REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DATA AUGMENTATION AND MISSING DATA IMPUTATION USING SIMILAR TRAFFIC CHARACTERISTIC OF ROAD SEGMENTS FOR IMPROVING LONG-TERM SPEED PREDICTION

Mustafa Mert KARA

MASTER OF SCIENCE THESIS
Department of Computer Engineering
Program of Computer Engineering

Supervisor
Assoc. Prof. M. Amaç GÜVENSAN

Co-supervisor
Assist. Prof. H. İrem TÜRKMEN

August, 2023

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DATA AUGMENTATION AND MISSING DATA IMPUTATION USING SIMILAR TRAFFIC CHARACTERISTIC OF ROAD SEGMENTS FOR IMPROVING LONG-TERM SPEED PREDICTION

A thesis submitted by Mustafa Mert KARA in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 03.08.2023 in Department of Computer Engineering, Program of Computer Engineering.

Assoc. Prof. M. Amaç GÜVENSAN
Yildiz Technical University
Supervisor

Assist. Prof. H. İrem TÜRKMEN
Yildiz Technical University
Co-supervisor

Approved By the Examining Committee

Assoc. Prof. M. Amaç GÜVENSAN, Supervisor
Yildiz Technical University _____

Assist. Prof. Göksel BİRİCİK, Member
Yildiz Technical University _____

Prof. Dr. Berk CANBERK, Member
İstanbul Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Data Augmentation and Missing Data Imputation using Similar Traffic Characteristic of Road Segments for Improving Long-Term Speed Prediction supervised by my supervisor, Assoc. Prof. M. Amaç GÜVENSAN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Mustafa Mert KARA

Signature

This work was supported by the Turkish Scientific and Technological Research Council of Turkey (TUBITAK) under the grant number TUBITAK1001-120E357.

Dedicated to my family
and my advisors

ACKNOWLEDGEMENTS

I thank my advisor Assoc. Prof. Dr. M. Amaç GÜVENSAN for all the work he has done for this work. Without him and his guidence this work would never reach the conlusion that it would reach. I also thank my second advisor Assist. Prof. Dr. H. İrem TÜRKMEN for helping me see the problems from different perspective and find a solution to these problems.

I thank my family which worried about me constantly and supported me unconditionally.

Mustafa Mert KARA

TABLE OF CONTENTS

ACKNOWLEDGEMENTS v

LIST OF SYMBOLS viii

LIST OF ABBREVIATIONS ix

LIST OF FIGURES x

LIST OF TABLES xiii

ABSTRACT xiv

ÖZET xvi

1 INTRODUCTION 1

1.1 Goal of This Thesis 2

1.2 Hypothesis 3

2 LITERATURE REVIEW 5

2.1 Traffic Speed Prediction 5

2.2 Missing Value Imputation 8

2.3 Data Augmentation Through Similar Road Segments 10

3 IMPORTANCE OF TRAFFIC CHARACTERISTICS AND THEIR AREAS OF USAGE 12

3.1 Factors Effecting Traffic Speed 14

4 DATASET 22

4.1 Dataset Problems Faced in Long-term Prediction Domain 23

5 MISSING VALUE IMPUTATION THROUGH FINDING SIMILAR ROADS 26

5.1 Clustering Similar Segments 27

5.2 Proposed Missing Value Imputation Method 33

6 DATA AUGMENTATION THROUGH FINDING SIMILAR ROAD SEGMENTS **39**
6.1 Similar Road Detection Through Clustering Methods 40
6.2 Similar Road Detection Through Prediction Based k-NN Method . . 45

7 DATASET BALANCING THROUGH PATTERN ORGANIZATION **48**
7.1 Training Models with Hour-wise Pattern Organization 50
7.2 Training Models with Speed-wise Pattern Organization 52
7.3 Training Models with Month-wise Pattern Organization 56

8 EXPERIMENTAL RESULTS **63**
8.1 Missing Data Imputation Results 64
8.2 Data Augmentation Results 69
8.3 Imbalanced Dataset Results 73

9 CONCLUSION **79**

REFERENCES **81**

PUBLICATIONS FROM THE THESIS **87**

LIST OF SYMBOLS

km/h	Kilometer per Hour

LIST OF ABBREVIATIONS

AI	Artificial Intelligent
CNN	Convolutional Neural Networks
DTW	Dynamic Time Warping
GCN	Graph Convolutional Networks
ITS	Intelligent Transportation Systems
k-NN	k-Nearest Neighbor
LLS	Least-Local Squares
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NN	Neural Networks
SMOTE	Synthetic Minority Oversampling Technique

LIST OF FIGURES

Figure 3.1 General overview of factors effecting traffic speed values. 14
Figure 3.2 Prediction horizon displaying short-term and long-term prediction 14
Figure 3.3 Monthly average traffic speed changes of 3 years(2017, 2018, 2019) in Istanbul 18
Figure 3.4 Average traffic speed changes based on week days for 3 years in Istanbul 18
Figure 3.5 Hourly traffic speed changes of 3 years in Istanbul 19
Figure 4.1 Histogram of the missing percentage of 441 road segments. The X-axis shows the percentage of missing data while the Y-axis shows the number of segments. The numbers on the bars shows the percentage of segments that belongs to the bin compared to the total amount of segments. 23
Figure 4.2 Distribution of speed in the dataset. 24
Figure 4.3 Percentage of data in each group 24
Figure 5.1 Block diagram of Base Clustering method. 29
Figure 5.2 Block diagram of Feature Based Clustering method. 29
Figure 5.3 Block diagram of Dynamic Time Warping Based Clustering method. 30
Figure 5.4 Block diagram of Traffic Hours Based Clustering method. . . . 30
Figure 5.5 Block diagram of DBScan Clustering method. 31
Figure 5.6 Block diagram of LLS Method[65]. 31
Figure 5.7 [76]Difference between Dynamic Time Warping and Euclidian Distance. In comparison between two-time series data, there might be shifts that render Euclidian distance inefficient. DTW takes this type of shift into consideration when comparing two time-series data. 32
Figure 5.8 The block diagram of the proposed methodology for the detection of k similar segments to a given segment and prediction of the corresponding timestamp with a missing value 34

Figure 6.1 Block diagram of the clustering process. The clustering method in this diagram changes through different methodologies used but the general process remains the same with data clustered and a model is trained for each cluster separately using the data of segments belonging to the cluster. 40
Figure 6.2 Silhouette score graphic obtained with base clustering methodology with the different number of clusters. 43
Figure 6.3 Silhouette score graphic obtained with traffic hours clustering methodology with the different number of clusters. 44
Figure 6.4 Silhouette score graphic obtained with model creation based on time of day methodology with the different number of clusters. The Figure shows from the top left to staring left in each row, morning, noon, afternoon, evening, and night 44
Figure 6.5 Block diagram of the proposed method of prediction based k-NN demonstrated with 4 segment scenario. Abbreviations used in this diagram like S1 and M1 mean segment 1 and model 1 respectively. The values in the matrix show the distances between segments for example D1-2 shows the distance between segment 1 to segment 2 and is used when finding similar segments to segment 1. 47
Figure 7.1 (a) Average speed of 441 segments separated into 5 different groups based on time of day. (b) Percentage of data in each group 49
Figure 7.2 An example of how model creation using Hour-wise Pattern Organization method works 51
Figure 7.3 MAPE comparison between model training using Hour-wise Pattern Organization method and normal LSTM(up) and the difference between two model results(down) where green parts are where the applied method is better and red parts are where normal LSTM is better. 52
Figure 7.4 An example of how model creation using Speed-wise Pattern Organization method works 54
Figure 7.5 Dataset grouping used in Speed-wise Pattern Organization. Upper side of the Figure show the grouping of the test samples while the down side show the grouping of the train samples. . . 55
Figure 7.6 Comparison of normal LSTM model and model training using Speed-wise Pattern Organization method. First graphic shows MAPE values of two methods while the second graphic shows MAPE difference between them. 56
Figure 7.7 Speed values of 5 consecutive months 57

Figure 7.8 MAPE values of every month 58
Figure 7.9 Histograms of months April and August 59
Figure 7.10 Month similarity between each other 59
Figure 8.1 The experimental duration diagram for our test scenarios 64
Figure 8.2 MAPE change regarding the missing value ratio. A percentage of 20% and lower is negligible while a percentage between 20% and 50% is acceptable and any percentage above 50% is not recommended. 68
Figure 8.3 MAPE scores obtained with different number of k values using Prediction based k-NN method 71
Figure 8.4 MAPE difference between regular long-term prediction and data augmented long-term prediction. 71
Figure 8.5 MAPE scores obtained with 4 base models. Base models show MAPE scores obtained with no additional method applied. Hour-wise, Month-wise and Speed-wise Pattern Organization shows the MAPE obtained with methods proposed for imbalanced dataset problem. Speed-wise and Month-wise shows the results obtained using the two methods together. . . . 73
Figure 8.6 Hourly MAPE difference between models trained with no additional process and models trained with Hour-wise Pattern Organization method. 75
Figure 8.7 Hourly MAPE Difference between models trained with no additional process and models trained with Speed-wise Pattern Organization method. 76

LIST OF TABLES

Table 4.1 Dataset and input specifications 22
Table 4.2 Dataset sizes of the year 2018 25
Table 5.1 The proposed techniques for finding similar segments 28
Table 5.2 Mape values for different numbers of k similar segments 35
Table 5.3 Mape values and number of clusters for the proposed clustering techniques 37
Table 6.1 Number of clusters created for each category based on the hour of day 45
Table 7.1 MAPE comparison between LSTM model trained with our 3 methods and LSTM model that has no methods applied in training 50
Table 7.2 6 closest months found 61
Table 7.3 MAPE scores obtained with using Month-wise Pattern Organization(MPO) for training with LSTM, Bi-Directional LSTM, GRU, and CNN models and Base model which uses no method for training 62
Table 8.1 Server spesifications 63
Table 8.2 Model parameters for long-term traffic speed prediction 65
Table 8.3 Long-term speed prediction MAPE for different percentage of missing value replaced in selected 20 road segments 65
Table 8.4 Long-Term traffic speed prediction results before and after the proposed missing data imputation method has been applied . . . 67
Table 8.5 MAPE scores obtained with clustering methods for 106 selected segments 70
Table 8.6 MAPE scores obtained with Prediction based k-NN Method . . . 72
Table 8.7 MAPE scores obtained with our method with LSTM, Bi-Directional LSTM, GRU, and CNN models 74
Table 8.8 Experimental setup and server spesifications 74
Table 8.9 MAPE scores received with using Month-wise Pattern Organization(MPO) for training with LSTM, Bi-Directional LSTM, GRU and CNN models and Base model which uses no other method for training 77

ABSTRACT

Data Augmentation and Missing Data Imputation using Similar Traffic Characteristic of Road Segments for Improving Long-Term Speed Prediction

Mustafa Mert KARA

Department of Computer Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. M. Amaç GÜVENSAN
Co-supervisor: Assist. Prof. H. İrem TÜRKMEN

Traffic speed estimation is a pivotal concern in Intelligent Transportation Systems. Tailoring models to distinct road segments is imperative due to varying traffic characteristics. However, speed data collected by road sensors are plagued by infrequent readings due to errors, leading to substantial missing data. This, coupled with the necessity of relying on limited road segment data, diminishes training dataset sizes for speed prediction models. Moreover, the inherent imbalance in datasets, attributed to traffic speed, introduces bias in trained models.

This thesis introduces novel solutions to address these issues—data scarcity, missing data, and imbalanced datasets. The first approach involves augmenting dataset size by integrating data from diverse road segments with comparable traffic traits. Using prediction models, we identify road segment similarities, leading to a 3.7% boost in long-term traffic speed prediction performance. Addressing missing data, we leverage data from analogous road segments, employing a k-NN algorithm variation. This technique yields a 3.6% prediction performance improvement. For imbalanced datasets, we propose three strategies that reshape the model's learning process, resulting in a 2.5% enhancement in long-term prediction. Notably, our methods excel in capturing specific patterns, particularly low-speed instances, elevating their performance by 5%.

This thesis concentrates on improving long-term prediction accuracy, solving issues

encountered in traffic speed datasets. We extensively validate our approaches across diverse long-term prediction models, highlighting their versatility and robustness. Performance enhancement is consistent across all trained models, affirming the effectiveness of our approaches. Experiments employ a traffic speed dataset collected by Istanbul Municipality, encompassing 441 main road segments from 2018 and 2019.

In essence, our research tackles crucial challenges in Intelligent Transportation Systems' traffic speed estimation. By increasing dataset size, handling missing data, and addressing dataset imbalances, our methodologies bolster the accuracy and robustness of long-term traffic speed prediction models. Consequently, our work contributes valuable insights for traffic management and transportation system optimization.

Keywords: Long-term traffic speed prediction, Missing data, Imbalanced dataset, Data augmentation

ÖZET

Uzun Vadeli Trafik Hızı Tahminini İyileştirmek için Benzer Trafik Karaktestliğine Sahip Yolların Kullanılarak Veri Arttırılması ve Kayıp Verilerin Giderilmesi

Mustafa Mert KARA

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. M. Amaç GÜVENSAN
Eş-Danışman: Dr. Öğr. Üyesi H. İrem TÜRKMEN

Trafik hızı tahmini, Akıllı Ulaşım Sistemleri içinde kritik bir problemdir. Farklı trafik karakteristikleri nedeniyle modellerin her yol segmenti için özelleştirilmesi gerekmektedir. Bununla birlikte, yol sensörleri tarafından toplanan hız verileri hatalar nedeniyle ciddi oranda eksik veriler içermektedir. Bu durum ve sınırlı yol segment verisi, eğitim veri kümesinin boyutunu düşürmektedir. Ayrıca, trafik hızına özgü veri kümelerinin doğasından kaynaklanan denge sorunu, eğitilmiş modellerde sapmaya neden olmaktadır.

Bu tez, bu sorunlara yeni çözümler sunmaktadır. İlk metot, farklı trafik özelliklerine sahip yol segmentlerinden gelen verileri birleştirerek veri kümesi boyutunu artırmayı içermektedir. Tahmin modelleri kullanılarak yol segmenti benzerliklerini bulunmaktadır. Bu benzerlik kullanılarak arttırılan veri seti, uzun vadeli trafik hızı tahmini performansını %3.7 arttırmaktadır. Eksik veri sorununu çözmek için benzer yol segmentlerinden gelen veriler kullanılmıştır ve bu benzer segmentler bir k-NN algoritması varyasyonu ile bulunmaktadır. Bu metot, %3.6 tahmin performansı iyileştirmesi sağlamaktadır. Dengesiz veri kümeleri için, modelin öğrenme sürecini değiştiren üç strateji önerilmiştir, bu metotlar da uzun vadeli tahminde %2.5 artışa sağlamaktadır. Özellikle düşük hız anlarını yakalama konusunda yöntemlerimiz %5 performans artışı göstermektedir.

Bu tez, trafik hızı veri kümelerinde karşılaşılan sorunları çözme ve böylece uzun vadeli tahmin doğruluğunu artırmaya odaklanmıştır. Yöntemlerimizi farklı uzun vadeli tahmin modelleriyle kapsamlı bir şekilde doğrulanmıştır. Deneylerimiz önerilen metotların esneklik ve sağlamlıklarını kanıtlamaktadır. Deneyler İstanbul Büyükşehir Belediyesi tarafından toplanan trafik hızı veri kümesi üzerinde gerçekleştirilmiştir. Bu veri kümesinde, 2018 ve 2019 yıllarında toplam 441 ana yol segmenti bulunmaktadır.

Sonuç olarak, bu araştırmada, Akıllı Ulaşım Sistemleri'ndeki trafik hızı tahminindeki temel zorluklar ele alınmıştır. Veri kümesi boyutunu artırma, eksik verileri giderme ve dengesiz veri kümelerini düzeltme metotlarıyla uzun vadeli trafik hızı tahmin modellerinin performansı arttırılmıştır. Ayrıca bu çalışma, trafik yönetimi ve ulaşım sistemi optimizasyonu için de önemli bilgiler barındırmaktadır.

Anahtar Kelimeler: Uzun dönemli trafik hızı tahmini, kayıp veri, dengesiz veriseti, veri sayısı arttırma

1
INTRODUCTION

Intelligent Transportation Systems (ITS) is a popular area of research as the population and incidentally, the size of the cities grows. This growth requires smart approaches to day-to-day transportation. While sometimes these approaches are utilized by governments for city planning other times they are used by individuals for problems like traffic jams, navigation, or accident avoidance. Within this area, a prominent problem is the prediction of traffic speed which is sometimes called as traffic flow prediction. This problem is a complex multi-variable problem that is plagued by many non-data-related problems.

There are two types of traffic speed/flow predictions. The first type is the most common one which is called short-term traffic speed prediction. This type of model predicts traffic speeds up to 6 hours in the future. The second type is called long-term traffic speed prediction which predicts from 6 hours to a week in the future. Although types of prediction are unanimously agreed, the prediction horizons of these models are less clear. Some researchers in literature call anything above an hour long-term speed prediction([1]) while others add a new type of medium-term prediction to separate the short and long-term predictions([2]). Even though prediction horizons are not clearly defined this distinction between long and short-term prediction is important as the methods and variables used in these two types differ from each other and many methods and variable that works on one doesn't work in the other.

To create a robust model to predict traffic speed, a model must learn patterns that depend on time, as traffic changes from hour to hour, day to day, and month to month while also learning spatial variables as traffic in one road is related to other road segments. While these variables are related to traffic speed directly there are non-speed data that also affect traffic speed such as weather([3]) which has a significant impact on speed, holidays and special events([4]), accidents([5]), and even whether schools are open or not([6]). A model doesn't necessarily consider all these variables to make a prediction. In fact, traffic speed is the only variable needed

to make a future prediction but in such cases, models are incapable of capturing the impact of these variables on traffic speed and most recent work tries to take more variables into account for prediction problems. Some of these variables are naturally more important in short-term predictions than long-term predictions such as spatial data that has a more direct effect in a one-hour horizon than a week in the future while periodicity like weekly speed is more important in long-term predictions than short-term predictions.

Regardless of whichever variables decided to be used, there is a substantial amount of data required to train a traffic speed prediction model and on the contrary, literature lacks datasets with the necessary amounts of data(for example [7] uses a dataset that has only 8 days). Most research is conducted on datasets with few months of data[[8], [9]] or with only a few road segments[[10], [1]] which limit the model's ability to learn these complex dependencies or show its capabilities. Thus, the aspect of data augmentation is inseparable from traffic speed prediction and ITS research. Another problem in this area is caused by data collection tools such as traffic sensors which are prone to error and cause missing values. This is a common problem in literature and requires additional effort to remedy this problem.

1.1 Goal of This Thesis

This thesis aims to address major problems existing within traffic speed datasets, namely *Missing data, Lack of data, and imbalanced data* issues. Our work focuses on finding novel solutions to increase the accuracy of long-term traffic speed predictions. We have chosen long-term predictions as our goal due to the lack of research in this domain regarding long-term predictions.

Missing data imputation methods can be categorized into two groups based on the data used for imputation. The first category uses the target segment's own data and finds similar time periods, such as a day, to fill the missing data. The second category is to find similar road segments and use the same time period to fill the missing data. The second method has the advantage of encompassing environmental factors, such as special events, accidents, and holidays, as the data used for imputation exist in the same time period. Using similar road segments also provides access to more data for utilization, resulting in a closer approximation to the missing data. In this thesis, we introduce a novel method of missing data imputation, which is a variation of the k-Nearest Neighbor (k-NN) algorithm. We find similar k segments and then measure the distances of these k segments to the targeted segment. Aterwards we apply these distances as weights for creating

imputation values. We compare our method with different methods used in the literature to test its effectiveness.

For data augmentation, we propose a method to find similar road segments and use data from these similar segments together for training the model. We utilize prediction models trained on each road segment to evaluate other road segment data and find similarity based on the error score obtained as a measure of characteristic similarity. We compare our results with clustering methods for long-term prediction and demonstrate the effectiveness of the proposed method.

Imbalanced datasets cause bias in prediction models and lower prediction performance. We propose a pattern organization method to change the training dataset to emphasize certain traffic characteristics based on hours, months, and speed values. Changing the training dataset mitigates the imbalance in the dataset, enabling the trained models to capture the nuances within the data more effectively.

We conducted experiments with different model types to prove the flexibility and robustness of our methods and show their effectiveness under different circumstances. These results also demonstrate the potential of our methods for use in future works. Our experiments were conducted on a dataset collected by Istanbul Municipality over 2 years, comprising 441 main road segments. Mean Average Percentage Error (MAPE) and Mean Absolute Error (MAE) were used for testing model accuracy. We believe we are the first ones using these methods in the traffic speed prediction domain and are among a few works that focus on the long-term speed prediction problem.

1.2 Hypothesis

For this thesis certain hypothesis are made for increasing long-term traffic speed prediction performance. These hypotheses are:

- Traffic speed datasets have smaller sample sizes compared to other deep learning problems thus deep learning methods used for traffic speed prediction have trouble learning the patterns in the dataset.
- Using any different road segment data may not always have beneficial effects on prediction performance.
- Finding road segments that have similar traffic characteristics via clustering may not result in better performance as each road segment is different and general solutions like clustering may not fit all the segments within the cluster.

Thus specialized solutions for each road segments produce better results than clustering techniques.

