segments with varying functional classifications. The functional classifications are labeled from 1 to 4, representing interstate, freeway, principal arterial, and minor arterial, respectively. Figure 4(a) shows the geographical distribution of CCS sites categorized by the functional classification of roads they reside in.

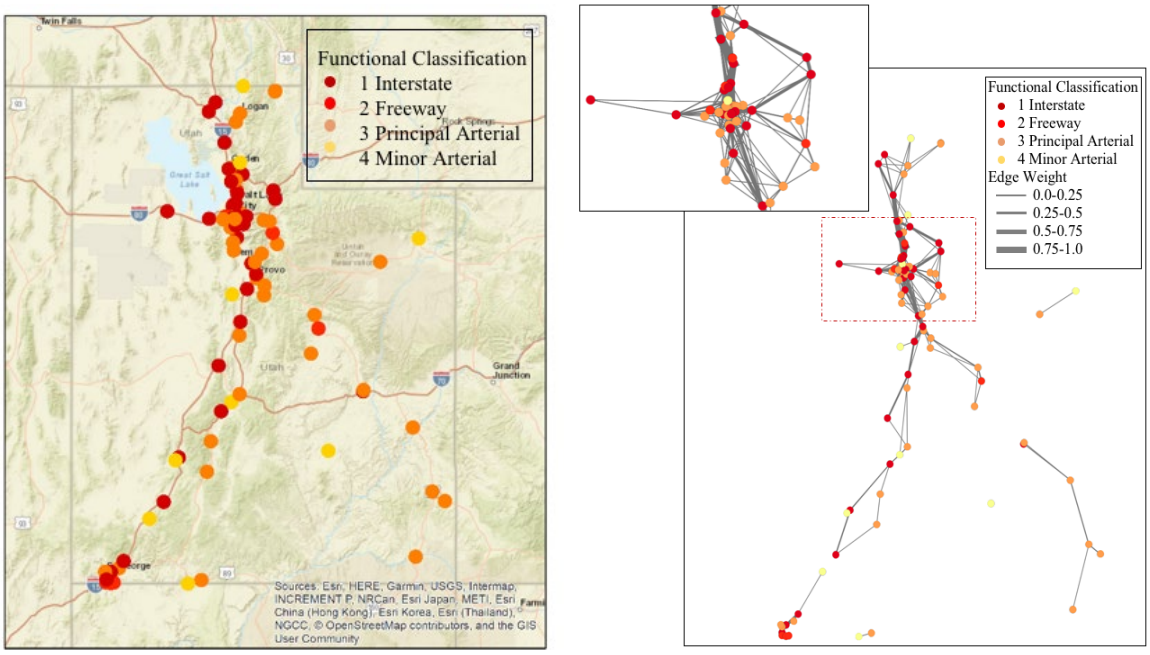


Figure 5.7 (a) The geographical distribution of CCS sites; and (b) the constructed traffic network graph in this study

In Figure 5.7(a), note that a significant portion of CCSs are congregated in urbanized areas (especially the Salt Lake City region), and the rest are sporadically distributed in rural areas. The distribution of CCSs for each functional classification is relatively uneven, where the numbers of CCSs on interstates and principal arterials outweigh the numbers on freeways and minor arterials. Specifically, the number of CCSs on road segments from Class 1 to Class 4 are 34, 8, 42, 10, separately. Figure 5.7(b) shows the weighted traffic network graph, where thickness of the line represents spatial correlation intensity of the edge. Note that edges between CCSs located along the interstates tend to have higher weights. This can be explained by the fact that interstates carry more traffic (therefore more trajectories) within the state to serve people's mobility. This further demonstrates the advantage of graph-based metric over Euclidean distance measure, which previous studies used, when trying to unveil the spatial dependencies of a road network. Everything else being equal, two CCS sites with the same distance apart located on different road segments (e.g., interstate vs. arterial) show varying spatial correlations using the graph-based method.

After constructing the traffic network graph, spatial dependency feature V_{st} can be calculated by setting the maximum allowable depth for BFS to extract all qualified neighboring CCS sites. Together with this proposed feature, other features introduced in Section 4.2 will be fed into XGBoost for model training.

5.3.2 Hyperparameter Tuning

The hyperparameter tuning process is crucial for the performance of ML algorithms. Unfortunately, such tuning may require domain expertise, rules of thumb, and sometimes brute-force search (Snoek, Larochelle, & Adams, 2012). In general, random search, grid search, and Bayesian optimization are commonly used methods for hyperparameter tuning. Compared with the latter two, random search, by searching over the space of one hyperparameter at a time, ordered by the ones that have more influence over other hyperparameters first, is the easiest to implement, yet the accuracy is only marginally affected. As a result, random search is performed in this study.

To tune the hyperparameters, the entire dataset is split into three parts: training set, validation set, and test set. Given a specific range for each hyperparameter, training set is used to fit the model repetitively with different hyperparameters. Hyperparameters with the highest accuracy score on the validation sets are selected for the final model. Finally, a test set is utilized to report on the model's generalization performance. In this calibration process, R^2 is used to indicate the prediction accuracy. CCS sites are randomly split into three groups: 70% as training data, 15% as validation data, and 15% as test data, respectively. The main hyperparameters for our proposed model are listed below:

(1) For BFS method, the maximum allowed depth H is set to refrain the source node from stretching endlessly to other connected nodes;

(2) XGBoost has a set of parameters (e.g., tree depth, the ratio of column sampling, sub-sampling, and L2 regularization coefficient, etc.). Among them, the number of trees T and learning rate $\boldsymbol{\eta}$ (shrinkage) are the main hyperparameters that need to be adjusted. Other hyperparameters are set as default.

Specifically, for the proposed model, the maximum allowed depth H for BFS is tuned from 1 to 8. The number of trees T and learning rate η in XGBoost are adjusted empirically from 100 to 500, and 0.01 to 1, respectively. Calibration results for the proposed model are shown in Figure 5.8.