- Finding similar road segments based on the long-term traffic characteristics and using the data of similar road segment together would increase the prediction accuracy of long-term predictions.
- Changes to the traffic characteristics based on spatio-temporal factors can be learned more efficiently if these factors are isolated in training dataset.
- Reorganizing training dataset based on aspects of traffic characteristics like hours, months or the speed values would balance the dataset and improve the prediction accuracy.
- Using similar segments of data for missing data imputation would increase the prediction performance.
- Using missing data imputation for road segments that have missing values more than 20% and less than 50% of the entire dataset would produce results with acceptable degradation of accuracy and if these segments are important then they can be used after their missing data are filled. Using missing data imputation for road segments that have missing values less than 20% of the entire dataset would produce results with marginal loss of accuracy and they can be used after missing data imputation. Road segments with more than 50% missing data should be discarded as any imputation attempt would results in substantial loss of accuracy.

2
LITERATURE REVIEW

Domain of this thesis has three aspects. First aspect is the traffic speed prediction, second aspect is the missing value imputation, and the third aspect is the Data augmentation through finding similar road segments. Although most works in literature includes all three in some ways, we review the literature mainly through these three categories.

2.1 Traffic Speed Prediction

Traffic forecasting is an important subject in Intelligent Transportation Systems. As such it has been a popular topic in literature. Historically there has been a moderate success with statistical models. Among these models, ARIMA(autoregressive integrated moving average) has achieved the most success and is still used as a benchmark because of its ability to capture the periodicity of traffic data and robustness to outliers. Ahmed et al.[11] used ARIMA with great success while comparing it to the other statistical model such as moving-average, double-exponential smoothing, and Trigg and Leach adaptive models. Later on, variations of ARIMA also saw popularity in literature, Van Der Voort et al.[12] used ARIMA with Kohonen maps to divide days into different clusters and build an ARIMA model for each cluster,Lee et al.[13] used subset ARIMA to get better results than regular ARIMA model, Williams et al.[14] on the other hand used Seasonal ARIMA with a weekly lag value to reach better results than Kohonen maps ARIMA. Another paper that uses a statistical method is [15] in which authors uses Gaussian maximum likelihood with both historical and real-time information to create a model that outperforms ARIMA and linear regression models.

After the rise of Machine Learning models, although ARIMA stayed as a hot topic among researchers, most of the attention has shifted to more complex models. k-NN and SVR models were extensively explored for traffic speed prediction and traffic flow prediction. k-NN as a non-parametric model has been used with moderate

success[16]. Over the years k-NN has seen improvement to its success[[17], [18]] and still worked on even recently [[19], [20]]. SVR, on the other hand, has shown its potential for traffic prediction earlier and is still used in many recent papers as a benchmark model. This is due to the robustness of the support vector model and its ability to capture periodic characteristics. [21] and [22] used SVR for traffic prediction more specifically using it for travel time prediction. Although SVR gives good results, it needs to be trained for each new entry so [23] suggested online training for SVR to predict traffic flow.

At some point, researchers started to think of traffic prediction as a non-linear problem and with the availability of data increasing over the years Neural networks and deep learning methods became the focus of the Traffic prediction problem. [[24], [25] and [26]] are early works of using Neural networks for short-term traffic prediction. More recent papers using NN like [27] used NN with ensemble model structure along with the average speed of the previous weeks. Qu et al.[1] used NN unlike previous papers to predict an entire day with success, using contextual data along with NN prediction. Lane et al.[10] used spiking neural networks which is a variant of the neural network architecture along with accident data to successfully make traffic predictions considering accident possibility.

Although NN could yield good results, it could hardly capture the complexity of traffic data which has both spatial and temporal qualities. To capture these CNN and LSTM[28] variants are used widely in current research. [[29], [30], [31]] are early work with LSTMs without much change to the original LSTM model or new preprocessing steps to adapt LSTM to traffic data. Due to the compatibility of LSTM and traffic data, even these types of models could produce better results than statistical or machine learning models. But in recent years different techniques have been used for increasing prediction accuracy. GRU(Gated Recurrent Units)[32] are a recurrent neural network model that works similarly to LSTMs but has less amount of variables to learn data relations. In certain domains, this change in variable number could result in better accuracy as the model would need less amount of data to train. Fu et al.[33] compared LSTM and GRU and in their experiments, they found that GRU performed better than LSTM on traffic flow prediction. Another interesting change to LSTM structure was the attention mechanism[[34],[35],[36], [37], [38], [2], [39], [40]] which became a de facto technique to use with LSTMs and GRUs.

CNN on the other hand is better at capturing spatial properties of traffic data. Different road segments and their relations with each other help speed prediction greatly. But this requires roads' relation to each other to be defined. Earlier work

with CNNs took this relation as a grid map to make this relation Euclidean so that CNNs could learn it. Ma et al.[41] took traffic information and transform it into a 2d image and used Deep CNN to extract information from this image. Zhan et al. [42] used residual networks to predict crowd flows. An et al.[5] uses CNN combined with fuzz logic to successfully consider accident situations. Zang et al. [43] uses convolution layers in the GAN structure to make a daily speed prediction through the generator then trains the generator with a discriminator that validates true speed data with generated data. To test this model authors used one year of two road segments and make predictions of the day after. They achieved 32 MAE in high traffic low-speed moments. He et al.[44] predicts with two different models. One of them makes a prediction based on the day of the week while the other one makes a prediction for all days and then combines both to make a long-term prediction that has a time horizon of a week. Zang et al. [45] used Conv-LSTM with three different time scales and combined them through CNN's to make a long-term speed prediction of a day ahead. In this paper, unlike most other research, a dataset of 1 year with 149 road segments is used. They achieved 6.5 MAE for all road segments.

Regular Convolutional methods fail to grasp the nature of spatial information in traffic data. Roads in a city build a graph and a grid cannot capture this complex relationship thus Graph CNN[[46], [47]] were used extensively in traffic prediction. Yu et al.[48] used Spatio-convolutional Blocks to capture graph structure while Wu et al.[49] used GCN structure in a Gated block to capture spatial and temporal characteristics. [50] uses graph convolution layers to guess traffic flow a day later with a MAPE score of 10. Most recent models use a temporal model alongside GCN to capture both of these characteristics. Bogaerts et al. [51] finds similarities of roads before training, it first uses a GCN to learn these spatial relationships and then uses LSTM to capture temporal information. Han et al.[52] uses daily traffic data to find other similar roads and uses this similarity to quantify the effect of each road on the predicted part of the day. [53] uses a custom block combining GCN and GRU structures to predict traffic speed.

Relationships between the roads are hard to find with statistical methods. State-of-the-art models utilize dynamic correlation between roads and learn this relationship while training a prediction model. [35] uses a dataset of the year 2015 that has speed values of 5-minute windows with a model made out of a hybrid of GRU and Graph Convolution layers and uses a self-attention mechanism to learn the k-hop neighbors' effect on the current road segment. It predicts 60 minutes ahead with 2 MAPE. [39] also uses an attention mechanism to find road relations but unlike other papers, it does not define a connected graph and learns the relations of every road on the map.

Another popular deep learning method is to extract information from data and then use this extracted data with supervised or unsupervised models to solve the initial problem. [54] used Deep Belief networks to learn a feature space for traffic speed and location data while James et al.[50] used stacked auto-encoders to predict traffic flow. Zheng et al.[40] used transformers to build an encoder-decoder structure. [55] uses graph CNN in seq2seq configuration to make a prediction directly based on encoded historical data. They make predictions of two hours with a MAPE score of 19.41.

2.2 Missing Value Imputation

Missing data is a common problem for all types of research that deal with datasets. It is usually handled either by removing the missing features and/or all samples with missing data or by filling in the missing values. Both methods have their own problems. Although removing the data cleans the dataset, if the dataset is small or there are many missing values then it may render the dataset useless. Filling the missing values on the other hand leaves us a bigger dataset but with an important question: What value to use? Simple statistical methods are generally calculates mean of a feature with or without considering classes of the sample, or a static value is determined through domain knowledge. Sometimes a null value is used for learning certain patterns from the missing portion of the data. Recently with the advancement of machine learning, the focus on missing data imputation has shifted towards more complex methods which produce more appropriate values tailored to the sample itself.

Since datasets created through sensors are more susceptible to missing values as sensors often have problems collecting or sending data, traffic speed or traffic flow datasets collected from multiple sensors might contain many missing values. Because traffic produces a time-series data, a missing value effect not just itself but every sample within its time window. That is why missing data imputation is very important for intelligent transportation research.

Generally, missing data imputation can be grouped into 4 categories: *statistical, prediction, decomposition,* and *data-driven*. Means, medians, and modes of features can be counted among the frequently used statistical methods. Class-specific data imputation yields better results in the case of having a sufficient amount of data in each class. For time-series problems, using the last observed values is a common approach along with using normal and weighted averages[[56], [8], [57]]. The prediction method trains a model with complete data and predicts missing values.

Cui et al. used a graph Markov network for filling in the missing values and then used another graph markov model for traffic forecasting[58]. In [57], researchers exploited a Long-Short Term Memory(LSTM) variant that they call LSTM-M to predict complex missing value patterns. This type of method has the disadvantage of requiring a sufficient number of full samples for training and is affected by the error of the prediction model. Therefore, it is usually unable to capture anomalies in the dataset which results in low accuracy of the model.

The decomposition model usually uses matrix or more recently tensor decomposition for missing data imputation. In [59], authors first determined a periodic value based on the historical data. Then a matrix of all the roads in the dataset is created. They proposed a novel matrix decomposition method and combined the periodic value and the values obtained from matrix decomposition in order to fill in missing values. In [60], researchers suggested using a variant of High Accurate Low-rank Tensor Decomposition (HaLRTC) which first benefits from a clustering method to find the most similar road segments. In the second step, they exploit HaLRTC for each cluster to fill in missing values. Chen et.al. used Singular Value Decomposition-based tensor decomposition [61], while Liu et.al. proposed High Accurate Low-rank Tensor Decomposition (HaLRTC) for missing value imputation [62].Chen et.al. used a tensor completion by emphasizing temporal characteristics of traffic data and produced values that capture seasonal and daily periodicity [63].

Data-driven methods process the available data to find the most appropriate value for the imputation. The main difference from statistical methods is the employment of more complex methods that usually involves deep learning. In [64] and [65], researchers used local least squares which is a variant of the k Nearest Neighbor(k-NN) method that uses a weighted average instead of a normal average while combining k closest vector. The first study seeks for the closest days among all the road segments while the latter looks for the closest day for each segment within their historical data. Another popular approach is clustering similar segments. In [66], researchers make a cluster of the historical data to find days similar to the ones that contain missing values. They also proposed a model that uses road segments geographically closer to each other to find missing values. Another data-driven method is data generation through Generative Adversarial Networks (GAN). Yoon et al. first proposed a GAN model to generate missing values in a dataset[67]. Wang et al. used the same logic as in [67] but they first made clustering to produce pseudo labels for samples[68], since GAN performs better using labels.

2.3 Data Augmentation Through Similar Road Segments

Data augmentation in traffic speed prediction is applied to increase the sample size of the training dataset. Usually, dataset size in this domain is not as big as other domains such as image or text. The reason is that road segments have different traffic characteristics and training a model using all of the data results in worse performance than when every segment is predicted via a model trained on that segment data only. Even in the case of using only one segment's data different months show different characteristics thus model training cannot be done with every data of the road segment as well. The solution to this problem is to simply have more data that contains the same characteristics as the road segment that is predicted. This solution creates the main problem of data augmentation in traffic speed: How to find similar characteristics in traffic speed data?

The simple solution to this problem is to use the adjacent road segments. Kim et al. [9] used the adjacent road segments to train a model with a bigger dataset. The authors worked with 4 months of data and found that this solution lowered the MAPE score from 21.4 to 19.72 which is a 7.8% improvement. The results obtained from this work are obtained with a short-term prediction model as is most research done in this domain. Another simple solution is to use methods to increase the sample size of minority classes. Synthetic Minority Oversampling Technique (SMOTE) samples the minority classes to increase their representation and increase the sample size for models to learn the patterns in the dataset better. Ramhormozi et al. [69] used SMOTE for short-term prediction of 6 hours with 2 months of data that has 30-minute data frequency. The authors were able to decrease the MAPE score from 7.11 to 6.88 which is a 3% performance improvement.

A popular method used in this domain is unsupervised learning which creates clusters of similar items that are road segments for our problem. Zhang et al.[8] used a clustering method with a dataset of 2 months that was collected with 5-minute frequency for short-term prediction. The authors were able to decrease the MAPE score from 11.82 to 10.39 which is a 6% increase. The trained model only made predictions between the hours 06:00 and 22:00 with a prediction horizon of 10-minutes. Another work that used clustering for data augmentation found that using data that belongs to the predicted road segment results in better prediction performance than training a model for a cluster [70]. GAN is another method used for creating synthetic data for data augmentation. [7] has used GAN for this purpose with 8 days of data that is collected with 5-minute frequency. The main goal of this work is not traffic speed prediction but rather cellular traffic prediction which has a very similar structure both in data type and problem structure. The resulting dataset

has managed to decrease the MSE score from 0.0057 to 0.0042.

The goal of this thesis is to investigate the prominent problems faced in traffic speed prediction that are related to the data characteristics. We defined three main problems and presented the work that is done on these problems along with research done on the prediction problem itself. These problems especially imbalances in data have not been researched widely and the focus in this domain remains on types of models rather than data engineering. Another bias that exists in this domain is the prediction horizon where the literature focuses on short-term prediction and neglects long-term prediction which is important and incompatible with short-term prediction methods. In this thesis, we focus on data processing rather than the model used for solving problems and our focus is on long-term traffic prediction.Our methods exist within the preprocessing step thus they can be used with any number of model types with marginal differences in their effectiveness. This flexibility of the proposed methods can be observed in the experimental results chapter. In the upcoming chapters, we propose 5 novel methods for the problems faced in traffic speed prediction. We test each of our methods with a dataset collected by Istanbul Municipality over the course of 2018 and 2019. The problem definition and factors that affect this problem are given in Chapter 3. Then in the next Chapter we give the spesifications of the dataset we have used in this thesis. We also show certain statistics that display the problems we have discussed previously. In Chapter 5, we show the methods we have experimented with along with the the proposed method for missing data imputation. In Chapter 6, we discuss the data augmentation methods we have attemted to use including the proposed method. In Chapter 7, we show our methodology for balancing the dataset with 3 novel techniques we have employed for this purpose. In Chapter 8, we give the experimental results and In Chapter 9, we summarize our findings and discuss future work that could be done in this domain.

3
IMPORTANCE OF TRAFFIC CHARACTERISTICS AND THEIR AREAS OF USAGE

Traffic speed prediction is a complex problem that depends on many different variables. Some of these variables affect traffic speed directly like weather or accidents while others are indicators of patterns for traffic speed like location or day of the week traffic speed average. All of these variables are used with different combinations and different significance in traffic speed prediction. While discussing these variables' significance for traffic speed prediction we must discuss the prediction horizon as it is the most important determinant of variable usage. The complexity of the problem scope is not the only problem faced in traffic speed prediction. Another problem is the dataset as traffic speed dataset has three inherent problems:

The first of these problems is caused by the way the dataset is collected. There are three popular methods to collect traffic speed: measuring the speed of one car through a road segment, measuring the difference between the entry and exit time of a car through a road segment, and detecting the speed of cars passing through a road segment directly. All of these methods have one thing in common, they are all collected by sensors which are prone to failings. These failings are common occurrences in all of the traffic datasets in the literature and they create the problem of missing values. As this data is a time series most methods require a window of time for input and so one missing value could cause many samples to be non-useable.

The second of these problems is the sample size of the dataset. Datasets collected for traffic speed prediction have two important aspects: first is the data collection frequency and the second is the data collection period. The first aspect, the granularity, limits the amount of data one day may have but it also affects the prediction performance. In the paper[71] authors found that less granularity increases the performance this is caused by the nature of the traffic speed data which

is volatile. This volatility can be remedied with a smoothing technique which is basically what happens when granularity increases. Although performance increase is an important goal for any research for traffic speed prediction a big granularity may make the prediction obsolete as an hourly prediction of traffic speed does not provide utility to anyone. Thus the granularity cannot be too big because it would make the prediction obsolete and it cannot be too small as it would make the problem too hard and lower the performance of the model. In the paper[71] authors found 10-min granularity the most efficient while literature mostly uses 5-minute granularity. This granularity limits the amount of data one day could have. A 10-minute granularity would have 144 values in every day while 5-minute granularity would have 288 values in every day. The second aspect of sample size concerns the number of days in the dataset used to train a dataset. Most commonly used datasets have 3 to 6 months of data while only a few have a year or more data. Even if a dataset had more data the problem of traffic characteristics changing over time means a model can be trained to predict a limited amount of time such as one month, and this training must also be done with a limited amount of data of a few months thus a model is trained with 120 to 180 days of traffic speed data. This means for a dataset that has 5-minute granularity a model is trained with 34.560 to 51.840 data which is considered small for deep learning methods. This lack of data causes models to underperform and it is also the reason why classical machine learning algorithms like Support Vector Machines and statistical models like ARIMA can perform competitively with more capable deep learning algorithms.

The third of these problems is the imbalance of the dataset. Machine learning and Deep Learning algorithms favor balanced datasets and any imbalance in the dataset causes the models to favor some labels more than others. Traffic Dataset is inherently imbalanced as the times where traffic speed is low meaning there is a high amount of traffic much smaller than the times where traffic speed is high meaning there is a low amount of traffic. One such example is the night times when there is small to no amount of traffic meaning nearly 8 hours which is one-third of a day is high traffic speeds. This imbalance causes models trained to predict traffic speed to favor higher speed values which are less important, as prediction errors in these moments do not cause a great amount of difference in travel time(ex. Somewhere that can be reached in an hour with 100 km/h can be reached 75 minutes with 80 km/h) while low-speed values which are more important as prediction errors in these moments cause a great amount of difference in travel time(ex. Somewhere that can be reached in an hour with 30 km/h can be reached in three hours with 10 km/h), predicted less.

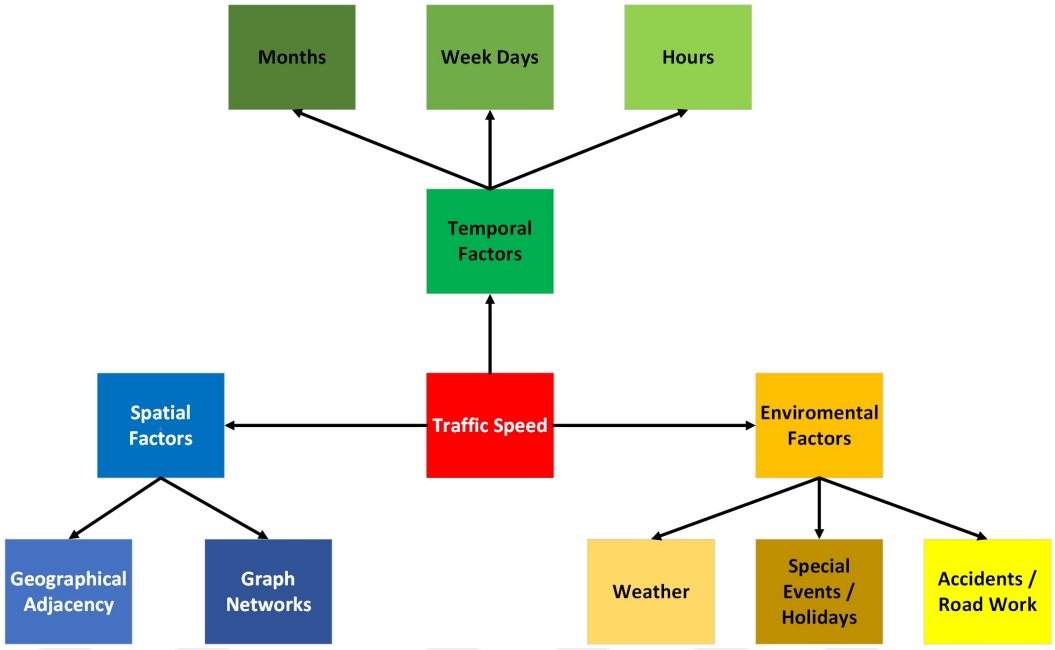


Figure 3.1 General overview of factors effecting traffic speed values.

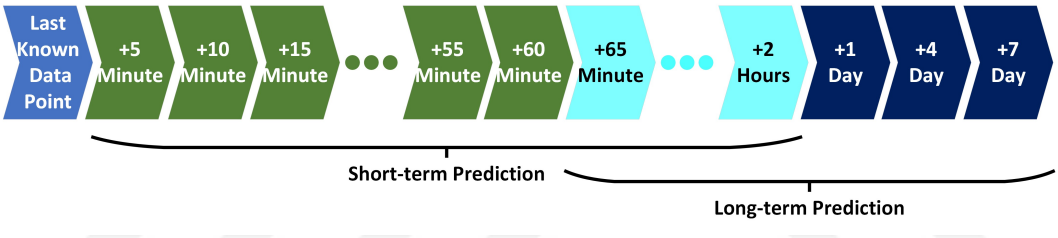


Figure 3.2 Prediction horizon displaying short-term and long-term prediction

In this chapter, we looked at the variables used in prediction and how long- and short-term predictions affect these variables. In Figure 3.1 we show the general overview of factors that effects traffic. We explain these factors in more detail for the rest of this Chapter.

3.1 Factors Effecting Traffic Speed

There are many factors that affect traffic speed values and we categorized these factors into three groups:*Spatial, Temporal, and Environmental* factors. Handling and effectiveness of these factors also change with prediction horizon thus each of these factors described for long- and short-term prediction problem.

In this section we discussed the prediction horizons and their inherent differences then we looked into each of the categories of factors that affect traffic speed.

3.1.1 Prediction Horizon

The traffic speed prediction problem can be summarized as the problem of predicting the speed value of a road segment t time after the latest input speed value. The time value of t determines the prediction horizon of the problem. There are two generally accepted types of prediction horizons that within themselves have different t values. These types are called short-time prediction and long-term prediction which as their name indicates separated by a t value smaller t values are called short-term prediction while bigger t values are called long-term prediction. The problem in this is the threshold of t that separates short- and long-term predictions. In Figure 3.2 we show the prediction horizon of a last known value used for prediction. Green colored t values represents short-term predictions while purple colored t values represents long-term predictions. Turquoise colored t values are disputed in literature where some works include them in short-term predictions while others would include them in long-term predictions.

Even though this conflicting definition and confusion regarding the definition of the prediction horizon these categorizations of model types persist because the methodology regarding the prediction problem changes greatly based on the type of prediction horizon problem. Some differences are regarding the prediction method while others are about the variables and how they are used. The main cause of this difference is what the models learn. In short-term predictions as an example, a 5-minute time frame requires a model to learn a pattern regarding the recent traffic speed predictions and thus it is more reactionary while long-term prediction as an example 1-week ahead, needs to learn a pattern that is generally true because of the length of the prediction horizon. Thus long-term predictions need to learn the traffic characteristics while short-term predictions need to learn the shifts in traffic speed.

In short-term predictions, a model is typically trained to predict a fixed amount of time in the future. This time usually coincides with the data collection frequency for example a dataset that consists of data collected every 5 minutes is used to train a model that predicts 5 minutes in the future. Models trained with this prediction goal use a recursive type of prediction process to predict multi-time step predictions. If we continue the example above the model trained to predict a 5-minute time frame uses this 5-minute prediction to predict 10 minutes and uses both of these predictions to predict 15-minute etc. Although less popular direct prediction, which trains a different model for each of the prediction goals (a different model for 5-minute, 10-minute, and 15-minute prediction), and multi-output prediction which trains one model to predict multiple periods at the same time is also used. These two models are less popular due to the increased need for data which is already

a problem in traffic speed prediction. In long-term traffic speed prediction due to its nature discussed above characteristics are more important than reactions to speed changes. This means that predictions made usually require older data than short-term predictions which means that predictions can be done by one model with real-speed data without using previous predictions. This is also a necessity as the error rate in long-term predictions is greater than in short-term predictions thus a prediction process similar to multi-step prediction used in short-term prediction would carry more errors in each step in long-term predictions than short-term prediction thus decreasing the performance of the model used.

Variables used also differ between these two types of predictions. The main reason is again because of the nature of the types of predictions. As an example, we can take the other road segments' speed value which is a spatial variable. This variable is used quite frequently in short-term speed prediction directly as it contains information that is very useful for this type of prediction. Two roads connected directly means that a traffic jam in the road ahead would indicate lower traffic speed values for the road behind. This shift in traffic speed happens in a short period. Once traffic opens up ahead it also indicates increased traffic speed. While this is very valuable information for short-term traffic which learns reactions more than characteristics for long-term traffic speed prediction it is less so because a traffic jam in the road ahead does not mean a traffic jam will happen next week thus this information does not represent traffic characteristics directly. There are of course ways to use spatial variables in long-term predictions but they are different in the ways they are used and the goals of their use than short-term predictions.

This thesis focuses on long-term predictions. The reason behind this decision is the lack of literature surrounding the long-term speed prediction areas specifically data augmentation regarding long-term traffic speed prediction even though both of these types of predictions are equally important for ITS and smart cities. We intend to remedy this problem by introducing novel techniques to both augment the speed data and also data engineer the dataset for a better training process.

3.1.2 Spatial Factors

Traffic speed data is a spatiotemporal dataset. Spatial factors are related to the location of the road segment that is predicted. Each road segment has its own characteristics and even between segments that follow each other, these characteristics may differ. This has many reasons and dependencies behind the cause is too complicated to produce a general rule. This difference in characteristics is not a big problem for short-term predictions but it means that in long-term

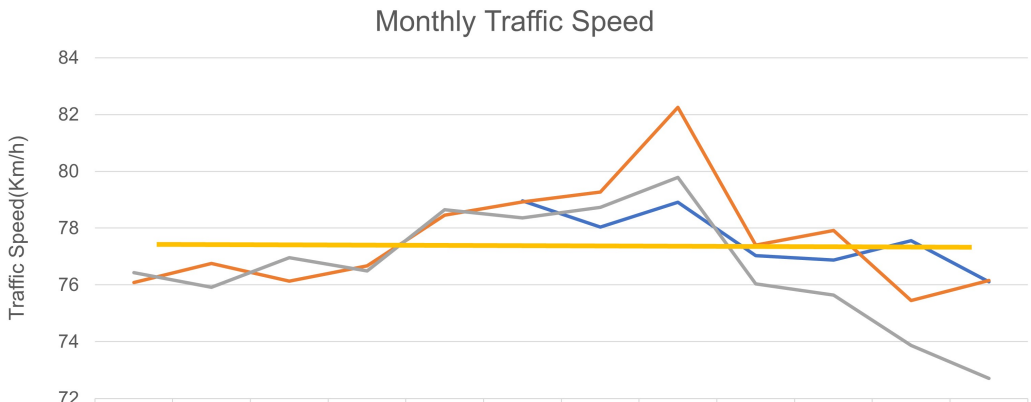
predictions spatial variables cannot be used directly.

In short-term predictions, spatial variables are handled directly, with most researchers taking the roads that come before and after the predicted segment without any other process. This works because the short period of prediction means that any change in the road near the predicted segment will have an immediate effect on the predicted segment. This enables models to react better to the rapid changes that occur in traffic. More recent research focuses on more sophisticated methods such as Graph CNN(GCN) where the spatial location of the road segments is not the only measurement of similarity but also the characteristics that determine the effect of other road segments on the predicted segment.

Long-term predictions also use the sequential road segments with lesser success as road characteristics are different and long-term predictions require solutions based on these characteristics to perform well. As each road segment's characteristics are different using these road segments that come before and after is not effective in all of them. In fact, an optimal solution using direct sequential road segments is using different amounts of sequential road segments for each predicted road segment meaning that some road segments may need 3 front and back road segments to produce the best results while some road segments may need 2,1 or maybe none. But even this type of segment-wise solution is not well suited to long-term speed predictions. A more widely used practice is to quantify the characteristics of segments and use these quantities to find similar roads. This method is more commonly used and in recent years with the development of GCN more effective as the question of how to extract the characteristics of segments becomes irrelevant. Of course, GCNs are not perfect as they require an increased amount of data to extract this complex relationship between road segments. This relationship is not only based on spatial thus the resulting map of dependencies and similarities is more complicated.

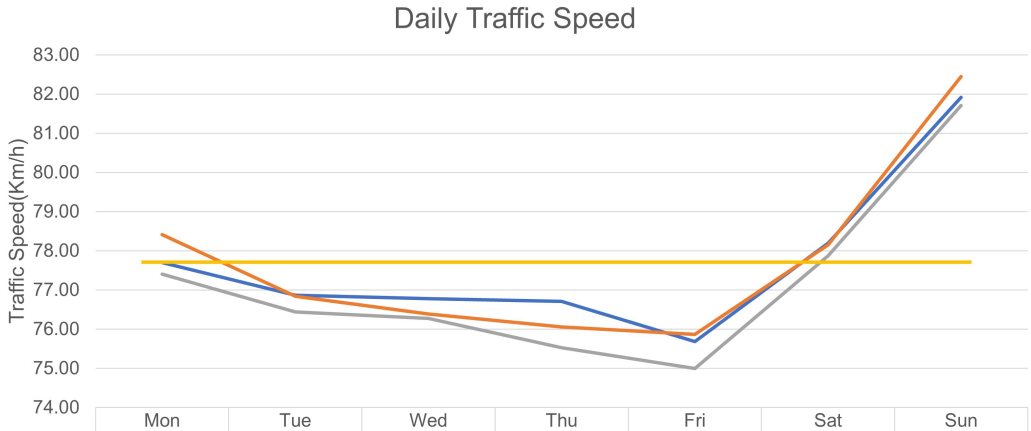
3.1.3 Temporal Factors

Traffic Speed data as mentioned above is a spatiotemporal dataset. Unlike the spatial aspect of the data, the temporal aspect is used similarly by both long- and short-term predictions. One difference between them would be the scope of the temporal aspect as short-term predictions focus on smaller time frames while long-term predictions require wider time windows to correctly learn the characteristics of a road segment. The most important temporal aspect of traffic speed data is the traffic speed data itself. As time flows traffic speed changes and although this change is volatile and complex it shows certain patterns. A prediction model whether short- or long-term



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2017						78.96	78.03	78.91	77.03	76.87	77.56	76.10
2018	76.08	76.75	76.13	76.68	78.46	78.92	79.27	82.26	77.41	77.92	75.45	76.16
2019	76.43	75.92	76.95	76.49	78.64	78.36	78.74	79.78	76.03	75.64	73.86	72.71

Figure 3.3 Monthly average traffic speed changes of 3 years(2017, 2018, 2019) in Istanbul

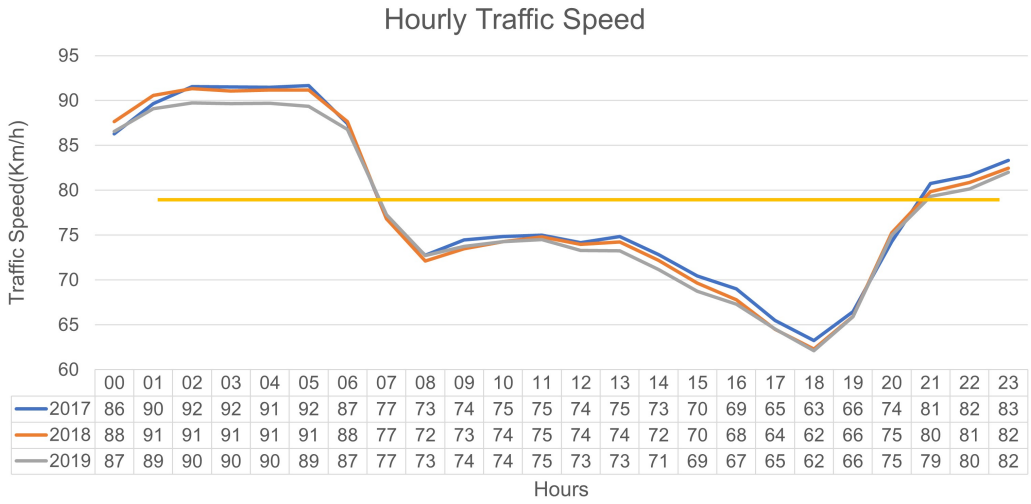


	Mon	Tue	Wed	Thu	Fri	Sat	Sun
2017	77.69	76.87	76.77	76.71	75.68	78.19	81.91
2018	78.41	76.84	76.39	76.05	75.86	78.15	82.45
2019	77.40	76.44	76.27	75.52	74.99	77.87	81.71

Figure 3.4 Average traffic speed changes based on week days for 3 years in Istanbul

learns these patterns for prediction. For short-term prediction, the important part of this change is the closest moments to the prediction while for long-term prediction it is more complex. As the distance between the most recent input and the prediction time increases the importance of the recent data loses its meaning. What is required for long-term predictions is characteristic of the road segment. This information is contained in two parts of the data and it represents two different parts of the training dataset. The first aspect is related to the input and the second aspect is related to the training dataset size.

The traffic dataset consists of volatile data. This volatility means the change in the data from one time to another thus taking one time moment may mislead the model



	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
2017	86	90	92	92	91	92	87	77	73	74	75	75	74	75	73	70	69	65	63	66	74	81	82	83
2018	88	91	91	91	91	91	88	77	72	73	74	75	74	74	72	70	68	64	62	66	75	80	81	82
2019	87	89	90	90	90	89	87	77	73	74	74	75	73	73	71	69	67	65	62	66	75	79	80	82

Figure 3.5 Hourly traffic speed changes of 3 years in Istanbul

the better way is to show a window of time that show before and after the predicted time. This shows the change in the traffic speed to the model and enables the model to see the data in a more stable way. Although change within certain time windows can be volatile we can observe that time characteristics are actually consistent within a day of the week meaning Mondays show consistent traffic speed characteristics and Tuesdays show different characteristics than Mondays yet similar to other Tuesdays. This similarity is given to the model so that it could learn this temporal aspect of the traffic speed dataset. Usually, both of these temporal features are used in input and they decide the shape of the speed values the model receives as input. In Figure 3.3, Figure 3.4 and Figure 3.5 we show the average speed values of main artery roads in Istanbul based on months, weekdays and hours. In these Figures we can see that traffic speed changes with temporal changes but it remains similar within the same period even through years.

Training datasets for traffic speed prediction include many days of speed data. The number of days in the training dataset usually includes more than 1 month's worth of data and most commonly includes somewhere between 2 to 6 months of data. The time period training datasets include different changes to the characteristics of the road segment. This is somewhat due to the change in seasons while some of these changes are caused by changes to the road topography of the city itself. Another important reason for this change is periodic events like months that schools are open which have a higher rate of traffic in specific hours. This temporal shift is hard for models to learn and causes errors to appear in the predictions of models. Long-term models are more affected by this than short-term models as long-term models require more data thus their training datasets include more different characteristics of the same road segment.

3.1.4 Environmental Factors

Traffic speed values are affected by environmental effects like weather and special events drastically and this effect usually disrupts the temporal pattern the road segments have. These environmental variables and the reaction the road segments shows is an important part of traffic characteristics and take an important part of any traffic speed prediction research.

The most important and commonly used environmental factor is the weather. It is commonly known that bad weather conditions affect traffic speed negatively and creates traffic jams at unusual times. While snow has similar effects open and sunny weather has the opposite effect on traffic speed. Less thought of variables like visibility and wind direction and speed has also a high degree of effect on traffic speed. Weather is used in both long- and short-term prediction in a similar fashion and used directly as input in most models.

Another important environmental factor is accidents. Accidents small or big have a negative effect on traffic speed. Although the effect of small accidents with no casualties has less effect for a shorter period, big accidents that have casualties and chain accidents have a bigger impact over longer periods. This variable is very valuable for short-term predictions and many models include this variable while predicting traffic speed [5]. Some research solely focuses on Anomaly detection which includes accidents with short-term prediction methods ([72], [73] etc.). For long-term prediction, accidents are not useable as predicting an accident happening a week before is impossible. Thus, there are two ways to deal with accidents that exist in the dataset. First is to remove the samples that belong to an accident and second is to create a model that is robust to these anomalies. Road work is an environmental variable that has a similar effect to accidents but unlike accidents, they are known before they happen and could be included in the prediction process.

Special events and Holidays are other environmental factors to have a big impact on traffic speed. Specially Holidays have varying effects on the traffic speed based on the length of the holiday. Most people visiting their homes also create a different effect based on the location of the dataset where datasets based on cities have lighter traffic rural areas have more traffic than usual. Both long- and short-term predictions could use special events but due to the little number of days that contain holidays or special events, it is hard for models to learn the special characteristics of traffic on these days.

We have not used environmental factors in our experiments as they are out of this thesis scope though methods described in following chapters can be applied in

conjunction with environmental factors with no conflict.

4
DATASET

All of our experiments were done on traffic speed dataset collected by Istanbul Municipality over the course of 2018 and 2019. In this dataset, there were 441 main road segments, each segment 1 km long on average. We selected 106 Segments among the 441 to lower the training time. In this selection process, we eliminated segments we believed to be problematic like Avrasya tunnel segments which have special circumstances. The rest of the segments were selected by eliminating every other segments in a consecutive order meaning that road segments A B and C that follow each other consecutively we eliminated B and kept A and C. This way we were able to retain different characteristics while lowering the training period.

Traffic speed data were collected at 1-minute intervals. In total, every road segment should contain 525.600 traffic speed data. Some of these data were missing as it is collected by sensor which are prone to errors. To fill some of these missing data and push our prediction problem to more acceptable range[71], we changed the 1-minute intervals to 5-minute intervals. To change the interval value we split the data into 5-minute windows and took the average of the values in this window. After this step we used a moving average with a window size of 6 (which is 30 Minute Past) and filled the remaining missing values as much as we can. Although there remains some missing values, we discarded them for training and testing.

Table 4.1 Dataset and input specifications

Dataset Feature	Feature Value
Year	2018 and 2019
Segment Size	106
Segment Length	1 Kilometer
Interval	5 Minute
Input Shape	7x6(15 Minutes Before and After x 6 Weeks Before Prediction)
Prediction Horizon	1 Week

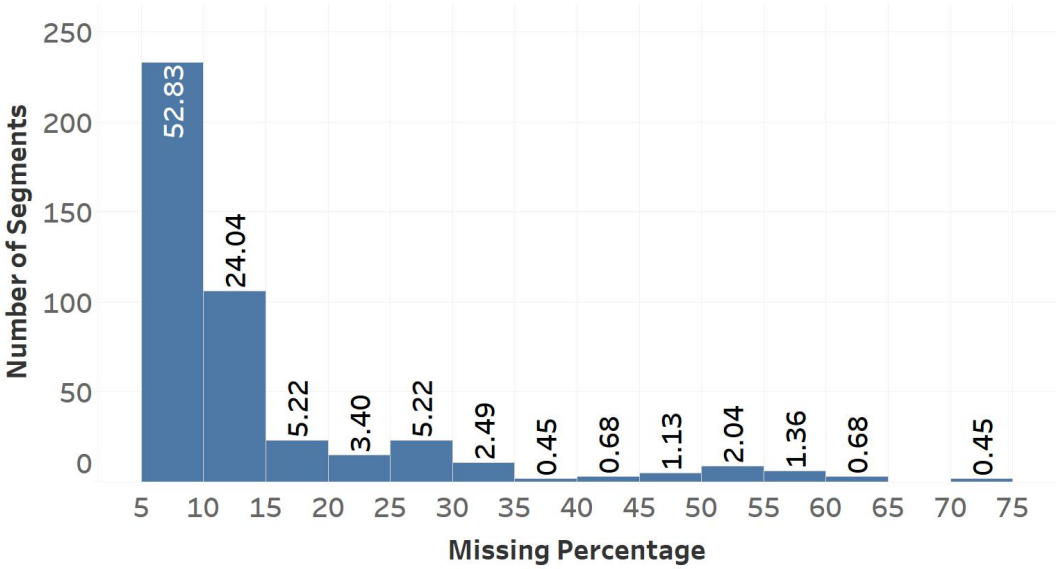


Figure 4.1 Histogram of the missing percentage of 441 road segments. The X-axis shows the percentage of missing data while the Y-axis shows the number of segments. The numbers on the bars shows the percentage of segments that belongs to the bin compared to the total amount of segments.

All the models in proposed were trained to predict a week in future with past data that contains 6 weeks of 15 minute before and after of the predicted moment. Our input only contains traffic speed of the same week day as the traffic characteristics remains similar in the same weekdays for the same hours and minutes.

Specifics of the dataset may change depending on the experiments so each conducted experiment also includes the experimental setup which explains details of the dataset as well. The specifications given in this chapter belongs to the data augmentation experiments.

4.1 Dataset Problems Faced in Long-term Prediction Domain

The traffic speed prediction problem is plagued by many problems some of which are mentioned in the previous sections. Mainly there are three major problems that is present in all of the datasets used in this domain:

1. Missing Values
2. Lack of Data
3. Imbalanced Datasets

In Figure 4.1, we show the histogram of missing value percentages belonging to the 441 road segments. In this histogram, we can see that 52% of all road segments

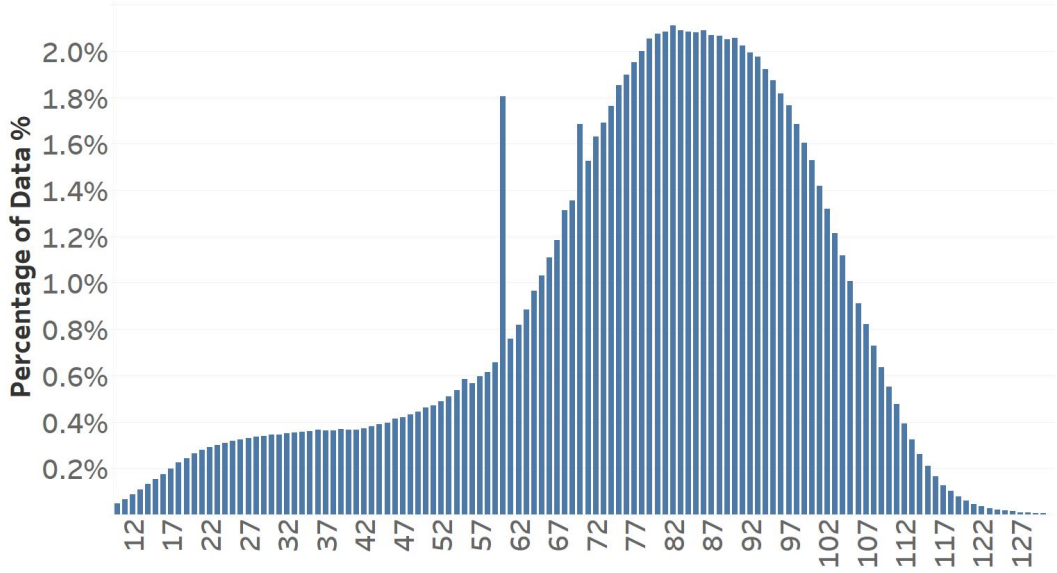


Figure 4.2 Distribution of speed in the dataset.