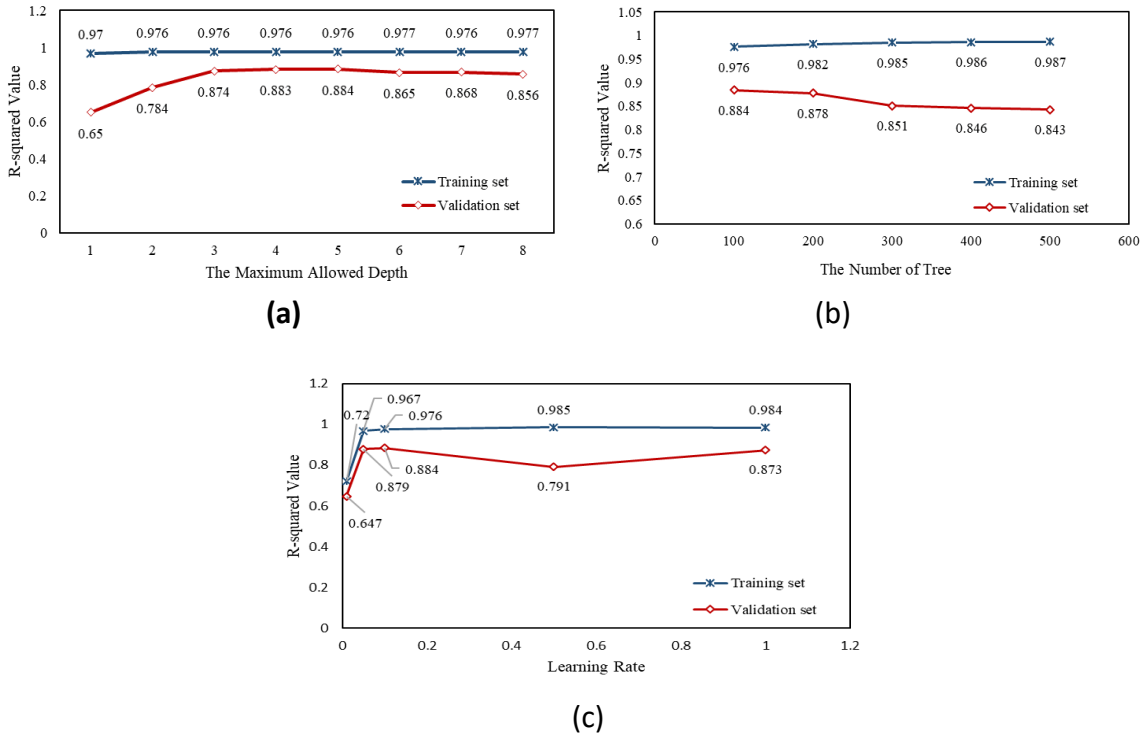


Figure 5.8 Calibration results of the proposed model, where (a) the maximum allowed depth for BFS; (b) the number of trees for XGBoost; and (c) learning rate are tuned separately

Figure 5.8(a) shows the effectiveness of BFS by different levels of depth. When the maximum allowed depth is set to 1, prediction on validation set is constrained by the limited number of neighboring CCS sites, thus causing large biased results. As the threshold of depth level increases, more neighboring CCS sites with the same functional classification are included, leading to improvement in prediction accuracy. However, when the maximum depth is larger than 5, prediction accuracy starts to decrease on validation set. Those CCS sites located far away from the source node may have the same level of traffic volume yet different traffic patterns. As a result, incorporating hourly volume information of those CCS sites may

introduce noise in the spatial correlation feature. Figures 5.8(b) and (c) indicate that the optimal number of trees T is 100, and corresponding learning rate η is 0.1.

For comparison purposes, three different sets of models are developed to benchmark our proposed one:

(a) XGBoost model without spatial dependency feature;
(b) XGBoost model with a spatial dependency feature characterized by Euclidean distance. Specifically, a ring buffer with radius of K km is created for each test site to average the hourly volumes of qualified CCSs within the buffer as spatial dependency feature;
(c) XGBoost model with a spatial dependency feature characterized by network distance[1]. Given a maximum allowed network distance D km, the shortest path algorithm is applied to calculate the spatial dependency value by searching qualified CCSs for each test site in the network and averaging their hourly volumes.

For model (a), only the number of trees T and learning rate η need to be tuned, where the optimal values are 200 and 0.1; for model (b), $K \in [10,20,30,40,50]$, and the optimal values for K, T, and η are 20, 100, and 0.1, respectively; and for model (c), $D \in [10,20,30,40,50]$, and the hyperparameters D, T, and η are optimized as 20, 150, and 0.1.

5.3.3 Model Performance and Comparison

Here, we present the numerical results from our proposed method (XGBoost+BFS) and other benchmarked models. To compare the models' effectiveness and test their generalization ability, the testing process is replicated five times by different random seeds. In each random seed, 15% of CCS sites are randomly selected and used as testing set. Prediction accuracy in regard to R^2, MAE, and MAPE on test set for each random seed is recorded in Table 5.2. The average training time across all random seeds for the proposed model is 54.6 seconds, which shows the strength of model's scalability.