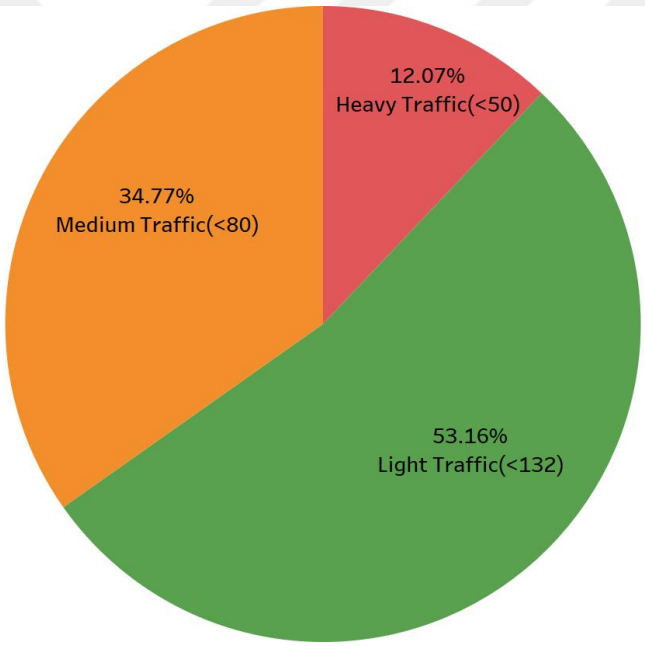


Figure 4.3 Percentage of data in each group

have 5 to 10 percent missing values while only 16% of the road segments have more than 25% missing values. There is only 20 road segments that have more than 50% data missing in our dataset.

Figure 4.2 shows the speed composition of our dataset. The spike in 60 km/h is because of the default value assigned in the speed collection process. In Figure 4.3, we show the percentages of light traffic which is the values higher than 80, medium traffic which is the values betweem 80 and 50, and heavy traffic which is the values lower than 50. In these figures, we can see the imbalance for high-speed values where more than 50% of our dataset is above 80 km/h and only 12% is less than 50

Table 4.2 Dataset sizes of the year 2018

Month	Optimal Size	Min Size	Max Size	Median	Percentage Full
Jan	35136	9291	33900	32138	91
Feb	35424	9555	30084	28749	81
Mar	34560	9274	28647	27302	79
Apr	34848	8587	28961	27867	79
May	34560	13251	29102	28452	82
Jun	34560	20324	33178	32490	94
Jul	35136	23254	34333	33763	96
Aug	35136	24732	34333	33707	95
Sep	35424	23522	34664	34150	96
Oct	35136	22507	34401	33978	96
Nov	35424	24021	34647	34229	96
Dec	35136	27247	34347	33839	96

km/h.

In Table 4.2, we give the sample sizes of our dataset for training each month. These values contain the data of 4 months prior to each tested month. The first column contains the tested months. The second column show the optimal number of samples each training dataset should contain. The third column shows the minimum number of samples found in the training dataset of our selected 106 segment. The fourth column contains the maximum number of samples found in the training dataset of our selected 106 segment. The fifth column shows the median of training dataset sizes while the last column shows the percentage ratio of median and optimal number of samples meaning the fullness of datasets. In this Table we can see that number of data used for training a model can be very low compared to the optimal number of samples and even with datasets close to the optimal number still contain not enough data samples for healthy training process.

All of these problems results in underperforming models so solving these problems would result in an increase in performance. In this thesis we propose novel solutions to all three of these problems that can be applied to any type of model and used with different datasets. Our introduced methods uses traffic characteristics to find patters for solving these 3 problems. For missing value imputation and data augmentation these patterns are found between different road segments while in balancing the dataset we use similar patterns within the segment itself.

5
MISSING VALUE IMPUTATION THROUGH FINDING SIMILAR ROADS

Like many other domains, traffic speed datasets, usually taken from sensors either placed on the roads or in a vehicle that traverses the city, might consist of unreliable or missing data because of sensor failures[74]. Various techniques were introduced to deal with missing values but two main strategies, either replacing them with a similar value or removing them from the dataset, are applied for such cases. Even though removing the missing values enables the most accurate and clean dataset, lowering the amount of data is not a good idea for the current trend of deep learning networks. Following this drawback, more research was focused on replacing the values rather than removing them. Although keeping the instance and replacing the missing value increases the amount of data, finding the most appropriate value is not a trivial problem for traffic speed data imputation. There are mainly three common approaches to fill the gaps between traffic speed data collected from several segments in a metropolitan city.

- Finding a similar day or period of time in the past of the road the missing value is in
- Identifying a similar road and using the value of the time the missing value occurred
- Training a model with a complete dataset and then making a prediction via this model

All three methods have their advantages and disadvantages. Finding similar time frames in the same road segment has the advantage of retaining the same characteristics with the missing values which is an important aspect of missing value imputation. While characteristics are important certain environmental variables have a significant impact on traffic speed which this method cannot replicate. For

example, missing values that exist on a special event may have unusual speed values for a specific time frame that does not affect the traffic speed on any other time frame. In this example, a similar day found in the same road segment data may produce results that deviate from the real values. The second method does not have this disadvantage as any special events would have a similar impact on the other road segments. While this is an advantage, it has the problem of finding road segments that have similar traffic characteristics which is another problem entirely. The third method in theory does not have any of the problems the other methods have because a model trained may learn these environmental variables and characteristics but in reality, it has the problem of data size and may not be able to produce any missing value data where the percentage of missing values are too high to learn road segment characteristics. In this work, we focused on the second method as we believe finding similar traffic characteristics is easier than trying to find appropriate values for missing data considering the environmental factors.

In this Chapter, we describe some methods used in the literature for missing value imputation including clustering, geographical, and k-NN-based methods. We also describe a novel k-NN-based method for Missing Value Imputation. We show the comparison between these methods in the Chapter 8.

5.1 Clustering Similar Segments

Clustering is a non-supervised method for finding similar samples and grouping them together. Missing traffic speed data imputation techniques seek similar traffic flow characteristics to find the most appropriate value for the missing timestamp. As such clustering is used extensively in the literature([8],[75] etc.), we applied multiple clustering methods to find the most effective way to group similar road segments together. In Table 5.1 we give the explanations of the methods used for missing value imputation. In this Table, you can see that we used different types of clustering techniques to test missing value imputation.

One of the benchmark methods we used is to take all of the segments as one cluster and use the entire dataset to find the missing value. Basic clustering was done by creating a vector of traffic speed for each segment using the entire year as a whole. This means that representing a vector is 105.120-speed values which are every 5 minutes for an entire day, for 365 days(12*24*365). Then a basic K-means clustering algorithm was used to create varying number of clusters which is shown in Figure 5.1. Chosen cluster numbers are 2 and 10 to represent the best Silhouette Coefficient obtained, 55,110, and 220 which represents an eight, a quarter, and half

Table 5.1 The proposed techniques for finding similar segments

Methods Name	Method Explanation
All Data	Takes the entire dataset as one cluster.
Basic Clustering (2 Clusters) Basic Clustering (10 Clusters) Basic Clustering (55 Clusters) Basic Clustering (110 Clusters) Basic Clustering (220 Clusters) (Figure 5.1)	Creates a vector for each segment containing all the speed data of a year. Uses k-Means algorithm on these vectors to create X amount of clusters.
Feature Based Clustering (Figure 5.2)	Extracts features that represent a segment rather than using all the speed values. Uses k-Means algorithm on these features for creating clusters
DTW Clustering (Figure 5.3)	Uses Dynamic Time-Warping method for finding distances between speed vectors. Exploits Hierarchical Clustering
Traffic Hour Clustering (Figure 5.4)	Uses only the time period between 06:00 and 22:00 to avoid deceptive night time values which are very similar for the entire city.
DBScan Clustering (Figure 5.5)	Uses DBScan algorithm instead of k-Means algorithm.
LLS[64]	Finds the most similar moment of day comparing all segments data of the entire year
LLS[65] (Figure 5.6)	Finds the most similar day in the segment data
Geographically Closest 2 Segment	Segments that geographically comes before and after the tested segment.
k-NN Method for Finding Similar Road Segments	Suggested novel idea that uses k-NN to find similar road segments

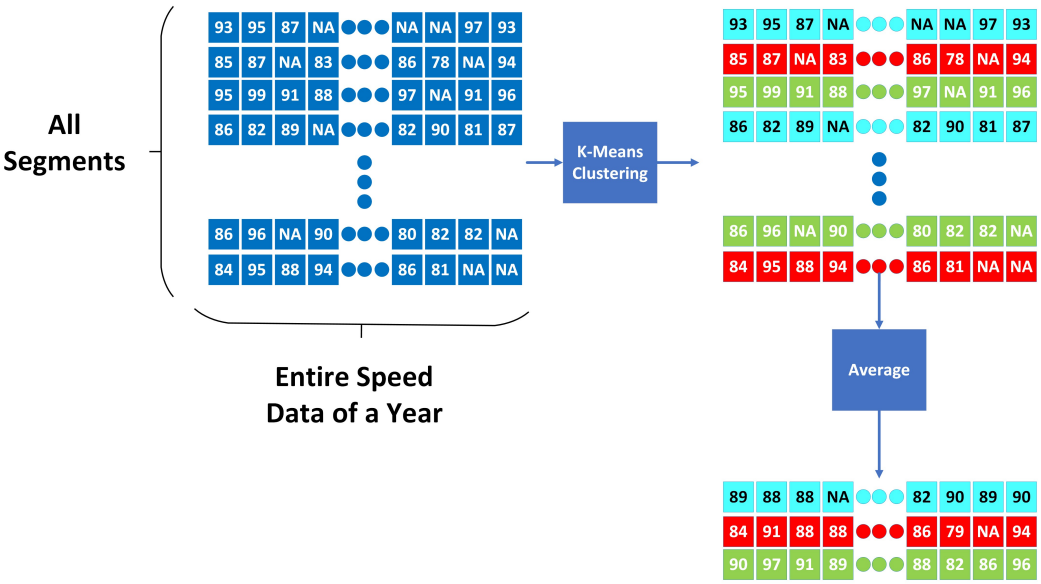


Figure 5.1 Block diagram of Base Clustering method.

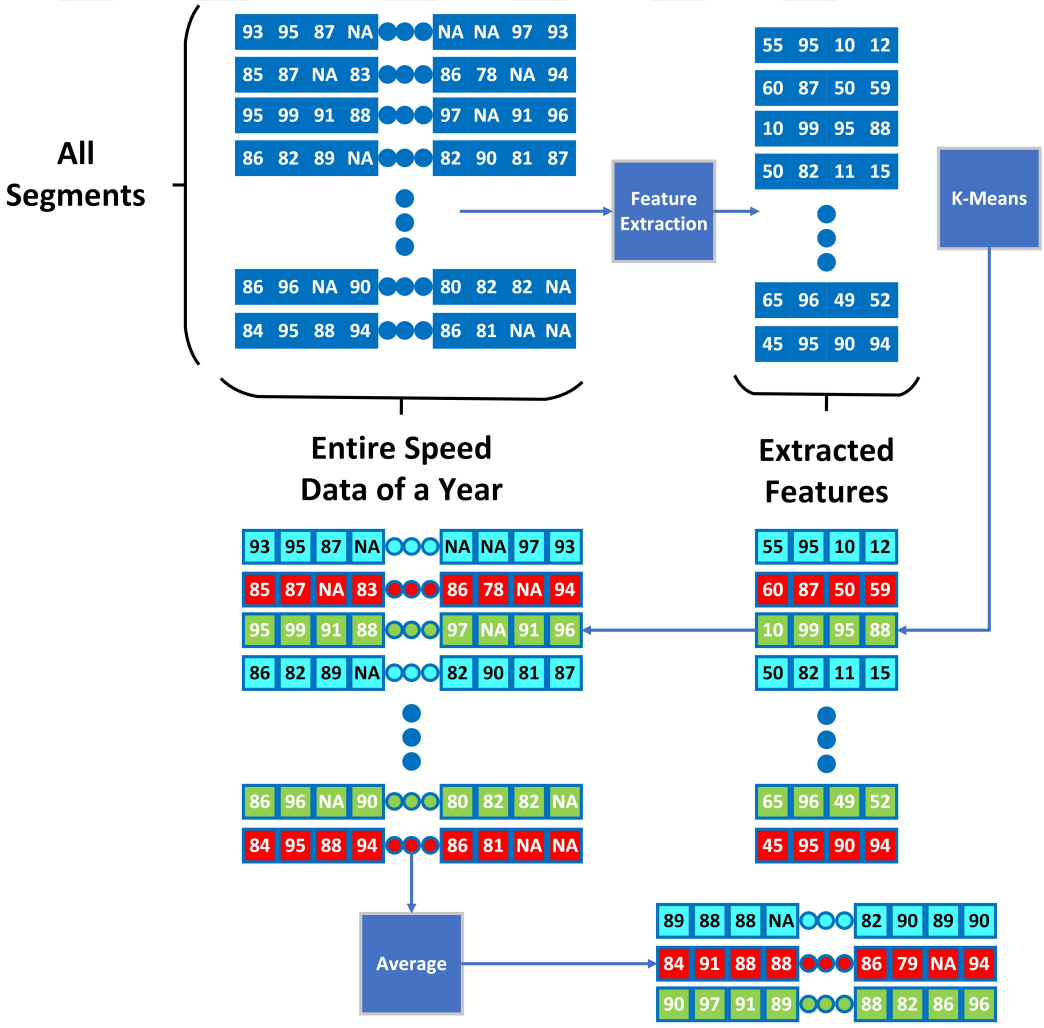


Figure 5.2 Block diagram of Feature Based Clustering method.

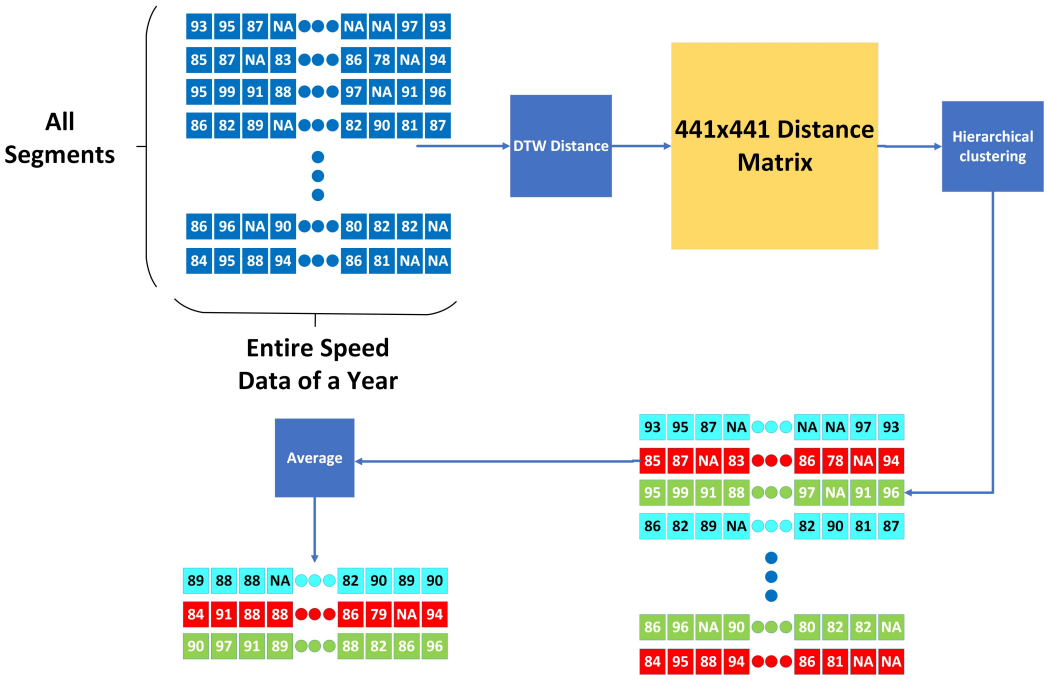


Figure 5.3 Block diagram of Dynamic Time Warping Based Clustering method.

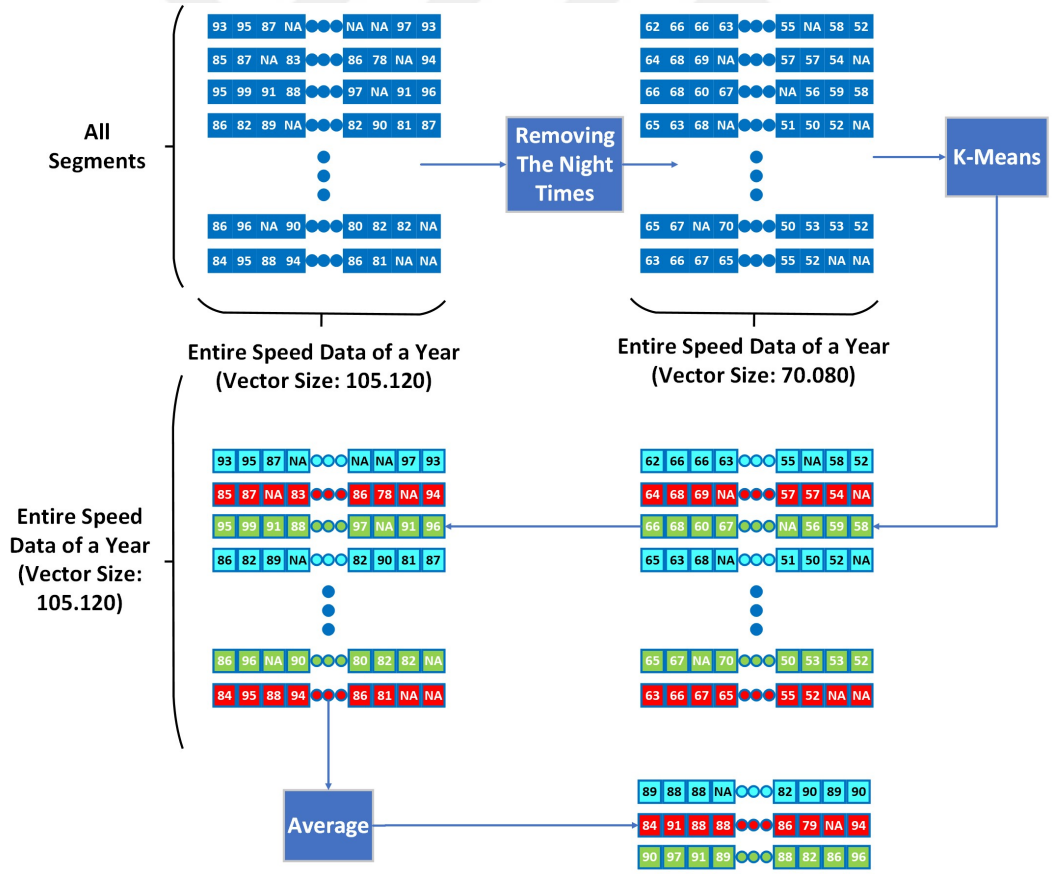


Figure 5.4 Block diagram of Traffic Hours Based Clustering method.

of the number of segments in the dataset that has 441 total segments.

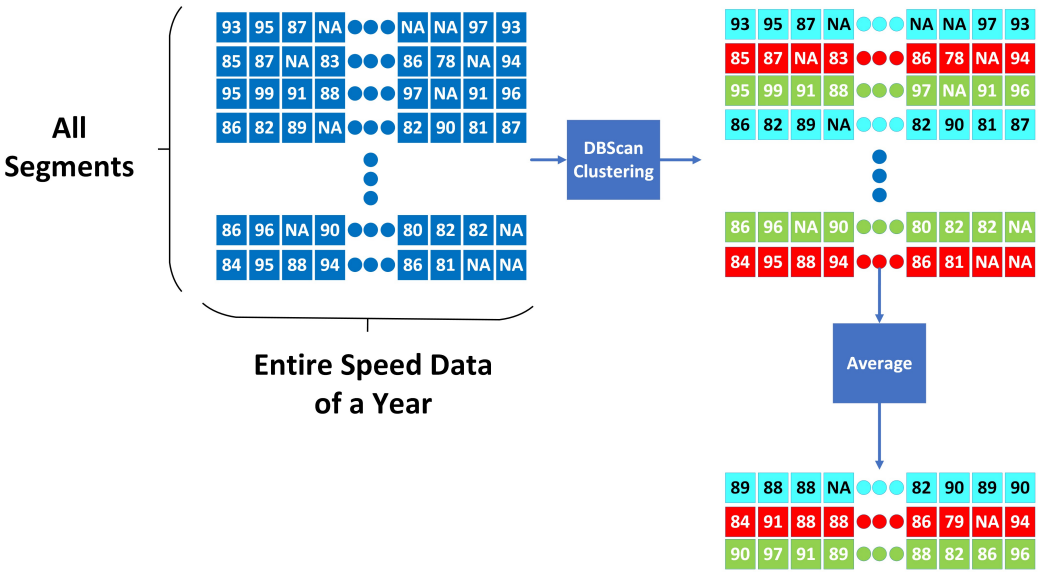


Figure 5.5 Block diagram of DBScan Clustering method.

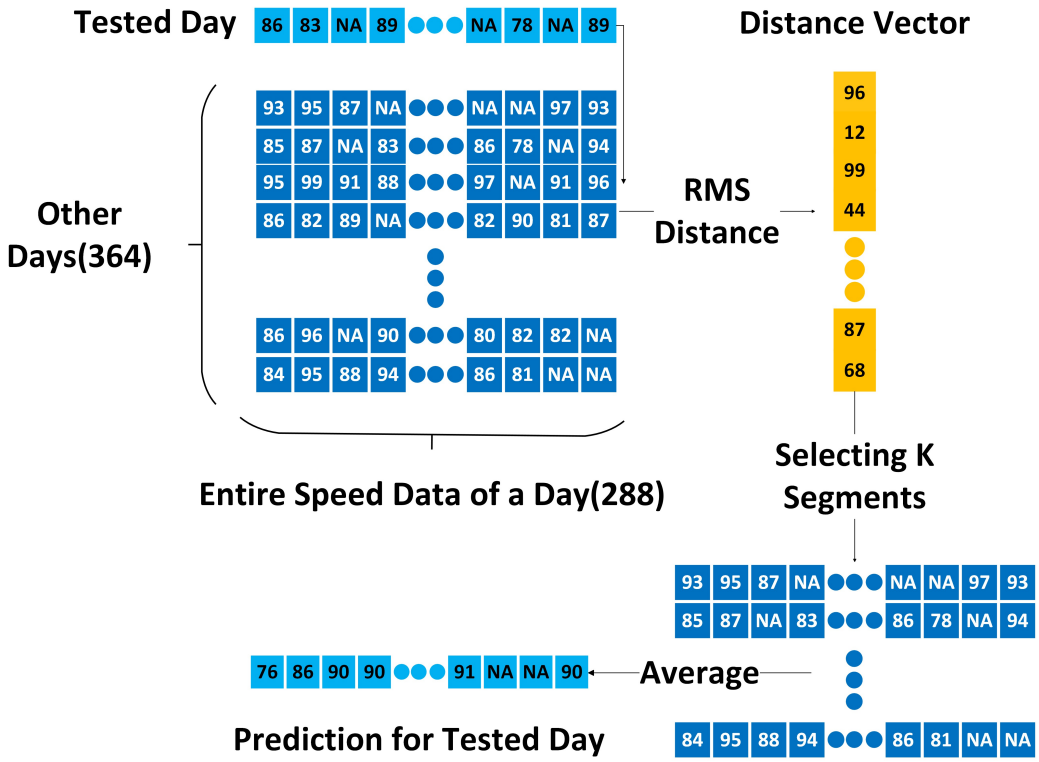


Figure 5.6 Block diagram of LLS Method[65].

Feature-based clustering uses the same traffic speed vector but rather than using the speed values, features extracted from this vector were used instead. Features used in this method were selected based on the variance they had on the road segments meaning that features that did not have high enough variance were not selected. This

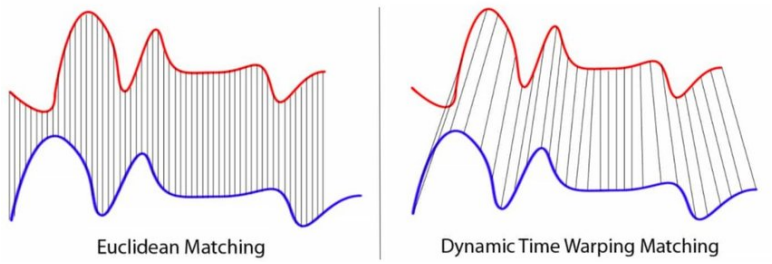


Figure 5.7 [76]Difference between Dynamic Time Warping and Euclidian Distance. In comparison between two-time series data, there might be shifts that render Euclidian distance inefficient. DTW takes this type of shift into consideration when comparing two time-series data.

eliminated extracted features that remained the same or similar below a threshold. Features that could not be calculated for most segments were also eliminated. Pearson Correlation between these features was then calculated and segments that have high correlation values were also eliminated. The remaining feature was then used as a representation of road segments and the k-means algorithm was used for finding clusters. In Figure 5.2 we show the process of feature-based clustering step by step.

In DTW clustering rather than Euclidian distance Dynamic Time Warping(DTW) was used for measuring the distance between road segments. The advantage of DTW can be seen in Figure 5.7 where DTW and Euclidian distance methods are visualized. In the Figure, we can see that changes in time-series data that naturally occurs like shifts in time are not considered in Euclidian distance. This means that using Euclidian distance would only give the correct distance if two time-series data were aligned before comparison. This is not always possible these shifts could happen in different values. DTW takes this into consideration and uses a dynamic algorithm to find time matching between two time series. This enables DTW to match values that are in different time periods with each other independent of the values in between. After finding the distances between two road segments, hierarchical clustering is used for finding similar road segments. In Figure 5.3 we show the process of DTW-based clustering.

Traffic hours are similar to the basic clustering but rather than the entire day only hours between 06:00 and 22:00 were used. This was done to avoid using nighttime speed values which are similar to each other and confuse the clustering methods. The resulting vectors which are 66% of the original vectors then clustered using the k-Means algorithm. The resulting clustering information is then used with the original vectors that include nighttime speed values to create missing data imputation values. In Figure 5.4 we show the process of the traffic hours-based

clustering method.

DBScan algorithm is also similar to Basic Clustering but rather than K-means it uses DBScan Algorithm. In Figure 5.5 we show this process which is similar to the base clustering method.

Least Local Squares(LLS) are a k-NN variant that was used in literature for missing value imputation. One of the papers that used LLS[64] found similar moments in a day by comparing speed values corresponding to day time frames of all segments in a year. The authors of this paper build a matrix that had 288 rows because of a data collection interval of 5 minutes. The rows represent the amount of data in a day. Columns were made of every day of every segment meaning that a dataset that contains a year of 100 segments would have 365*100 columns. Comparisons were done between rows of this matrix. The second paper[65] used LLS to find similar days in a road segment for missing value imputation. The methodology used in the second paper is shown in Figure 5.6 where the tested day is compared with the rest of 364 days of the same segment and then similar days are used for finding the imputation values.

Geographically Closest segments represent the road segments that are sequential geographically meaning they are the roads that come before and after the road segment where missing values exist. For segments that has no roads before or after, only the road that existed were used.

5.2 Proposed Missing Value Imputation Method

For missing value imputation we propose a novel technique that uses a variant of k-NN to find similar road segments. To find more specific segments that are similar to the predicted segment, we decided to exploit the philosophy of the k-NN algorithm for each cluster and only use the average speed of the k-closest road segments in that cluster for prediction. We used 2018 data to find similar segments and then made tests with traffic speeds of 2019 for prediction so the results would be fair and reliable.

In Equation (5.1), we give the formula of finding k closest segments. By determining k closest segments, we calculate the distance of traffic speed values between the respective segment and all other remaining segments in the cluster (5.1b). To calculate the distance between two vectors consisting of the speed data of a whole year with a 5-minute resolution, 105.120 (365*288) values are computed via Root-Mean-Square Distance (RMSD) (5.1a). Afterward, we normalize the

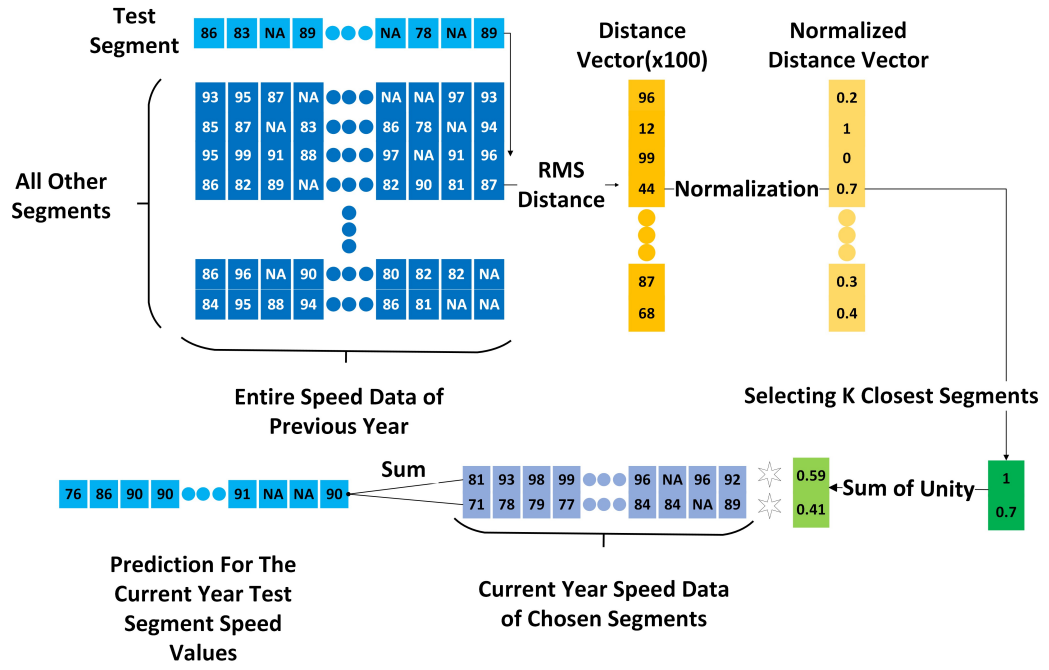


Figure 5.8 The block diagram of the proposed methodology for the detection of k similar segments to a given segment and prediction of the corresponding timestamp with a missing value

distances into 0 and 1. To make certain calculations later easier, we subtract these normalized values from 1 so that greater distances get closer to 0 (5.1c). Then, we select k segments closest to 1.

To create values for replacing missing data in 2019, we calculated distances using 2018 data of 441 segments. We normalized the distance values of selected k segments so that their sum is equal to 1 using Equation (5.2b). Afterward, we multiply the speed vectors of selected segments by the normalized distance values and sum them to find the replacement values (5.2c). Finally, we calculate the MAPE values using predicted values and real values. In Figure 5.8, we demonstrate the general outline of the proposed methodology.

$$RMSD = \sqrt{\sum_{k=1}^{12}\sum_{j=1}^{31}\sum_{i=1}^{288}\frac{(\hat{y_{kji}} - y_{kji})^2}{365 * 288}} \tag{5.1a}$$

$$\mathcal{D} = \sum_{i=0}^{N-1} RMSD(\mathcal{X}, x_i) \tag{5.1b}$$

$$\overline{\mathcal{D}} = \sum_{i=0}^{N-1} 1 - \frac{(\mathcal{D}_i - \min \mathcal{D})}{(\max \mathcal{D} - \min \mathcal{D})} \tag{5.1c}$$

Table 5.2 Mape values for different numbers of k similar segments

Number of Similar Segments (k)	MAPE(%)
1	11.81
2	**10.99**
3	11.49
5	12.66
10	14.66

$\hat{y}$ = Predicted value
y = Real value
N = Number of segments in the dataset
$\mathcal{X}$ = Vector of test segment
x_i = Vector of i$^{\text{th}}$ segment in the dataset
$\mathcal{D}$ = Vector of dist. b/w tested and other segments
$\overline{\mathcal{D}}$ = Normalized distance vector

We also examined the optimal value by running experiments for parameter k from 1 to 10. We completed our tests just for One cluster setting to use the entire dataset while eliminating the risk of not having a cluster with less number of segments than k. Taking the weighted average of the distance values obtained from 2018 data has given better results than regular averaging. While the regular averaging had a MAPE value of 11.08, the weighted average resulted in 10.99 while using the entire dataset to look for similar segments. The number of similar segments (k) was examined by using the weighted average method. The results in Table 5.2 were obtained via weighted average from only One Cluster, i.e. all data. The weights were calculated by the distances of k similar segments.

$$MAPE = 100 * \frac{1}{n} \sum_{i=0}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \tag{5.2a}$$

$$\overline{\mathcal{D}}_s = \sum_{i=0}^{k} \frac{\overline{\mathcal{D}}_i}{Sum(\overline{\mathcal{D}})} \tag{5.2b}$$

$$\bar{\mathcal{X}} = Sum(\overline{\mathcal{D}}_s * x_s) \tag{5.2c}$$

$$MAPE_{\text{Error}} = MAPE(\bar{\mathcal{X}}, \mathcal{X}) \tag{5.2d}$$

$\hat{y}$ = Predicted value
y = Real value
$\overline{\mathcal{D}}_i$ = i$^{\text{th}}$ dist. among k selected segments
$\overline{\mathcal{D}}_s$ = Normalized dist. of selected segments
$\bar{\mathcal{X}}$ = Wght. avg. of k segments
x_s = Vectors of selected segments
$MAPE_{\text{Error}}$ = Difference between pred. and real v.

MAPE values get worse with the increase in the number of similar segments. We obtained the best results when k is equal to 2. A decrease in success when k is equal to 1 shows us that having more than one segment for prediction generalized the results enough for the entire year.

In Table 5.3, We give the MAPE scores of the methods we experimented with for the missing value imputation problem. These scores represent the MAPE values of real speed values of the year 2019 and speed data values produced using the methods described earlier. The results shown in Table 5.3 are obtained from 441 main roads of Istanbul. The dataset details are described in Chapter 4.

In this Table, the first column is the names of the methods while the second column is the number of clusters produced with these methods. For the methods that do not use clusters like LLS methods and Geographical methods, this column shows how many clusters are used for producing the imputation values. For LLS this column shows the value of k while in the geographical method case, it shows 2 which means one back segment and one forward segment. The third column shows the value of segment errors which is the number of segments in the methods that were unable to produce imputation values. This is due to no other segment existing in the same cluster the tested segment is in. For example, the 220 Base cluster has 105 segment errors which means that 105 segments out of 441 are in a cluster that has only one member in them. For the Geographical method cluster error is because there is only one segment near them meaning that they are the beginning or end of the road network. The fourth column shows the MAPE values obtained by using the entire cluster segments to produce imputation values. For example, the all-data method which takes the entire dataset as one cluster takes the average of 440 segments (441 total segments, one segment is a test segment which is excluded) to produce imputation values. For methods of LLS, this column is empty because LLS uses k similar segments for each segment. In the fifth column, MAPE scores are obtained by finding k similar segments within each cluster and then using the distance of two segments' speed values as weights for weighted averages. These averages are the imputation values and the MAPE score is found between imputation values and real

Table 5.3 Mape values and number of clusters for the proposed clustering techniques

Methods	Number of Clusters	Segment Error*	MAPE(%) Normal Average	MAPE(%) Weighted Average
All Data (No cluster)	1	0	28.67	**9.26****
B. Clustering (2 Clusters)	2	0	24.89	11.45
B. Clustering (10 Clusters)	10	0	21.87	13.11
B. Clustering (55 Clusters)	55	2	14,93	11.98
B. Clustering (110 Clusters)	110	26	13	11.2
B. Clustering (220 Clusters)	220	105	**10.61**	10.21
Feature Based Clustering	2	0	25.2	11.44
DTW Clustering	16	0	24.91	11.98
Traffic Hour Clustering	4	0	27	11.3
DBScan Clustering	7	1	26	11.83
Geographically Closest 2 Segment	2	2	12.37	13.84
LLS[64]	30***			17.77
LLS[65]	45***			14.1

* The number of segments in which the clustering method was unable to produce predictions due to an insufficient number of segments within the cluster.
** Results obtained from the segments which Basic Clustering (220 Clusters) method has managed to create a prediction for.
*** Number of closest neighbors (i.e. k)

speed values.

Since the number of predicted segments decreases with the increase in the number of clusters, we evaluated only the MAPE results of 336 segments for each case. As the options, i.e. available segments, in All Data are more than clustered data, the weighted MAPE for All Data produced the lowest value of 9.26 which is better than 10.21. These results show us that our method of finding the k similar segment gives better results than taking the average of every segment. Thus, using the principle of the k-NN algorithm on All Data has outperformed any clustering technique we experimented on.

6
DATA AUGMENTATION THROUGH FINDING SIMILAR ROAD SEGMENTS

Traffic speed characteristics although unique for each road carry similar patterns depending on different aspects of traffic. The goal of finding these similar patterns is to use different speed data for different tasks like data augmentation or missing data imputation. There are two ways to find similar traffic speed data that exhibit similar traffic characteristics which differs by which of the data similarity is searched through. The first way is to look for similarities in the same segment and the second way is to look for similarities between different segments.

There are two major advantages to using different road segments: the first one is the amount of data available to use which is limited to different methodologies that are used for data augmentation and missing data imputation tasks. The second advantage is replication of the environmental variables which are hard to account for. The segments that show similar traffic characteristics are bound to respond to environmental variables like weather, holidays, and special events in a similar fashion. This saves the need to adjust the speed variable or try to match these different variables with other time periods. The disadvantage of this method is that there needs to be a reliable method to find similar traffic characteristics between road segments. Although looking through the same segment has the advantage of having already similar characteristics even then these characteristics change depending on month or time of day which means even within the same segment there needs to be a process to select similar data windows.

In this chapter, we introduced the methods we have applied to find similar traffic characteristics among different road segments. We begin with the clustering techniques we have used which were discussed in the Chapter 4 under the missing value imputation section and some problems we have faced. Afterward, we looked at the proposed methodology in detail and some results we have obtained. More detailed results for our methodology are included in the Chapter 8.

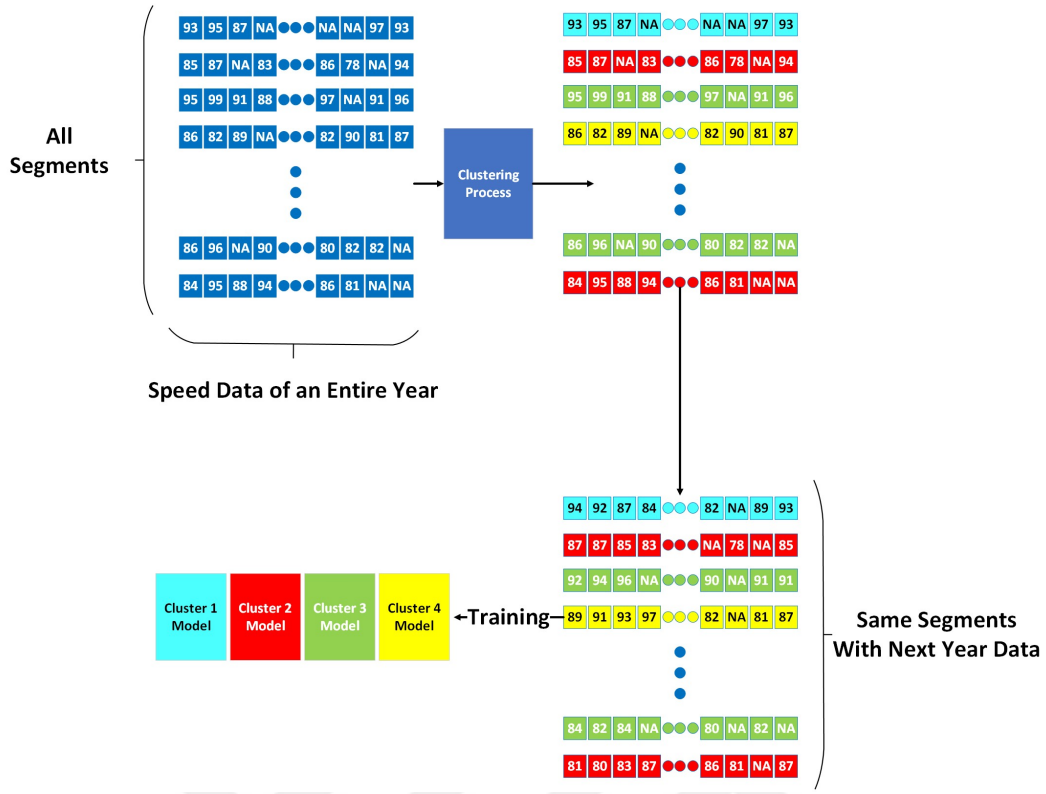


Figure 6.1 Block diagram of the clustering process. The clustering method in this diagram changes through different methodologies used but the general process remains the same with data clustered and a model is trained for each cluster separately using the data of segments belonging to the cluster.

6.1 Similar Road Detection Through Clustering Methods

Unsupervised learning is a machine learning algorithm type that has no initial label data for using or predicting. This type of machine learning algorithm finds the similarities between data samples and creates groups or clusters for its main purpose. Unsupervised learning is used in many different areas that require finding similarities as such for the purpose of finding similar traffic characteristics it is frequently used.

For the purpose of data augmentation with traffic speed datasets, we used multiple clustering methods to create clusters that contain similar road segments. The general process of our methodology is given in Figure 6.1. In this Figure, we can see that the data used for clustering is the entire year of speed data for each segment. We experimented with different time intervals such as 5-minute, 15-minute, and 60-minute to see their effect on clustering results but found no major difference between intervals thus we used 5-minute intervals because it is the original intervals of our dataset. Each vector represents the road segment and contains its traffic characteristics. We then applied clustering methods discussed in this section, and

create clusters that contain similar road segments. In the Figure, we have found 4 clusters for our dataset. We used this clustering information to train models for each cluster using the speed data of all the segments that are in the cluster model trained. Data used in the training process is different from the data used in the clustering process to prevent data leaks between the two processes. Data used in the clustering process is one year prior to the data used in the training process. For each cluster, we created 12 models for each month of the year, and for these models, we used 4 months prior to the month the model trained to predict.

6.1.1 Clustering Methods Utilized

Clustering methods utilized for finding similar road segments are the same methods given in Table 5.1. To summarize:

- **Base Clustering:** No additional process is applied to the dataset while the k-Means clustering method is used.
- **DBScan Clustering:** Same with base clustering but the DBScan algorithm is used instead of the k-Means algorithm for clustering.
- **Feature Extraction Clustering:** Statistical features are extracted from the data before the clustering process is applied. Extracted features are used instead of speed data.
- **Dynamic Time Warping Clustering:** DTW measure is used instead of RMS distance to find the distances between segments. Afterward, these distances were used with the hierarchical clustering method to create clusters.
- **Traffic Hours Based Clustering:** Night-time speed values are similar in all of the road segments with low traffic and high-speed values which makes it harder for clustering algorithms to differentiate between road segments. So we tried taking these values out and clustering with the speed values between 06:00 AM and 10:00 PM.