Table 5.2 Prediction performance of different models

Seed	R^2 on test set				MAE on test set				MAPE on test set			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
1	0.849	0.780	0.860	0.895	643	720	619	548	0.80	0.56	0.65	0.57
2	0.839	0.862	0.811	0.934	645	554	562	404	1.06	0.93	0.51	0.88
3	0.817	0.875	0.879	0.885	887	773	799	733	1.26	0.57	0.68	0.52
4	0.859	0.866	0.847	0.868	631	397	420	522	1.96	1.03	1.58	1.33
5	0.898	0.831	0.936	0.916	347	681	311	308	2.09	1.66	1.07	1.67
Mean	0.852	0.842	0.866	0.900	631	625	542	503	1.43	0.95	0.90	0.79

M1: XGBoost model without spatial dependency feature
M2: XGBoost model with a spatial dependency feature characterized by Euclidean distance
M3: XGBoost model with a spatial dependency feature characterized by network distance
M4: XGBoost model with BFS algorithm

As observed from the results across five random seeds, the proposed approach achieves the highest R^2 and the lowest values of MAE and MAPE on average compared with other models. In contrast, M1 has the worst performance in terms of MAE and MAPE. This indicates that the spatial dependency feature does improve prediction accuracy of the model. M2 shows an inconsistent performance. In some random seeds, the predictions are exacerbated by the spatial dependency feature, which indicates that using Euclidean distance to capture the spatial correlation is coarse to some extent. M3 yields better predictions than M1 and M2, suggesting the effectiveness of quantifying the spatial dependency using network

[1] Network distance is defined as the length of shortest path between two locations in the network.

distance. Yet, M4 outperforms M3 with regard to R^2, MAE, and MAPE on average. Such results illustrate that the quantifying distance by depth in a weighted graph enables a better capturing of spatial correlation over Euclidean distance and network distance. R^2 values of all random seeds are bounded by narrow variations; whereas, MAEs across different random seeds vary significantly from 308 to 887 veh/hr. Such variation is due to the fact that hourly volume samples could vary widely across different locations from 0 to 12,000 veh/hr, and the functional classification of CCS sites in test set can be significantly different across each random seed scenario. The same reason applies to the variation of MAPE (from 0.51 to 2.09). For example, with the same prediction bias (e.g., 100 veh/hr), MAPE on CCS sites with large traffic volume is smaller than those with lower traffic volume. As a result, random seed scenario with a large proportion of low volume road segments in test set generally will achieve lower MAE and higher MAPE (e.g., Seed 5). To further demonstrate the variation of MAPEs with different levels of hourly traffic volumes, Figure 5.9 shows the violin plots of MAPEs on test set across all random seeds categorized by different levels of hourly traffic volumes.

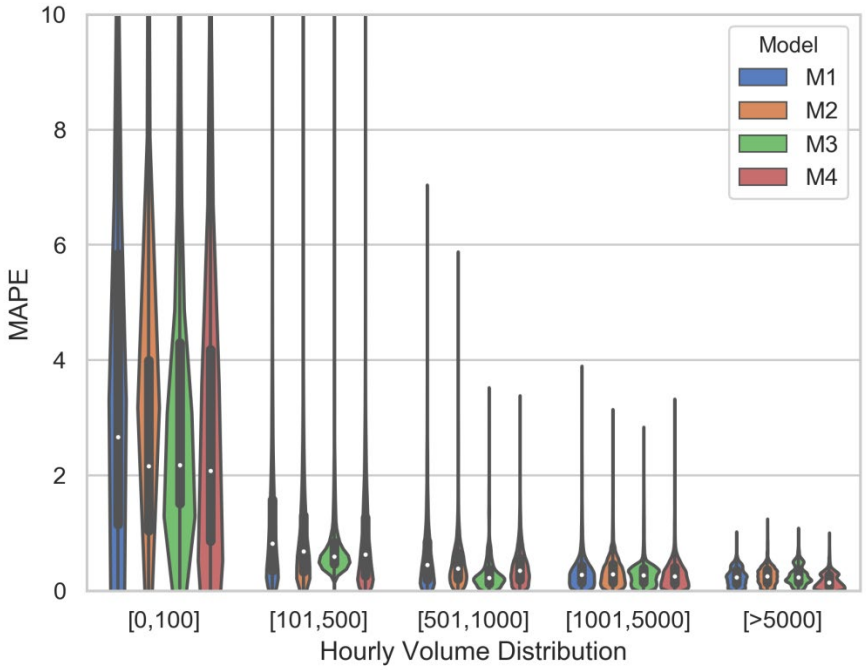


Figure 5.9 MAPE distribution of hourly volume prediction in test set across five random seeds classified by volume range

For each "violin" in the figure, a box plot is drawn inside, and its margin shows the Gaussian distribution of the dataset. In Figure 5.9, it is noted that predictions for lower volumes generally have higher MAPE value, especially when the actual hourly volume is below 100 veh/hr. This is because MAPE calculated with "low volume" scenarios are more sensitive to prediction errors as explained earlier. As the hourly volume increases, the median of MAPEs drops significantly. Meanwhile, models considering spatial dependency outperform the XGBoost model without the spatial dependency feature on all volume categories, indicating the advantage of utilizing the information of neighboring road segments. To better visualize model performance, ground truth hourly counts from test set in Seed 1 are compared against predicted values in Figure 5.10.