We also used the proposed method of the k-NN variant for finding similar road segments to see its performance in data augmentation rather than missing value imputation.

6.1.2 Problems Faced in Clustering Traffic Speed Data

Clustering success is measured with different methods. These measurements are generally used for finding the optimal cluster number and they show the success of

separating different characteristics of the samples. The most basic measurement uses simple distance to the center of the cluster while more sophisticated measurements use other cluster information such as the closest sample that belongs to other clusters or other cluster centers to not just measure the cohesion of the cluster within but also the separation of samples that carries different characteristics from each other.

Traffic speed data carries different characteristics not only in different road segments but also through different time periods such as different seasons or different hours of day. This complicates the clustering process as clustering algorithms perform best when there are common characteristics between samples while traffic speed data is composed of different characteristics and like a DNA sample it is close to being unique. One solution to this problem is to limit the time period of segment data so that the changing characteristics are separated but this is rather finding similar time periods rather than similar road segments and although it has the advantage of finding similarities more easily it has the disadvantages of being susceptible to environmental factors, having to decide the length of the time period as most characteristics are displayed in different time lengths, needing more complex structures for data augmentation because creating input and output for model training requires more than the time period decided for example a model that uses previous 6 weeks of speed data as input would need that previous data characteristics and its relationship to the label data to match but a methodology that finds similar days would lack the ability to capture these characteristics and fail to find similar samples for data augmentation.

$$\mathcal{S}(i) = \frac{b(i) - a(i)}{max(b(i), a(i))} \tag{6.1a}$$

i = Data point tested
$b(i)$ = Average Dissimilarity to the data points the next-nearest cluster
$a(i)$ = Average Dissimilarity to the data points within the same cluster

This problem of road segments carrying different characteristics that makes it too complex for clustering algorithms to capture and create clusters that contain segments with similar characteristics can easily be observed with the Silhouette score. Silhouette score is a measurement of a cluster's cohesion and separation between themselves. The formula of this coefficient is given in Equation 6.1. It contains not just the sample's distance to the cluster center they belong to but also

considers its distance to the closest cluster center which measures the separation of clusters. The closer this score gets to 1 better clusters the method has created.

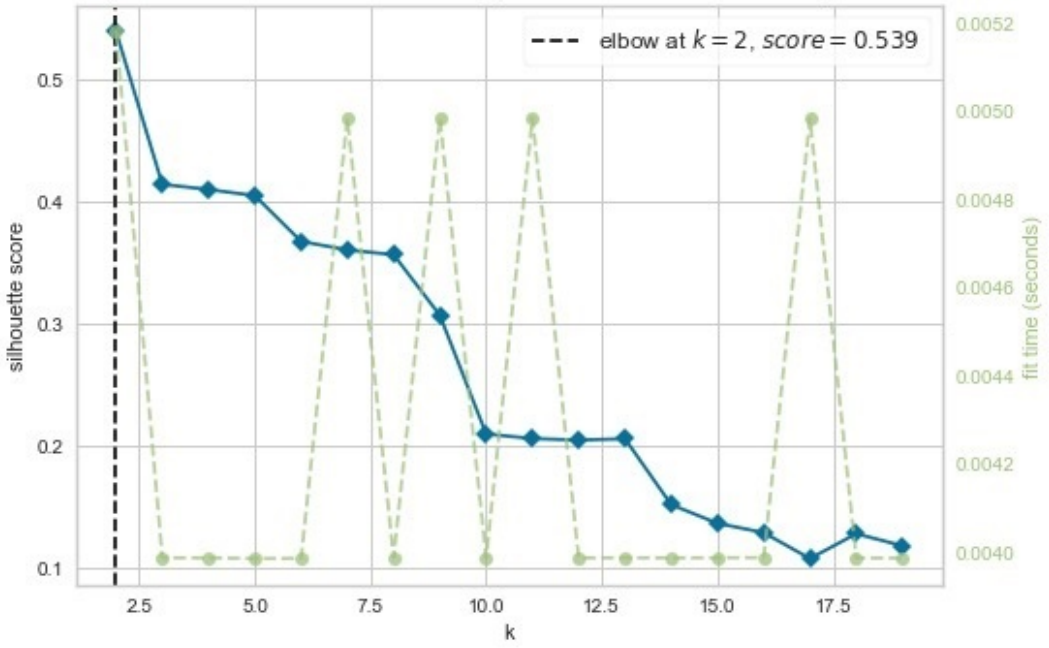


Figure 6.2 Silhouette score graphic obtained with base clustering methodology with the different number of clusters.

Figure 6.2 shows the silhouette score graphic we obtained using the base cluster method with the different number of clusters. In the Figure, it can be seen that the silhouette score decreases over time as opposed to increasing like the silhouette score is supposed to do. As the number of clusters increases in theory the separation between data points also becomes better thus increasing in silhouette score from the formula we can deduce that when each sample is separated into different clusters silhouette score becomes 1. In the Figure 6.2, we can see that is not the case with our data. The reason behind this divergence from normality is because of the similarity each segment has due to having complex traffic characteristics. Data points in our dataset which are the vectors of traffic speed data that represent each road segment have too many similarities with each other so any separation lowers the silhouette score as it measures similarities between clusters and decreases when two clusters are too similar to each other.

To remedy this problem we attemted to use the traffic hour clustering method in which we remove the nighttime traffic speed values which are too similar to each other. In Figure 6.3 we can see the results although still not how the silhouette score is supposed to be is different from the initial graphic obtained. In this Figure, we can see that cluster number four has a massive increase in silhouette score which

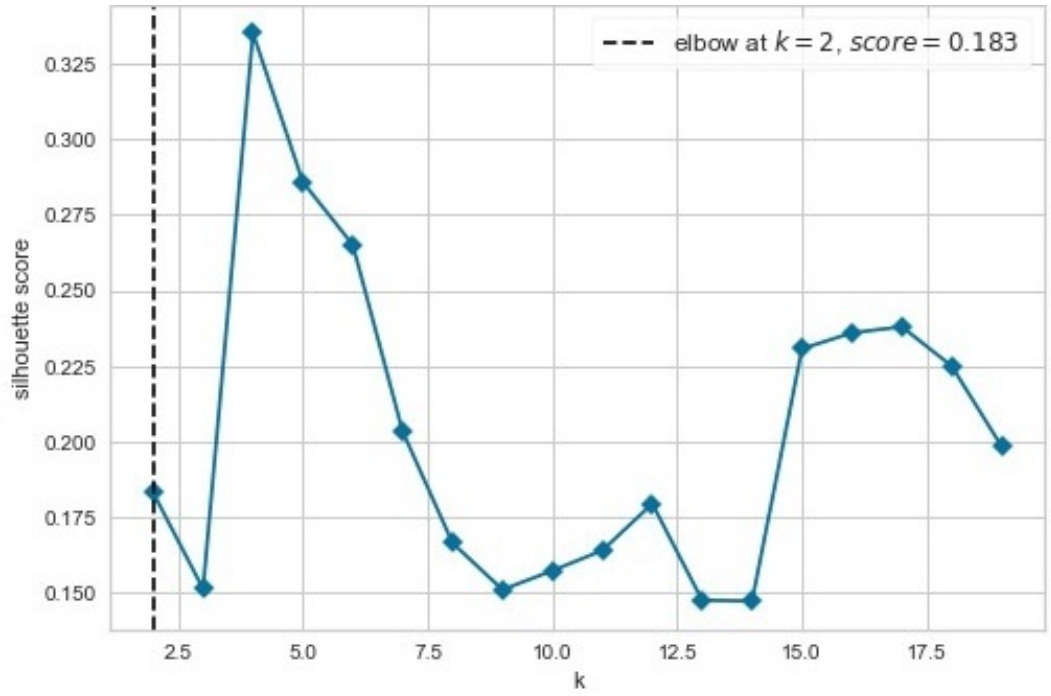


Figure 6.3 Silhouette score graphic obtained with traffic hours clustering methodology with the different number of clusters.

shows that similarities between data points cause the silhouette score to not be able to asses clustering success.

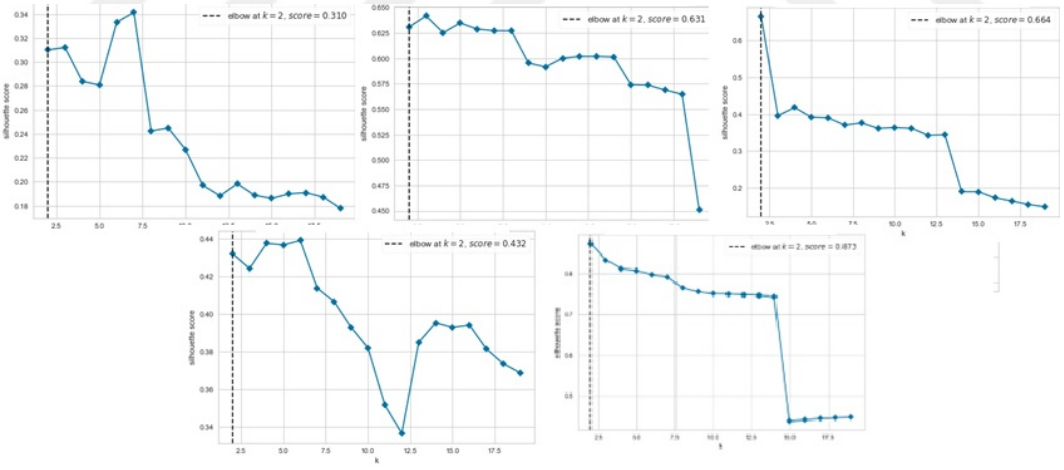


Figure 6.4 Silhouette score graphic obtained with model creation based on time of day methodology with the different number of clusters. The Figure shows from the top left to staring left in each row, morning, noon, afternoon, evening, and night

After the initial success obtained with traffic hours-based clustering, we decided to use the method that will be described in Chapter 7 which separates the speed values based on time of day to isolate the traffic characteristics existing in these time periods. We decided that it would be a solution to our problem of similar characteristics existing between data points. For this method, we split each segment

Table 6.1 Number of clusters created for each category based on the hour of day

Category Name	Cluster Count
Morning	6
Noon	3
Afternoon	2
Evening	2
Night	1

into 5 different categories based on hours speed data is recorded and clustered each category separately. Afterward, a model was trained for each cluster that is created from each category separately meaning that if the morning category has 4 clusters created and the noon category has 3 clusters created 7 models would be trained for the morning and noon categories. In Figure 6.4 we show the silhouette scores obtained with each category. In the Figure, we can see that for some categories afternoon and night, the problem has not changed much but for the rest, there is an improvement which shows better clustering. In Table 6.1 we show the number of clusters created for each category.

Of course, the silhouette score is not the final error measurement as the data augmentation is done for increasing the performance of traffic speed prediction and its improvement is the determining factor for a method's success.

6.2 Similar Road Detection Through Prediction Based k-NN Method

Clustering is not necessary for unsupervised learning methods as methods like k-NN although an unsupervised learning method does not create clusters rather it finds similarities for each sample directly. Our experience in missing value imputation showed us that similarities between two segments do not guarantee similarities between other segments. For example for three segments A, B, and C if a similarity exists between A and B and B and C that does not guarantee the similarity between A and C. This means that for the purpose of data augmentation, clustering techniques for some segments give worse results but the k-NN type of method generally has a positive impact on performance. Therefore, we have focused on k-NN for finding similar road segments. We believe that for the subject of traffic speed prediction, this type of method that finds solutions for each segment separately is a better fit.

We have used a k-NN variant for missing value imputation that used the traffic

speed values of an entire year for each road segment to represent them. We applied a weighted average based on the distance for finding the final value to fill the missing parts in the dataset. With this method, we have obtained better results than other methods we have applied for this problem including clustering methods, other k-NN methods, and geographical methods. The results obtained were the results of missing values filled and not data augmentation which is different from each other. The difference is the data that is added to the dataset with missing value imputation the data added is the missing values that belong to the data while data augmentation adds data that does not exist within the dataset. This means that values filled with missing value imputation exist within the traffic characteristic the road segment has and fits the relation that prediction models use while for data augmentation this relationship must be found specifically other wise data augmented would confuse the model trained for prediction. This is why the method we used for missing values loses its effectiveness in data augmentation. The method we used for missing values looks at the distance between two-speed vectors which only measures the closeness of two segments' speed values while for data augmentation the relation between past weeks' data to current value is more important as it is this relationship that models learn for making predictions.

This relationship between past and present speed values is hard to capture with distance measures but vital for the data augmentation method. As this relationship is the goal of prediction models we propose a method for using prediction models for finding similar road segments. As this method uses deep learning methods to capture the complex relationship between past and present speed values we can use the learned patterns to find road segments that carry similar traffic characteristics. Our goal is to increase the performance of long-term traffic speed predictions thus the models we use for finding similar characteristics are trained to learn the patterns that exist in long-term traffic speed prediction.

In Figure 6.5 we give general methodology on an example of 4 segments. The first step of this methodology is to train models based on the 4 segments' data. For each segment, 12 models will be trained for each month so in this example, 48 total models will be trained using 4 segments past 4 months. The data used in this step is the data belonging to the previous year of the year data augmentation will be applied. For example, if the data augmentation will be applied to the year 2019 then this step will use the data of 2018. Models trained for each segment are exclusive to each other meaning that any model trained in this step will only use the data belonging to the same segment. This training step is shown with red straight arrows in Figure 6.5. The next step is to find the distance between each segment using the models trained in the previous step. In this step, we use the data belonging to the

other segments as a test dataset for each model to predict and evaluate the results with MAPE score. For example, in the Figure, we can see 4 models each numbered for the segment the model trained for. Using these models we test the rest of the segments

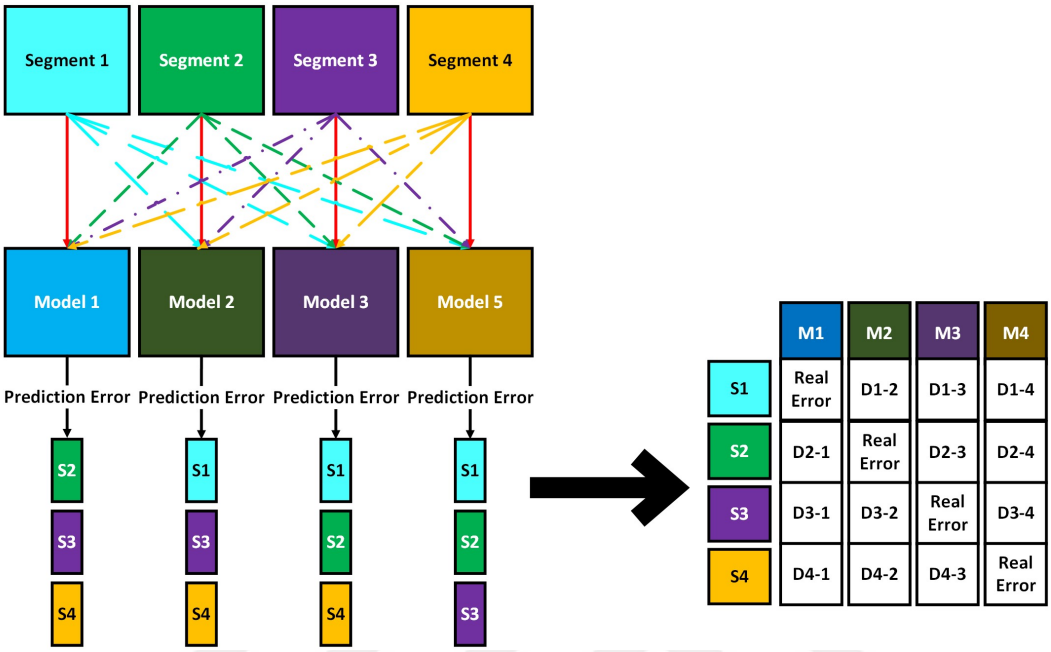


Figure 6.5 Block diagram of the proposed method of prediction based k-NN demonstrated with 4 segment scenario. Abbreviations used in this diagram like S1 and M1 mean segment 1 and model 1 respectively. The values in the matrix show the distances between segments for example D1-2 shows the distance between segment 1 to segment 2 and is used when finding similar segments to segment 1.

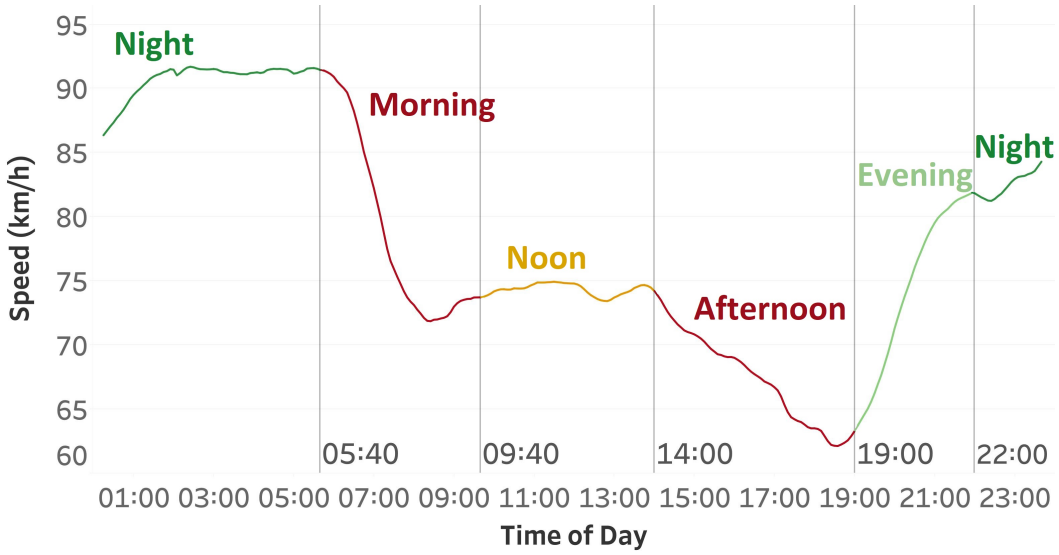
Represented by dotted arrows is the process of testing each segment with model trained for other segments to find the distance between using the MAPE values obtained. We do this step for every other segment and then repeat it for every model. At the end of this step, we create a distance matrix that has the MAPE values of every segment tested on every model created. The values here are the averages of 12 months so it represents the error obtained with a year's worth of data. The rows of this matrix represent the segment that data used for prediction belongs to while the columns represent the segment that the model belongs to. In the last step, we pick the lowest k MAPE scores to use for augmentation.

Our method is able to find similar road segments based on prediction thus for data augmentation this gives the best results. In this thesis, our goal is to increase the performance of long-term traffic speed predictions so we found the closest segments based on the similarity of traffic characteristics from a long-term perspective but our method could be used just as efficiently with short-term predictions as well.

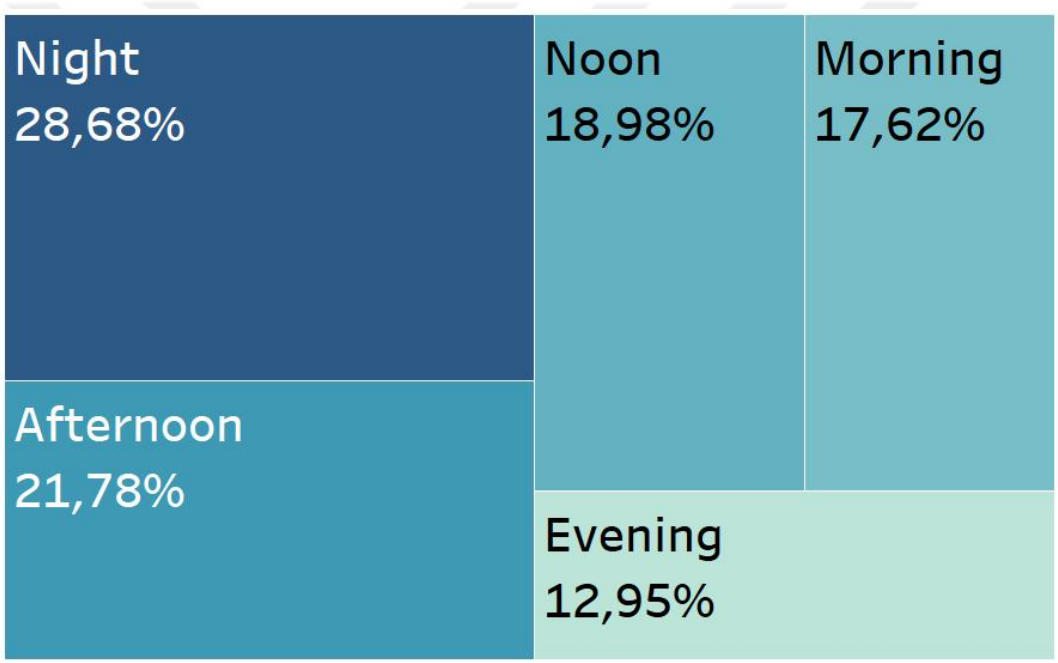
7
DATASET BALANCING THROUGH PATTERN ORGANIZATION

A common problem faced by both long- and short-term prediction is unbalanced datasets. Traffic speed changes throughout the day but most of the time there is little to no traffic which means high traffic speed values. An obvious difference is between nighttime which is nearly a third of the day and rush hours which are the only moments for traffic speed to reach a minimum that lasts only a few hours each day. This unbalance is contained in every traffic data set and limits the prediction success of traffic speed prediction. Because of this unbalance models trained on 4 months of historical speed data tend to make predictions usually higher than it suppose to be on very low traffic speeds. There are two ways to handle this type of problem one of them is to resample the dataset to have more balanced samples and the second one is to increase the number of samples that have less number of representations. The first method does not fit traffic speed predictions as the number of samples is already a problem and any more reduction results in under-trained models. The second method is data augmentation which is a complex problem in the Intelligent Transportation domain that we looked at more closely in this thesis. We propose a third method as an alternative to this imbalanced dataset problem. Rather than change the dataset size we emphasize aspects of data in the learning process. These aspects thus would be learned better and the bias that is caused by imbalance would be mitigated. This allows models trained for prediction to focus on certain patterns and ease the process by lowering the required information to be learned. Another strong point of our methods is the reusability of them with different model types which is not common within deep learning literature where most researchers suggest new models that are incompatible with each other.