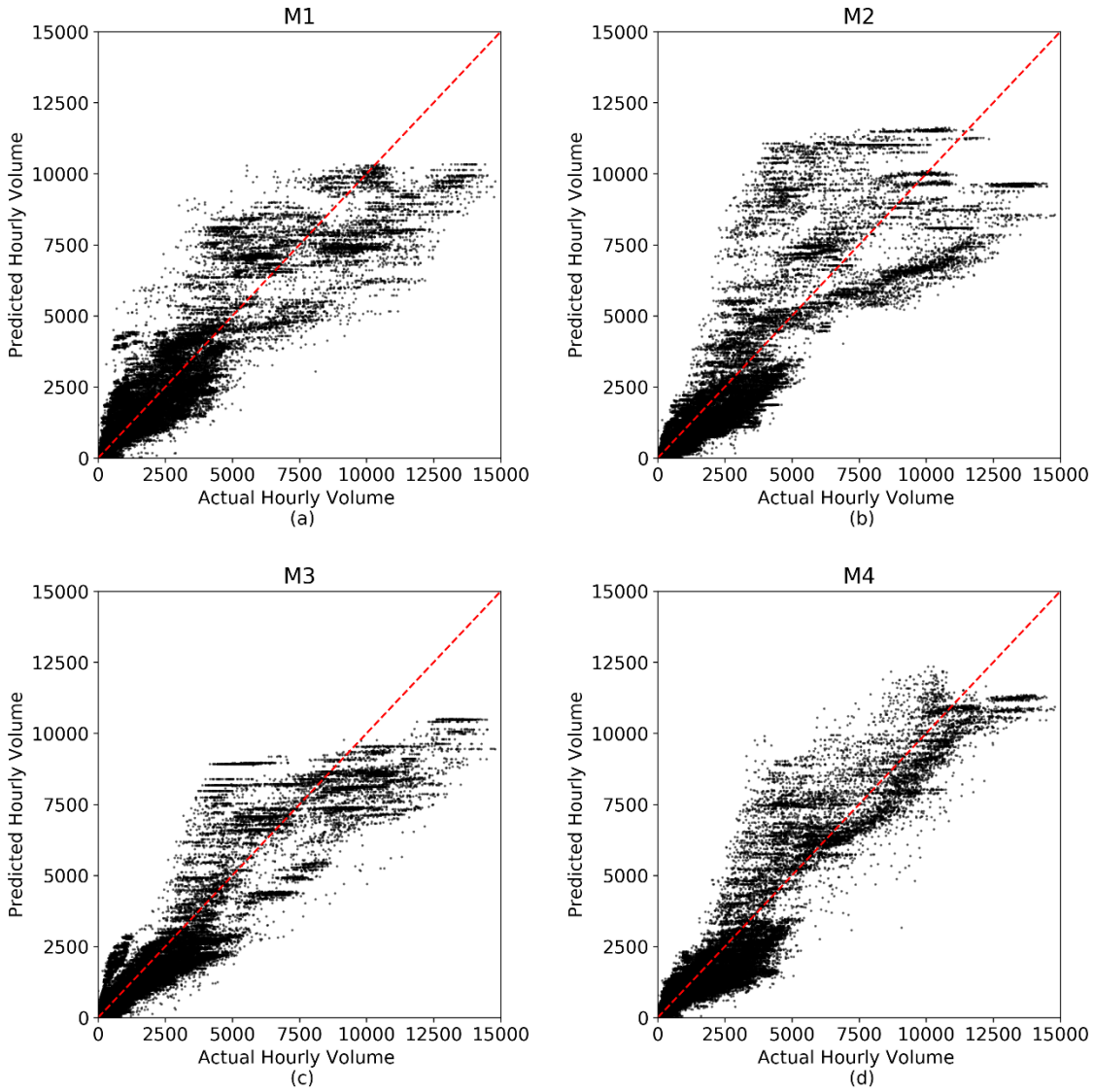


Figure 5.10 Comparison of ground-truth hourly volumes vs. predicted outputs from Seed 1 by (a) XGBoost model without spatial dependency feature; (b) XGBoost model with a spatial dependency feature characterized by Euclidean distance; (c) XGBoost model with a spatial dependency feature characterized by network distance; and (d) XGBoost model with BFS algorithm

As shown in Figure 5.10(d), results from the proposed model are the closest to the benchmarked diagonal overall, which indicates the best prediction performance among all. In contrast, a greater portion of data points tend to deviate further from the diagonal by only using the XGBoost model for prediction (Figure 5.10(a)). Such deviation is more pronounced when the actual hourly volume ranges between 5,000 and 10,000 veh/hr, which subsequently causes high MAEs and MAPEs. On the other hand, it is also observed that prediction is unexpectedly exacerbated using the Euclidean distance buffer to capture the spatial dependency feature (Figure 5.10(b)). One possible explanation is that the Euclidean distance ring buffer captures spatial correlation in a coarse level. For instance, in dense road network where nodes (CCSs) are distributed closely, two neighboring connected roads (spatially connected) might be irrelevant in terms of traffic pattern, yet Euclidean distance will deem them as spatially dependent given the close vicinity in distance. As a result, the aggregated neighboring volumes may incorrectly reflect the traffic condition at

prediction sites, inducing over- or under-estimation of traffic volumes. Note that using network distance can boost the prediction accuracy compared with the model without using spatial dependency feature (Figure 5.10(c) vs. Figure 5.10(a)). Yet, the improvement is less significant than the proposed method. Overall, the BFS approach decreases the average prediction errors steadily and effectively.

5.3.4 Spatial Correlation Analysis

Compared with the XGBoost model without spatial dependency feature, the proposed model obtains better performance based on all indicators. As mentioned earlier, one highlight of tree ensembles is their ability to interpret feature importance. Mean decrease impurity is implemented for each split in XGBoost during the training process. Thirty-three features are ranked based on their importance to the prediction. The feature importance ranking and feature category split are presented in Figure 5.11.