We evaluated this problem based on three aspects of traffic data which are based on Time of day, based on direct speed, and based on the month. From the research we have done we believe ourselves to be the first ones to use this type of pattern organization to improve long-term traffic speed prediction. The methods discussed



(a)



(b)

Figure 7.1 (a) Average speed of 441 segments separated into 5 different groups based on time of day. (b) Percentage of data in each group

in this thesis uses traffic speed data alone to demonstrate its flexibility to be used with any type of model or dataset that may contain a different set of feature. We show this flexibility in Chapter 8 with 4 different models that are commonly used for speed prediction.

In this Chapter, we propose three novel techniques to improve the success of traffic speed prediction:

- **Hour-wise Pattern Organization**

Table 7.1 MAPE comparison between LSTM model trained with our 3 methods and LSTM model that has no methods applied in training

	LSTM MAPE Score	Difference with No Method
No Methods	13,74	
Hour-wise Pattern Organization	13,49	0,25
Speed-wise Pattern Organization	13,18	0,56
Month-wise Pattern Organization	13,6	0,14

Hours in a day show consistent patterns independent of other variables. Pattern organization through splitting the day based on the period that shows similar patterns and training a model for each of these groups.

- **Speed-wise Pattern Organization**
  Training models for similar speeds increase the model's capacity to learn certain patterns better. Pattern organization through splitting speed data with boundaries that encompasses similar patterns and training a model for each of these groups.

- **Month-wise Pattern Organization**
  Training models by using immediate past months of the test month results in models learning patterns that do not match the test month like using Patterns in August to learn September. Pattern Organization through using similar months for training models. This method can be used in conjunction with both Speed-wise Patterns Organization and Hour-wise Pattern Organization.

Examples given in this Chapter, are LSTM-based models. LSTM models used in this Chapter, have one LSTM layer followed by a dense layer and the output layer. The LSTM layer has 256 hidden units while the dense layer has 128 hidden units. Both layers have a dropout layer with a 0.4 dropout value. The dense layer is batch normalized while the LSTM layer is not. Experimentation with CNN, GRU, and Bi-Directional LSTM based models can be found in the Chapter 8.

7.1 Training Models with Hour-wise Pattern Organization

Most simple pattern traffic data display is dependent on the time of day. Certain hours in a day display similar characteristics concerning the day of the week or month. The most well-known example of this is rush hours in city traffic which

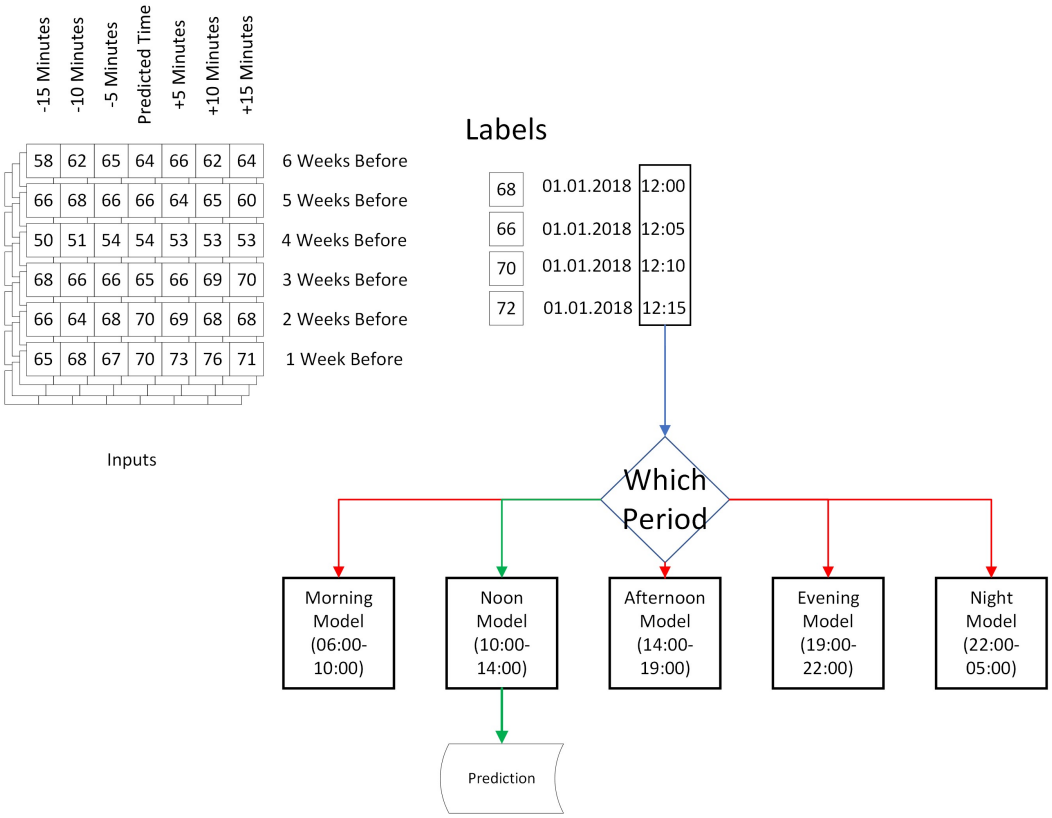


Figure 7.2 An example of how model creation using Hour-wise Pattern Organization method works

displays heavy traffic regardless of which day it is. Another example could be at night times when there is no traffic congestion recorded. Although weather and special days create differences, most of the time these differences only affect the value of speed rather than the pattern it displays.

As it can be seen in Figure 7.1a, certain times of day display similar characteristics, with this in mind we decided to split a day into 5 different periods. The first period is from 05:40 to 9:40, the second period is from 9:40 to 14:00, the third period is from 14:00 to 19:00, the fourth period is from 19:00 to 22:00 and the last period is from 22:00 to 05:40. We determined these groups to isolate increasing or decreasing characteristics from each other to help models learn these characteristics better. For each of these 5 periods, a different model is employed, and to predict a certain hour in a day, the model that matches the tested hour is used. An example of this process can be seen in Figure 7.2. In the figure, a matrix of 6 rows of the past week's data and 7 columns of 30-minute windows of traffic speed data is shown. Labels of these samples and the date these samples belong to are also given. According to the time of day data of the corresponding sample, we determine by which time group model, these samples will be trained and tested with. In the example, samples were

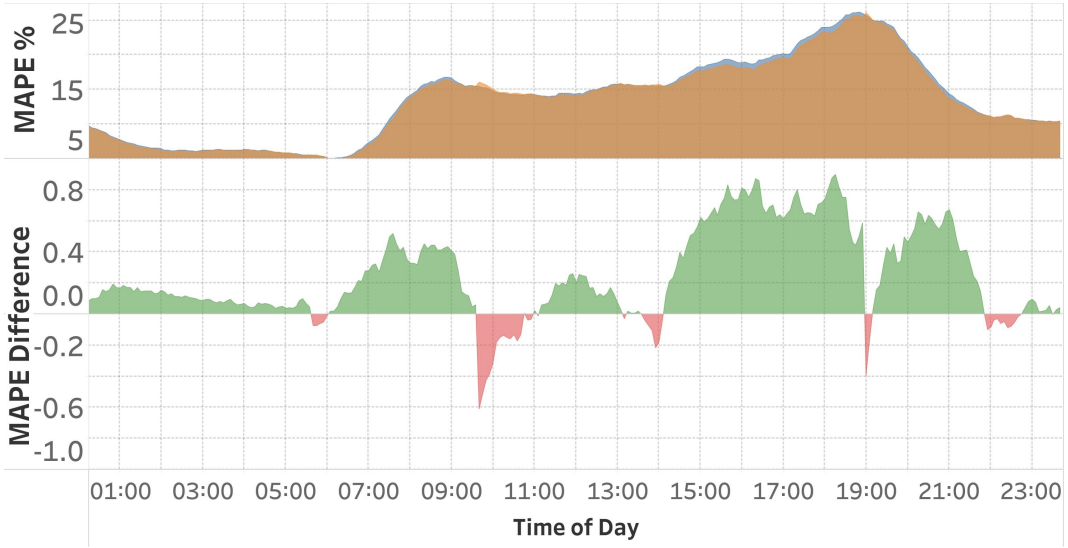


Figure 7.3 MAPE comparison between model training using Hour-wise Pattern Organization method and normal LSTM(up) and the difference between two model results(down) where green parts are where the applied method is better and red parts are where normal LSTM is better.

taken around 12:00 PM so these samples belong to Noon Model which encompasses 10:00 AM to 14:00 PM hours.

To measure the success of this method, a model is trained for all data, and results are compared to a model trained with the mentioned method. Table 7.1 shows the results of the base LSTM model and the contribution of the proposed pattern organization methods. Hour-wise pattern organization improved normal training MAPE by only 0.1 percent. In Figure 7.3 we give the results obtained in hourly basis. The graphic above shows the comparison between regular training process with a LSTM model and suggested method of Hour-wise Pattern Organization training process with a LSTM model. The below graphic shows the difference between these two training methods green regions representing the time periods where Hour-wise Pattern Organization has outperformed and red regions showing where it has underperformed the regular training process. An interesting detail can be found in this Figure that although the general increment is quite low, in certain hours specifically in rush hours an average improvement of 0.5 percent MAPE can be seen.

7.2 Training Models with Speed-wise Pattern Organization

Traffic Speed datasets contain an inherent imbalance since low-speed samples constitute only %11 of the total data. This imbalance makes learning these samples hard and most models generally predict higher than normal speeds. To fix this

problem we try training different models based on different speed groups. The basic idea of this method is to split the dataset into different speed groups and then train a model for each group. The most important part of this method is determining the speed boundaries. There are two important aspects to take into consideration. The first aspect is the amount of data in each group. Since deep learning methods require a certain amount of data, if a group has too few samples to train then results for this group will be worse. The second aspect is more complex as it involves the volatile nature of traffic speed data. Traffic speed especially on lower sampling frequencies such as 5 or 15 minutes, can change rapidly so creating groups with strict boundaries might underperform on border speeds. To give an example, a model trained to predict speeds between 60 and 80 will have many samples with a label of 55 or 85. Boundaries created must be flexible to account for these samples.

To determine the boundaries for this method we have to two problems into consideration: the amount of data at each speed group and the volatility of data. It is not realistic to make each speed group equal. The goal is to reach a certain percentage of data for each group so each group has an equal amount of data as possible. Besides, each group should have enough data to train a deep learning method. Volatility can be managed through overlaps between speed groups. An overlap occurs when the average of input data which is made out of 30 minutes of data 6 weeks before label speed, is in a different speed group than the label itself. In this case, we count this sample as an overlap between these two-speed groups. We want the narrowest overlaps possible while determining test sample boundaries to create predictions for samples on models trained on their characteristics. We want some overlap on training samples so that the model trained would be robust to overlaps in test datasets. While train models being robust to different labels is desired, we also have to remember that the purpose of this method is to limit variations of traffic characteristics in each model so the amount of overlap in one dataset cannot be so large since it leads to include multiple traffic characteristics in one model. We saw in our tests that 10 km/h range contains enough overlaps to make models robust while not including other traffic characteristics. With this contrast in train and test datasets demands we employ two different methods for creating train and test datasets that are depicted in Figure 7.4.

- Pattern Organization of *Train Samples*:

  To make models trained robust to the volatility mentioned above train samples should be split with intersecting boundaries. Values used in this process will be the label values of train samples which is the value predicted. As an

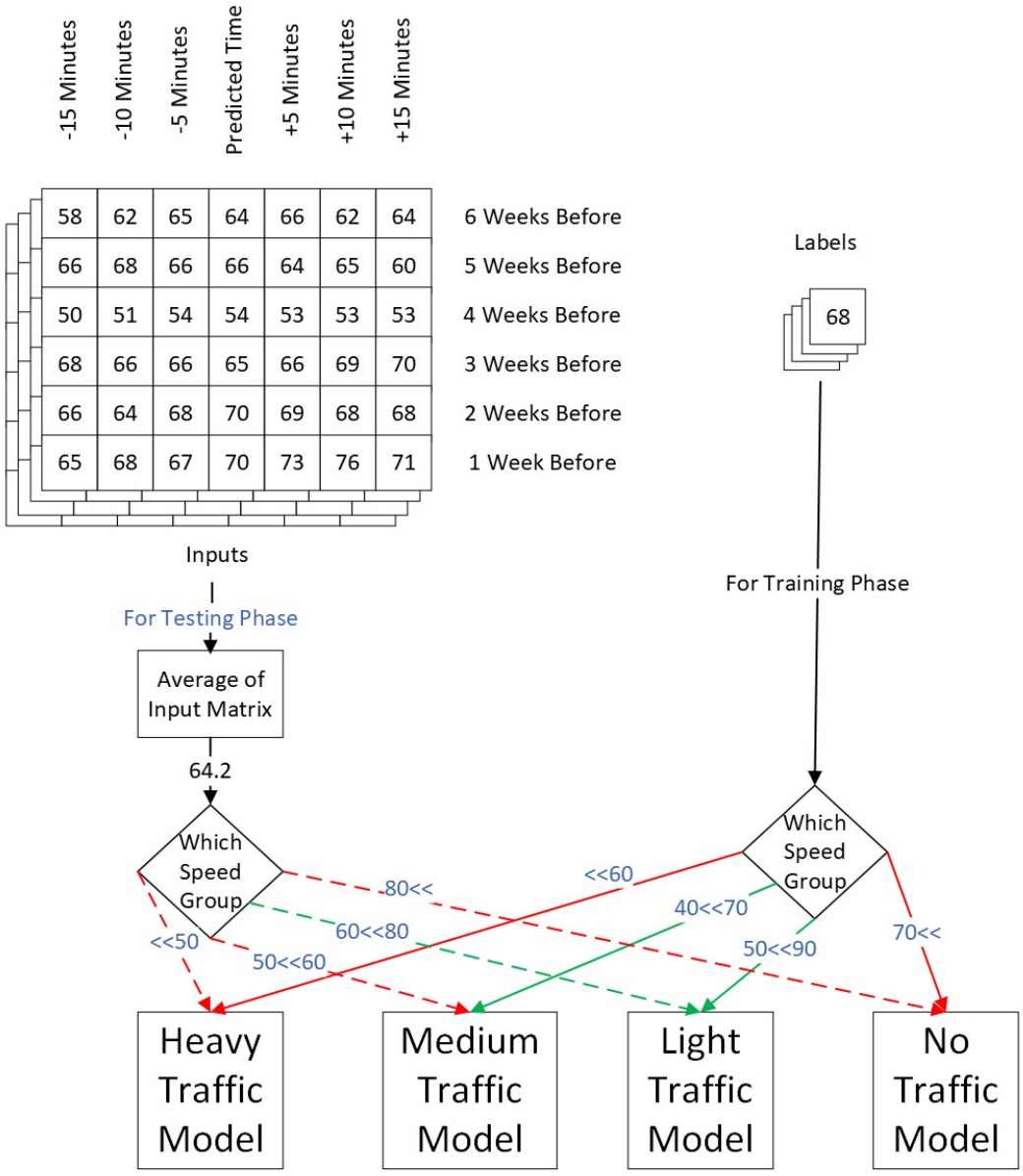


Figure 7.4 An example of how model creation using Speed-wise Pattern Organization method works

example in Figure 7.4 we see an input sample and its label. For the training phase, we use the label value which in this example is 68 and we determine which speed group this sample will belong to in the training. In the figure, we see that 68 belongs to two groups with intersections, one of them is the Light Traffic Model which has boundaries of 50 and 90, and the second one is the Medium Traffic Model which has the boundaries of 40 and 70. In the training phase, this sample will be used in both of these models. These intersecting boundaries not only make the model trained able to predict values far more than initially, but it also increases the number of samples models trained with.

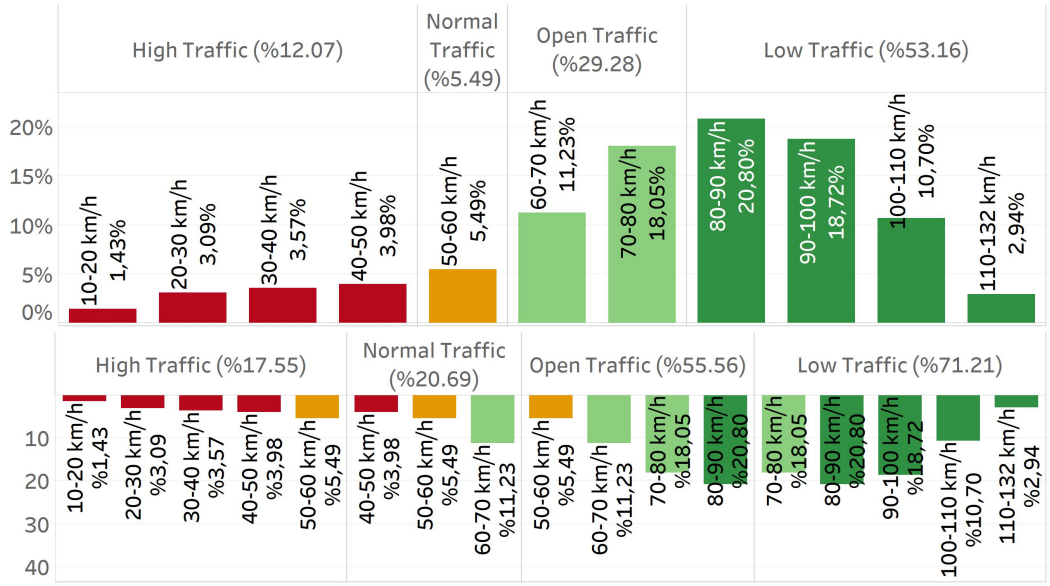


Figure 7.5 Dataset grouping used in Speed-wise Pattern Organization. Upper side of the Figure show the grouping of the test samples while the down side show the grouping of the train samples.

- Pattern Organization of *Test Samples*:

  After models are trained using boundaries found with labels, test samples must be assigned to a model appropriate to their speed. Unlike the training process, labels cannot be used in this decision. To make this assignment, input values that consist of historical speed must be used. We choose to take the average of all historical data that was meant to be the input values and choose the appropriate model based on that average. Another difference to the training process is that boundaries are not intersecting but independent of each other. As an example in Figure 7.4 we see an input sample with its label. We use the average of the input matrix which is 64,2 to determine which speed group this test sample belongs to. We can see in this Figure that 64,2 belongs to the Light Traffic Model and only this model as there are no intersections between speed groups in the testing phase. We use this speed group model to make a prediction.

For the dataset we used we determined the boundaries as given in Figure 7.5. In the same figure, we can also see the percentage of samples in each group. The upper side of the Figures hows the split of test samples. Each bar represents a range of 10 km/h while the percentage value shows how many samples exist for that range of speed. The colors of these bars show the category assigned to these ranges. Red for heavy traffic speeds, yellow for medium traffic, light green for light traffic, and dark green for no traffic. It can be seen that there is no overlap when predicting test

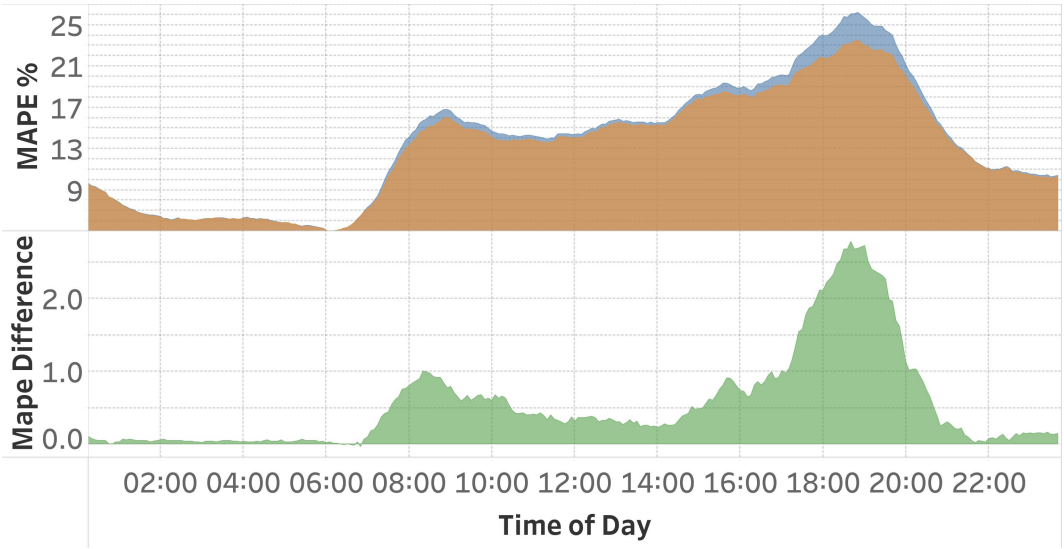


Figure 7.6 Comparison of normal LSTM model and model training using Speed-wise Pattern Organization method. First graphic shows MAPE values of two methods while the second graphic shows MAPE difference between them.

samples. The downside of the Figure shows the split of the train samples. Each bar still represents a range of 10 km/h of speed. The color of the bars in this Figure is the same color assigned in the test split. As it can be seen there are many overlaps between different groups of speed. For instance, a range of 50-60 km/h traffic flow constitutes a medium traffic state while making predictions, data within this range is also used for the training of heavy, medium, and light traffic states.