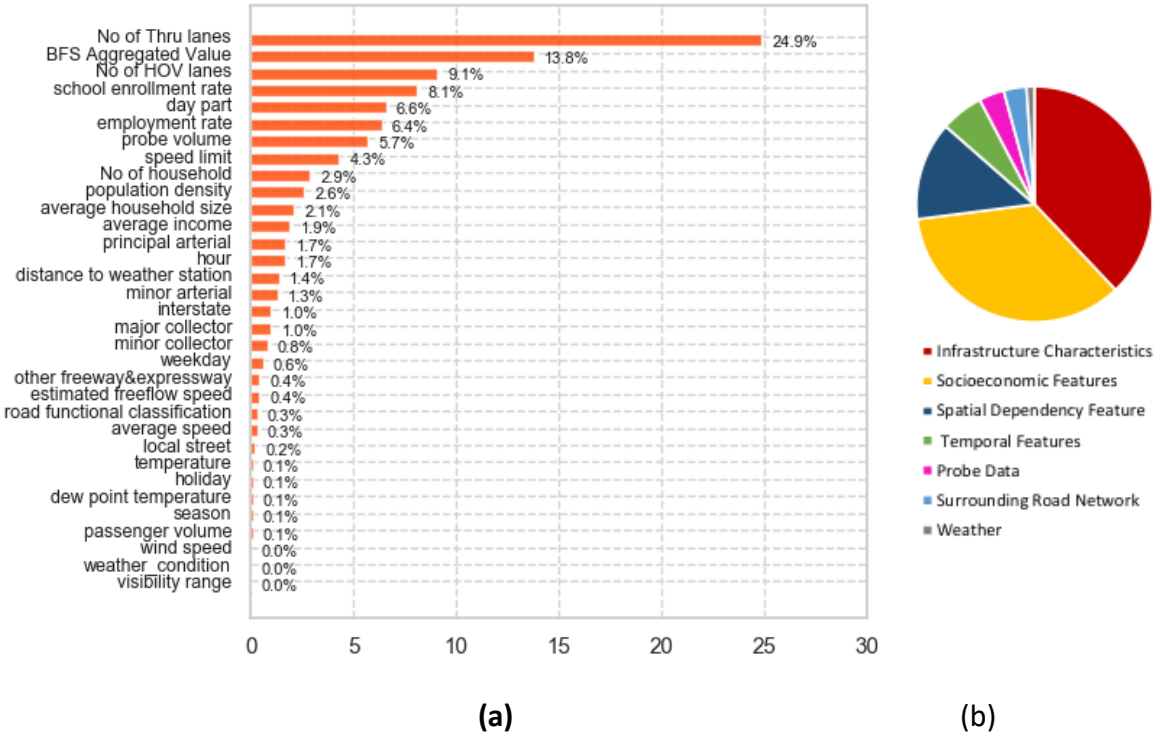
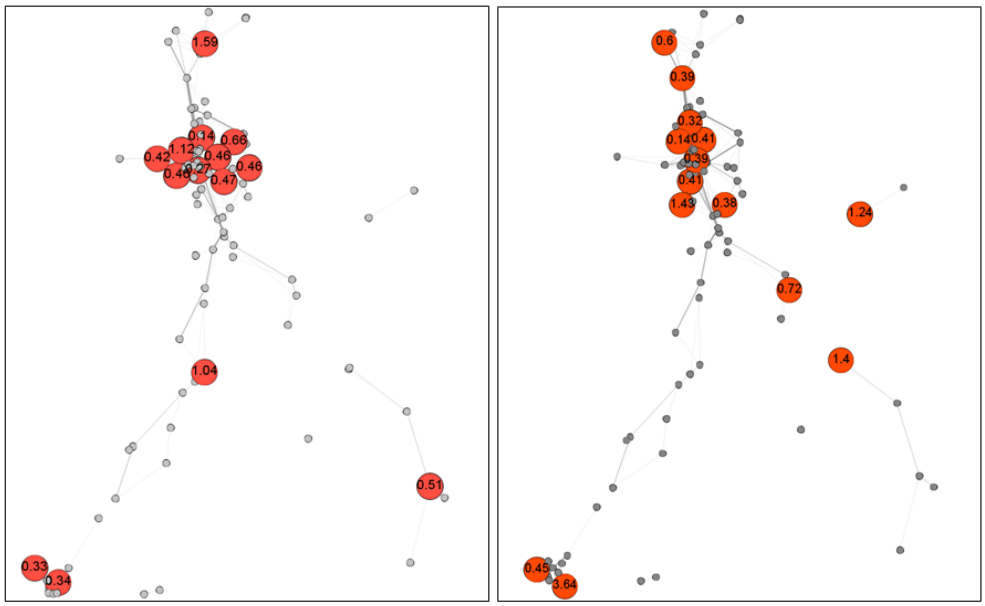


Figure 5.11 (a) Feature importance ranking; and (b) pie chart of importance split via mean decrease impurity by feature category, where the sum of importance coefficients equals to 1

In Figure 5.11(a), variables' contributions to prediction accuracy are highly imbalanced, with only 7 variables' importance coefficients exceeding 0.05. The spatial dependency feature ranks as the second most important feature for prediction with a coefficient of 0.138. This result highlights the fact that spatial dependency plays a critical role in estimating small-granularity traffic volumes. By effectively quantifying spatial correlation, prediction accuracy can be significantly improved. Figure 5.11(b) indicates that road characteristics, socioeconomic factors, spatial dependency, and temporal features are the most influential feature categories, which is consistent with the findings in Zhao and Chung (2001) (F. Zhao & Chung, 2001). Probe data and surrounding road network features, such as the total length of nearby interstates, also contribute to traffic volume prediction to a certain extent. However, weather factors only account for 0.01 of the total importance. This might be because the data collection period was mostly under normal weather conditions. Such feature importance analysis provides insightful guidance for future traffic volume prediction projects. Such techniques may ultimately provide transportation agencies or state DOTs with the tools to accurately estimate traffic volumes at relatively low cost. First,

given that infrastructure characteristics, socioeconomic features, and temporal features are significantly influential to traffic volume and can be readily available, those features should be regarded as essential data for estimating traffic volume. This prioritization can ensure a relatively good model performance while saving a huge amount of time on data collection and data filtering processes. Second, exploring spatial dependency can augment the prediction accuracy. Thus, we recommend utilizing additional features (such as trajectory data) to build the traffic network and quantify such spatial correlation. Lastly, probe data, surrounding road network information, and weather data are more variant and require extra time to collect and preprocess; whereas, they contribute relatively less to the prediction. If higher prediction accuracy is required, those features might be collected and fed into the model.

We further explore the scenarios where the spatial dependency feature demonstrates prediction superiority and reveals underlying reasons. Specifically, MAPEs for CCSs in testing dataset of each random seed are displayed in Figure 5.12. It is observed that, in Seeds 1 and 3, there are more testing CCS sites congregating around the center of the Salt Lake City cluster compared with other scenarios. Figures 5.12(a) and (c) indicate relatively lower prediction errors for those CCS sites, which correspondingly result in better average prediction results on test set for those two seeds. The reason lies in the fact that those nodes have higher degree centrality (due to the congregation), and consequently have more neighboring nodes to obtain the spatial dependency feature. Such augmented information would result in better prediction result. On the contrary, for testing nodes that are sporadically located, prediction error can be significant since very limited information, if any, can be retrieved in the neighboring region. In other words, if BFS only captures a few neighboring nodes, there is a higher chance to obtain larger biased prediction, as illustrated in Figures 5.12(d) and (e).



(a) (b)

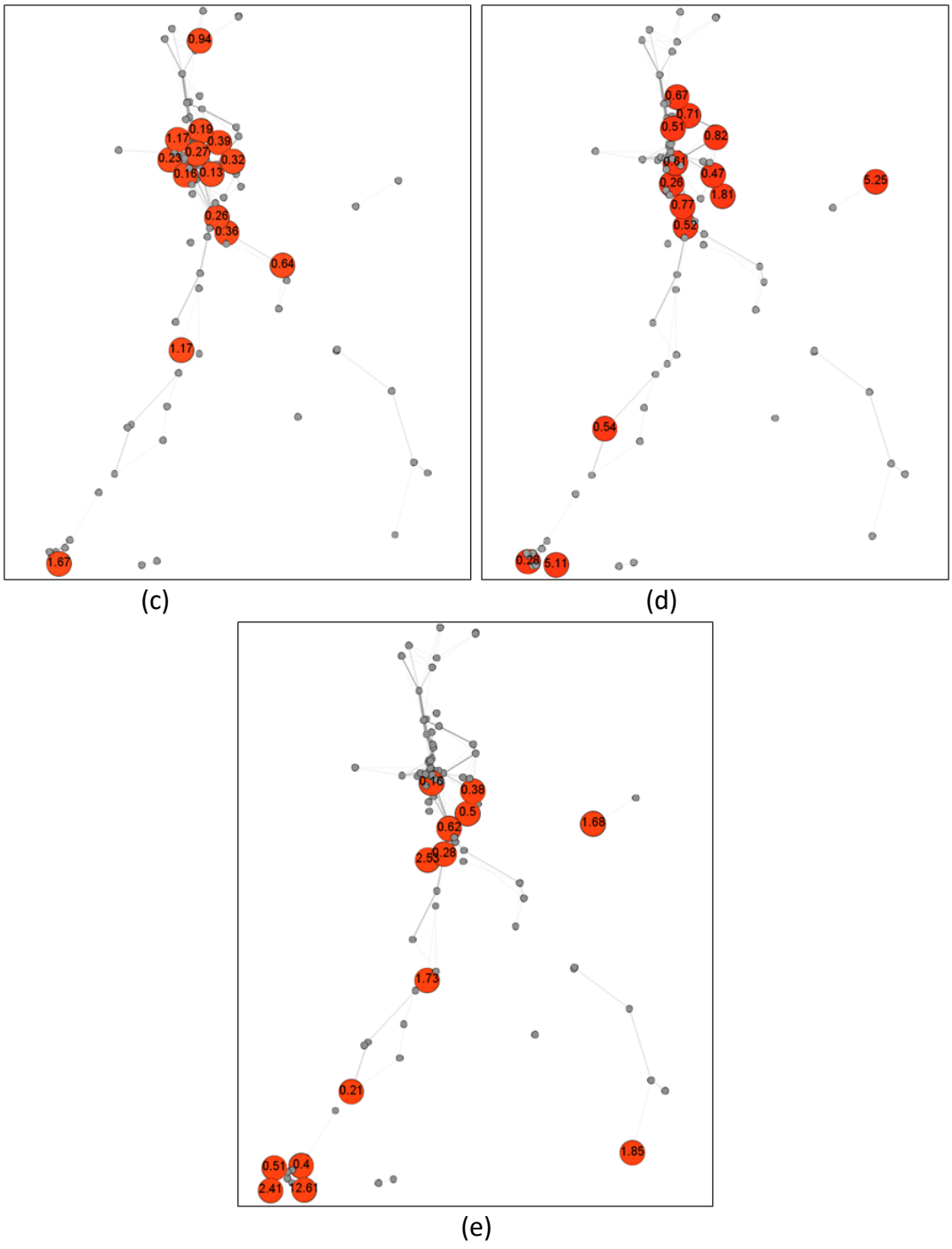


Figure 5.12 MAPE for the CCS sites in the test set, where (a) to (e) displays results across Seeds 1 through 5

Based on the above spatial correlation analysis, suggestions are provided for future CCS deployment strategy. In a traffic network, roads with high spatial correlation, such as adjacent segments along the freeway and roads with higher value of edge weight, tend to present similar traffic patterns. In contrast, isolated or remote roads in such networks might exhibit their unique traffic patterns and are difficult to extract information from neighboring areas because they are too distant. As a result, we suggest that CCSs should be extended with a wider coverage in remote or isolated areas, since traffic volume in spatially

correlated regions can be inferenced with higher accuracy, while estimating traffic volume on roads distributed sporadically tends to generate larger errors. Besides, the deployment of CCSs should be balanced with regard to the roads' functional classification. The level of traffic volume is highly relevant to functional classification. Yet the distribution of CCSs on roads with different functional classifications in our dataset is uneven. Such distribution results in the roads that fall into a minority class of functional classifications not being able to extract relevant traffic information from neighboring regions.

6. CONCLUSIONS

Hourly traffic volume possesses more valuable information than AADT for micro-level operational analysis. However, estimating hourly traffic volume at a new location with high accuracy can be quite challenging. This paper applies ML approaches to predict hourly traffic volume in a statewide road network managed by UDOT. The models utilize a series of features accounting for the spatial-temporal and traffic flow characteristics. Predictive capability and computational efficiency are measured and compared among proposed models through CCS sites in the state of Utah for 2017.

First of all, the performance of linear SVM and non-linear SVM are analyzed. It is found that the non-linear SVM model is not suitable for a large-scale training dataset due to its computational complexity. Although linear SVM can be trained with faster running time, it only achieves 0.52 for R^2 value on a test dataset, indicating a relatively poor prediction performance.

We further applied a computationally efficient tree ensemble model – XGBoost – to predict hourly traffic volume. The model utilizes a series of features that might affect traffic flows along with the proposed spatial dependency feature. In this study, a weighted traffic network graph is created to explore the spatial correlation between road segments. A graph-based approach – BFS – is implemented to search the neighboring sites and subsequently compute the spatial dependency feature. Predictive capability of the proposed model is assessed and compared with three different models: XGBoost model without spatial dependency feature, XGBoost model with a spatial dependency feature characterized by Euclidean distance, and XGBoost model with a spatial dependency feature characterized by network distance. The numerical result demonstrates the advantage of the proposed model over Euclidean distance and network distance when quantifying spatial correlation in a network. Specifically, numerical results of five random seeds show consistent outperformance of the proposed model with an average R^2, MAE, and MAPE being 0.9, 503, and 0.79, respectively. The average MAE and MAPE are reduced by 20.3% and 44.8% compared with XGBoost alone. Moreover, XGBoost is proven to be capable of training a large-scale dataset with high computational efficiency. Feature importance analysis further verifies the relevance of the proposed spatial dependency feature, accounting for 13.8% of the total importance among all features. In addition, spatial analysis shows that the proposed spatial dependency feature demonstrates its superiority for densely clustered regions. Future research will focus on exploring spatial prediction for nodes in sparse network and/or isolated islands.

7. REFERENCES

Alajali, W., Zhou, W., & Wen, S. (2018). "Traffic flow prediction for road intersection safety." *Proceedings - 2018 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovations, SmartWorld/UIC/ATC/ScalCom/CBDCo*, *d*, 812–820.

Alpaydin, E. (2009). *Introduction and Supervised Learning*. MIT press.

Breiman, L. (2001). "Random forests." *Random Forests*, 5–32.

Castro-Neto, M., Jeong, Y., Jeong, M. K., & Han, L. D. (2009). "AADT prediction using support vector regression with data-dependent parameters." *Expert Systems with Applications*, *36*(2 PART 2), 2979–2986.

Chen, T., & Guestrin, C. (2016). "XGBoost: a scalable tree boosting system." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794.

Chen, Z., Liu, X. C., & Zhang, G. (2016). "Non-recurrent congestion analysis using data-driven spatiotemporal approach for information construction." *Transportation Research Part C: Emerging Technologies*, *71*, 19–31.

Cheng, S., Lu, F., Peng, P., & Wu, S. (2018). "Short-term traffic forecasting: an adaptive st-knn model that considers spatial heterogeneity." *Computers, Environment and Urban Systems*, *71*(May), 186–198.

Du, L., Song, G., Wang, Y., Huang, J., Yu, Z., & Ruan, M. (2018). *Model for Expressway Network : A Network Flow Approach*. *January*, 107–120.

El Esawey, M., Mosa, A. I., & Nasr, K. (2015). "Estimation of daily bicycle traffic volumes using sparse data." *Computers, Environment and Urban Systems*, *54*, 195–203.

Friedman, J. (2001). "Greedy function approximation: a gradient boosting machine." author(s): Jerome H. Friedman source: *The Annals of Statistics*, vol. 29, no. 5 (Oct. 2001), pp. 1189-1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, *38*(4), 367–378.

Gastaldi, M., Gecchele, G., & Rossi, R. (2014). "Estimation of annual average daily traffic from one-week traffic counts. a combined ann-fuzzy approach." *Transportation Research Part C: Emerging Technologies*, *47*(P1), 86–99.

Gavirangaswamy, V. B., Gupta, G., Gupta, A., & Agrawal, R. (2013). "Assessment of arima-based prediction techniques for road-traffic volume." *Proceedings of the 5th International Conference on Management of Emergent Digital EcoSystems, MEDES 2013*, 246–251.

Habtemichael, F. G., & Cetin, M. (2016). "Short-term traffic flow rate forecasting based on identifying similar traffic patterns." *Transportation Research Part C: Emerging Technologies*, *66*, 61–78.

HERE Probe Data. (n.d.). https://udot3p.iteris-pems.com/

Inrix probe trajectory data. (2019). https://inrix.com/

Karlaftis, M. G., & Golias, I. (2002). "Effects of road geometry and surface on speed and safety." *Accident Analysis & Prevention 34.3*, *34* (November 1998), 357–365.

Kelley, J., Kuby, M., & Sierra, R. (2013). "Transportation network optimization for the movement of

indigenous goods in Amazonian Ecuador." *Journal of Transport Geography*, *28*, 89–100.

Kerkman, K., Martens, K., & Meurs, H. (2017). "A multilevel spatial interaction model of transit flows incorporating spatial and network autocorrelation." *Journal of Transport Geography*, *60*, 155–166.

Kurant, M., Markopoulou, A., & Thiran, P. (2010). "On the bias of BFS (breadth first search)." *2010 22nd International Teletraffic Congress - Proceedings, ITC 22*, 1–8.

Lam, W. H. K., & Xu, J. (2000). "Estimation of AADT from short period counts in Hong Kong - a comparison between neural network method and regression analysis." *Journal of Advanced Transportation*, *34*(2), 249–268.

Leduc, G. (2008). "Road traffic data: collection methods and applications." *EUR Number: Technical Note: JRC 47967*, *JRC 47967* (January 2008), 55.

http://ftp.jrc.es/EURdoc/EURdoc/JRC47967.TN.pdf

Leskovec, J., & Mcauley, J. J. (2012). "Learning to discover social circles in ego networks." In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 539–547). Curran Associates, Inc.

Liebig, T., Piatkowski, N., Bockermann, C., & Morik, K. (2017). "Dynamic route planning with real-time traffic predictions." *Information Systems*, *64*, 258–265.

Lowry, M. (2014). "Spatial interpolation of traffic counts based on origin-destination centrality." *Journal of Transport Geography*, *36*, 98–105.

M, S., Abd-EI-Latif, H., & Badra, N. (2007). "Comparison between regression and arima models in forecasting traffic volume." *Australian Journal of Basic and Applied Sciences*, *VI* (April 2015), 126–136.

Malenkovska Todorova, M., Donceva, R., & Bunevska, J. (2009). "Role of functional classification of highways in road traffic safety." *Transport Problems*, *4*(3), 97–104.

MesoWest. (2019). https://mesowest.utah.edu/

Pourebrahim, N., Sultana, S., Niakanlahiji, A., & Thill, J. C. (2019). "Trip distribution modeling with Twitter data." *Computers, Environment and Urban Systems*, *77* (July), 101354.

Salamanis, A., Kehagias, D. D., Filelis-Papadopoulos, C. K., Tzovaras, D., & Gravvanis, G. A. (2016). "Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction." *IEEE Transactions on Intelligent Transportation Systems*, *17*(6), 1678–1687.

Sekuła, P., Marković, N., Vander Laan, Z., & Sadabadi, K. F. (2018). "Estimating historical hourly traffic volumes via machine learning and vehicle probe data: a Maryland case study." *Transportation Research Part C: Emerging Technologies*, *97*(July), 147–158.

Selby, B., & Kockelman, K. M. (2013). "Spatial prediction of traffic levels in unmeasured locations: applications of universal kriging and geographically weighted regression." *Journal of Transport Geography*, *29*, 24–32.

Shi, Y., Gong, J., Deng, M., Yang, X., & Xu, F. (2018). "A graph-based approach for detecting spatial cross-outliers from two types of spatial point events." *Computers, Environment and Urban Systems*, *72* (October 2017), 88–103.

Smith, B. L., & Demetsky, M. J. (1996). "Multiple-interval freeway traffic flow forecasting." *Transportation Research Record: Journal of the Transportation Research Board*, *1554*, 136–141.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). "Practical bayesian optimization of machine learning algorithms." *In Advances in neural information processing systems*, 2951-2959.

Song, Y., Wang, X., Wright, G., Thatcher, D., Wu, P., & Felix, P. (2019). "Traffic volume prediction with segment-based regression kriging and its implementation in assessing the impact of heavy vehicles." *IEEE Transactions on Intelligent Transportation Systems*, *20*(1), 232–243.

Sun, Y., Yu, X., Bie, R., & Song, H. (2017). "Discovering time-dependent shortest path on traffic graph for drivers towards green driving." *Journal of Network and Computer Applications*, *83*, 204–212.

Tuv, E., Borisov, A., & Torkkola, K. (2006). "Feature selection using ensemble based ranking against artificial contrasts." *IEEE International Conference on Neural Networks - Conference Proceedings*, 2181–2185.

Wang, X., & Kockelman, K. M. (2009). "Forecasting network data spatial interpolation of traffic counts from Texas data." *Transportation Research Record*, *2105*, 100–108.

Williams, B. M., & Hoel, L. A. (2003). "Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results." *Journal of Transportation Engineering*, *129*(6), 664–672.

XGBoost Python API. (2016). https://xgboost.readthedocs.io/en/latest/python/python_api.html

Xia, Q., Zhao, F., Chen, Z., Shen, L. D., & Ospina, D. (1998). "Estimation of annual average daily traffic for nonstate roads in a Florida county." *Transportation Research Record*, *1660*, 32–40.

Xu, Y., Kong, Q. J., & Liu, Y. (2013). "Short-term traffic volume prediction using classification and regression trees." *IEEE Intelligent Vehicles Symposium, Proceedings*, *Iv*, 493–498.

Yuan, J., Bae, E., & Tai, X. C. (2010). "A study on continuous max-flow and min-cut approaches." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*, 2217–2224.

Zhan, X., Zheng, Y., Yi, X., & Ukkusuri, S. V. (2017). "Citywide traffic volume estimation using trajectory data." *IEEE Transactions on Knowledge and Data Engineering*, *29*(2), 272–285.

Zhang, D., & Wang, X. C. (2014). "Transit ridership estimation with network kriging: a case study of Second Avenue subway, NYC." *Journal of Transport Geography*, *41*, 107–115.

Zhang, Y., Lin, D., & Liu, X. C. (2019). "Biking islands in cities: an analysis combining bike trajectory and percolation theory." *Journal of Transport Geography*, *80* (February), 102497.

Zhao, F., & Chung, S. (2001). "Contributing factors of annual average daily traffic in a Florida county." *Transportation Research Record: Journal of the Transportation Research Board*, *1769*(01), 113–122.

Zhao, Z., Chen, W., Wu, X., Chen, P. C. V., & Liu, J. (2017). "LSTM network: a deep learning approach for short-term traffic forecast." *IET Intelligent Transport System*, *11*(1), 68–75.

8. APPENDIX A: MONTHLY FACTOR AND DAY-OF-WEEK FACTOR CALCULATION

Monthly factor M_i for each month based on functional classification

Month/Class	1	2	3	4
1	1.25	1.29	1.39	1.96
2	1.10	1.09	1.21	1.62
3	1.01	0.98	1.04	1.10
4	1.01	1.01	1.00	0.84
5	0.97	0.99	0.94	0.77
6	0.92	1.04	0.91	0.81
7	0.95	1.09	0.93	0.87
8	0.91	0.97	0.92	0.93
9	0.97	0.94	0.93	0.83
10	0.98	0.90	0.95	0.91
11	1.02	0.86	1.04	1.24
12	1.07	0.97	1.14	1.72

Day-of-week factor DOW_i for each day of week based on functional classification

DOW/Class	1	2	3	4
Sun	1.34	1.39	1.38	0.95
Mon	1.03	0.97	1.01	1.08
Tue	1.00	0.94	0.99	1.14
Wed	0.97	0.90	0.97	1.09
Thu	0.94	0.90	0.95	1.05
Fri	0.87	0.89	0.87	0.90
Sat	1.01	1.19	1.05	0.86