In Table 7.1 results for this method tested on a simple LSTM are given. The model has one LSTM layer with one fully connected layer. Table 7.1 demonstrates that this method improved MAPE by 0.5 which is a 3% percent improvement. Hourly MAPE results for both the naive method and model training using Speed-wise Pattern Organization are shown in Figure 7.6. In the second graphic, we can see that in rush hours which is when the traffic speed starts to slow down our method managed to improve results by more than 1.0 MAPE score, especially in the evening rush hour when the improvement reaches 2.8 which is a 10% improvement to MAPE obtained at that time.

7.3 Training Models with Month-wise Pattern Organization

Most of the Intelligent system literature uses previous months of the test month for training a model. Some works may use training samples picked as random from the entire dataset and others use previous weeks to predict. Although these methods are used widely, they do not consider the changes from month to month. Latter two of these methods are used mostly due to the lack of data while the former of these

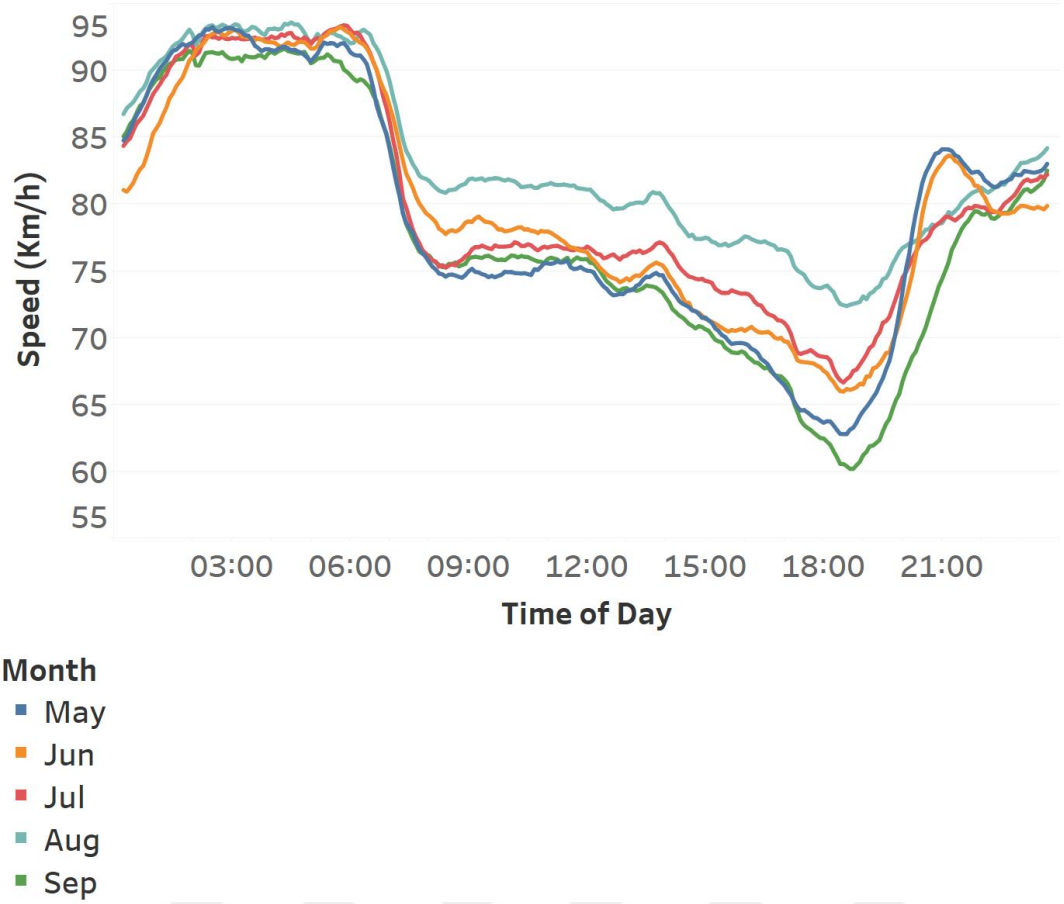


Figure 7.7 Speed values of 5 consecutive months

methods is used due to the lack of research done to inspect the similarities between months. In Figure 7.7 we see speed values of 5 consecutive months starting from May and ending with September. This Figure shows the dissimilarities between these months but specifically between September and August. Even though these months follow each other, we can see that they are the most dissimilar in this group. This difference isn't only limited to speed but also patterns due to external factors such as the weather in August in Istanbul is one of the driest months while September is usually rainy and windy. This difference extends to June and July while May seems most similar as it also is rainy most of the time in Istanbul.

This dissimilarity can also be seen in the MAPE scores we receive from regular speed prediction training. Figure 7.8 shows MAPE scores of an LSTM model trained to predict one month using 4 months that come before the test month. In this figure, we see that months that are a transition between seasons or months that have a clear difference from previous ones, like school holiday months, have higher MAPE scores than the others which continue trends that come before them.

We first used histograms to find the similarity between months. This method is able

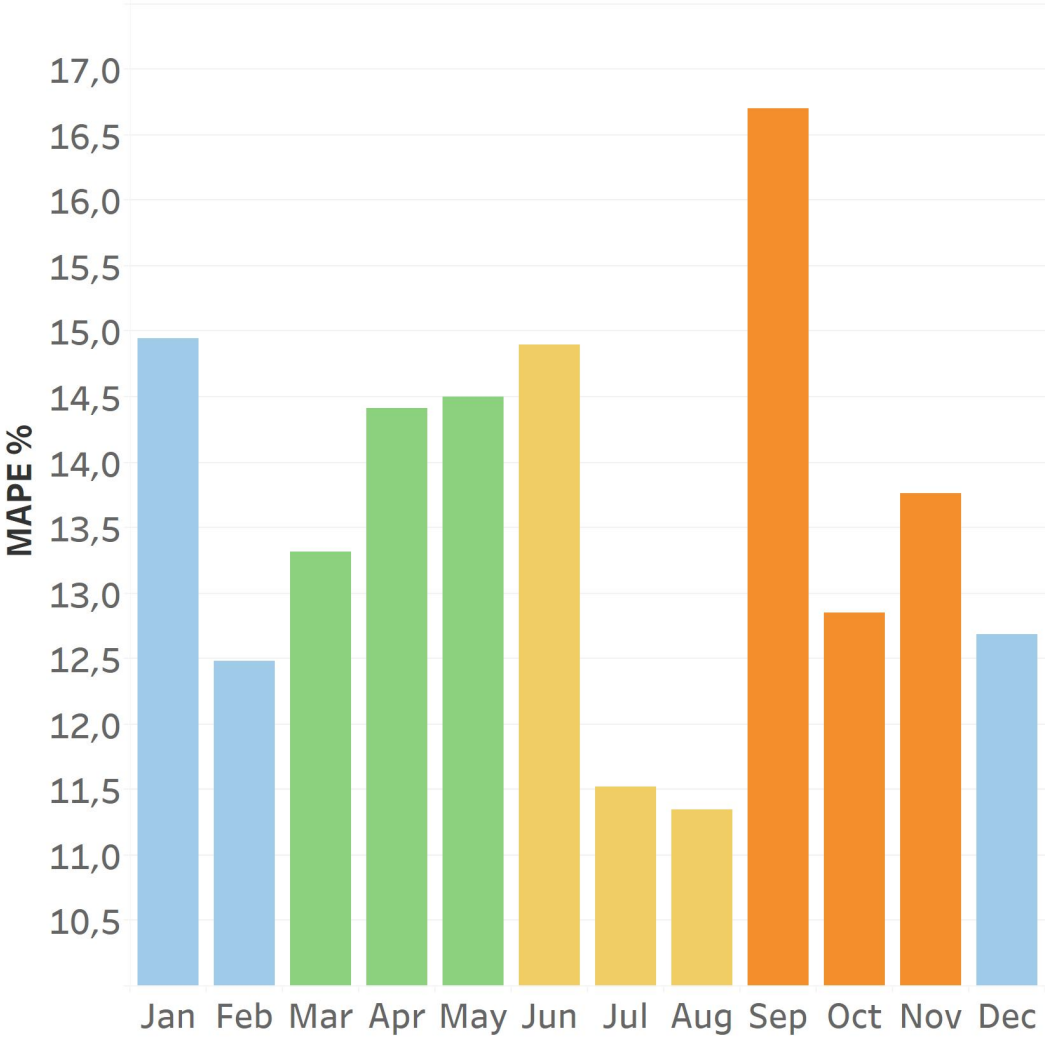


Figure 7.8 MAPE values of every month

to capture the distribution of speed data which is a good representation of traffic. To be able to give a better representation of patterns depending on time, we create 7 histograms for each month containing speed data of every day of the week thus minimizing the effect of different patterns each of these days carries. Histogram bins start from the minimum speed limit which is 10 in our dataset, and continue to the maximum speed limit, which is 132, increasing by 5 at each step. All of the 441 segments are aggregated into these histograms. For each day we take the average of the same Time of Day values meaning that for each Day of the Week, there are 288(24*(60/5)) values. In Figure 7.9 we show the histogram averages of April and August. The values shown in this Figure are the average values of the 7 histograms created and used for similarity measure. We show these two months' histograms overlapped to emphasize the difference between the two months. As can be seen in the figure, April has lower speed values while August shows greater speed values on average. Specifically August does not have any values lower than 20 on average. This does not mean there were no speed values recorded lower than

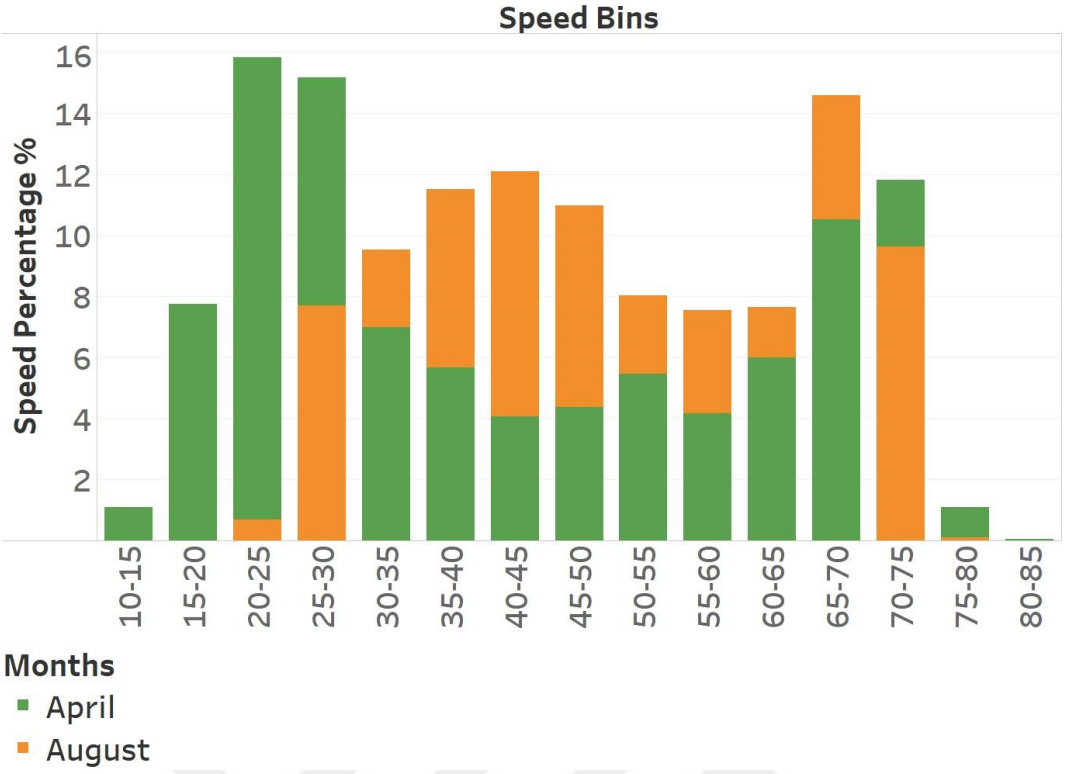


Figure 7.9 Histograms of months April and August



Figure 7.10 Month similarity between each other

20 in August but, on average August does not have traffic speed values lower than 20. To measure the distance we used the Euclidian distance method. While looking for similar months we compared 12 months that came before every month. For example, March 2019 compared to the months that followed March 2018.

In Figure 7.10 we can see the heat map representing the distance found between

months. In the same figure, we can see the distances in numerical form. The results we find support our idea of some months being dissimilar to months that came before them. Specifically, the September example given above is found to be more similar to May and April than August which is a month before September. In Table 7.2 we give the 6 most similar months for each month.

Table 7.2 6 closest months found

	Most Similar	2nd Most Similar	3rd Most Similar	4th Most Similar	5th Most Similar	6th Most Similar
Jan	Dec	Nov	Oct	Sep	Apr	Mar
Feb	Jan	Dec	Nov	Oct	Sep	Mar
Mar	Nov	Feb	Dec	Oct	Jan	Sep
Apr	Mar	Nov	Dec	Oct	Feb	Jan
May	Apr	Nov	Mar	Oct	Dec	Jan
Jun	May	Apr	Mar	Nov	Dec	Jan
Jul	Jun	May	Apr	Nov	Mar	Oct
Aug	Jul	Jun	May	Apr	Nov	Mar
Sep	Jul	May	Apr	Jun	Mar	Nov
Oct	Sep	Jul	May	Apr	Jun	Nov
Nov	Sep	Oct	Mar	May	Apr	Jul
Dec	Nov	Sep	Oct	Mar	Jul	May

Table 7.3 MAPE scores obtained with using Month-wise Pattern Organization(MPO) for training with LSTM, Bi-Directional LSTM, GRU, and CNN models and Base model which uses no method for training

	LSTM Base	LSTM MPO	Difference(Base-MPO)
Jan	15,57	15,52	0,05
Feb	12,82	12,78	0,04
Mar	13,6	13,42	0,18
Apr	14,61	14,45	0,16
May	14,63	14,68	-0,05
Jun	15	14,82	0,18
Jul	11,68	11,67	0,01
Aug	11,48	11,41	0,07
Sep	16,81	16,51	0,30
Oct	12,91	12,58	0,33
Nov	13,85	13,61	0,24
Dec	12,72	12,55	0,17

In Table 7.1 we show the results of an LSTM model trained with Month-wise Pattern Organization method. Results show an improvement of 0.1 MAPE score over an LSTM that has no methods applied with its training which is a 0.7% improvement. In Table 7.3 we give the results for each month obtained with this method compared to the regular training. We can see that with more drastic changes in weather, our method improves model performance better. While the passage to spring is milder thus improvements are smaller, with May having a slightly worse performance, transitioning to fall with rains. Since it coincides with the seasonal transition, when we make predictions about the months in the spring and fall periods, up to two times improvement is achieved. An example in the month of October, we see a MAPE decrement of 0.33 which is a 2.55% improvement in model performance.

8
EXPERIMENTAL RESULTS

In this Chapter we give the results of the methods we have described in the previous chapters pertaining to the problems of Missing data, Lack of data and Imbalanced datasets. The experiments were run with the purpose of increasing the accuracy of long-term predictions by fixing the problems datasets contains. We tested our methods with multiple model types to show their flexibility and robustness. In each section we give the experimental setup for the tested methods.

$$LossFunction = \frac{3 * MAPE + MAE}{4} \tag{8.1a}$$

All the models utilized a custom loss function combining MAE and MAPE scores. In our experiments for long-term traffic speed predictions we saw that using MAE as a loss function results in worse predictions for low speed values while using MAPE results in worse predictions for high speed values because models trained with MAE are more biased for high value predictions while the opposite is true for models trained with MAPE score. To make a balance between the two loss functions we combined them for training models that are not biased. In Formula 8.1, we give the custom loss function used for training our models. The coefficients in this formula aims to give weight to MAPE scores as low speed values are more important than high speed values.

Table 8.1 Server spesifications

Server Spec	Spec value
Server 1 Cpu	Intel Xeon W-2255 CPU
Server 2 Cpu	Intel Core i9-11900K
Server 1 Gpu	2xA5000
Server 2 Gpu	RTX 2080
Server 1 RAM	128 Gb
Server 2 RAM	64 Gb
Server 1 Hard Disk	512 Gb SSD and 1 Tb HDD
Server 2 Hard Disk	512 Gb SSD

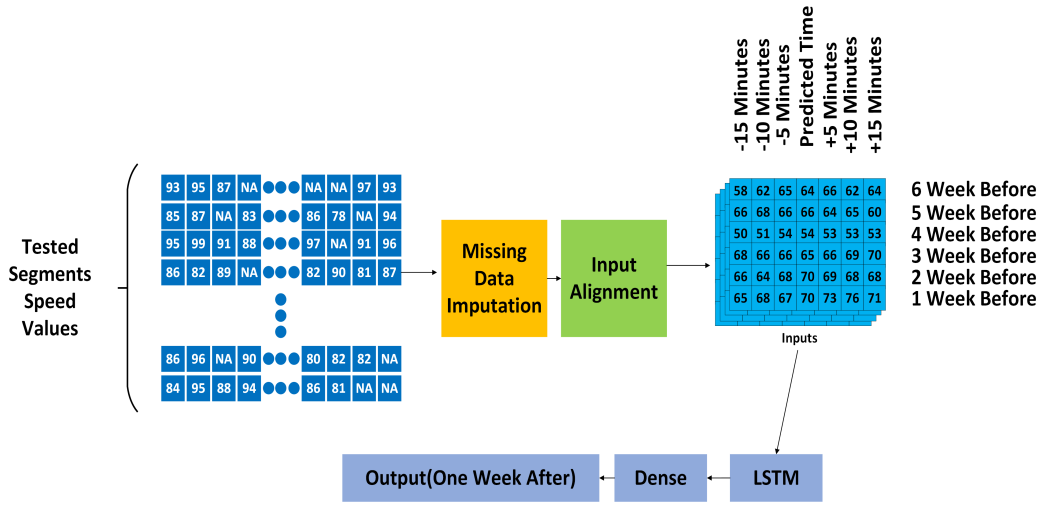


Figure 8.1 The experimental duration diagram for our test scenarios

Experiments in this Chapter were run on two different servers. The spesifications of these servers are given in Table 8.1.

8.1 Missing Data Imputation Results

In the previous chapter, we examined missing data imputation techniques using the MAPE values calculated with the ratio of real speed values and proposed values for replacing them. Most of the literature benefits from this simple evaluation method for performance measurement. On the other hand, there is a more sophisticated methodology [8][77] where the results of an imputation technique are inspected through a prediction problem perspective. The prediction time window in these studies is mostly short such as minutes to hours, whereas to the best of our knowledge, evaluating the effects of missing data imputation on long-term traffic speed prediction has not been studied yet.

In this section, we exploited long-term traffic speed prediction results by running two types of tests both to find the effectiveness of our imputation method and to prove its practicality. The first one, from an augmentation perspective, is to fill in real missing values and train a model based on the dataset with real-like values, whereas the other one is to replace the real speed values based on different percentages of missing values and observe the suitability of our missing data imputation method.

8.1.1 Experimental Setup

In Figure 8.1, we demonstrate the general outline of our data processing steps for our test scenarios, and in Table 8.2, we give the model parameters used in testing methods through long-term prediction. We made predictions for 20 road segments in Istanbul using the whole year's data in 2019. Table 8.4 gives the missing value percentage of the dataset before and after our method has been applied. We used

Table 8.2 Model parameters for long-term traffic speed prediction

Model Feature	Feature Values
LSTM Layers[78]	1
Dense Layers	2
Prediction Horizon	1 Week
Data Resolution	5-Minute
Train Data Size	4 Months
Input Window Size	6(Weeks Past)x7($\pm$ 15 minute)

Table 8.3 Long-term speed prediction MAPE for different percentage of missing value replaced in selected 20 road segments

Missing Percentage	Long-term Prediction MAPE
0% (No Change is Made)	14.6
5%	14.72
10%	14.74
20%	14.80
30%	14.93
40%	15.10
50%	15.32
60%	15.50
75%	15.87
90%	16.17

20 segments rather than the 441 segments used in the methodology to create a subset with more balanced missing value percentages. 20 road segments consist of 5 segments with a low percentage of missing values, 5 segments with a high percentage of missing values, and 10 segments which have an average percentage of missing values. The complexity of the proposed methods is O(N^2) but because in each step only a simple vector subtraction is done, less than an hour is needed for 441 segments.

Table 8.4 Long-Term traffic speed prediction results before and after the proposed missing data imputation method has been applied

Segment ID	Prediction MAPE Before Imputation	Prediction MAPE After Imputation	MAPE Difference	Geographicly Similar	Percentage of Missing Data Filled
1745_0	33.35	30.58	2.77	30.98	4.57
440_0	29.82	27.51	2.31	28.46	1.27
470_0	23.82	21.61	2.21	23.22	3.91
2093_1	19.94	19.85	0.09	19.98	18.63
1703_1	19.98	19.38	0.6	19.85	5.06
1755_0	17.78	16.92	0.86	17.67	0.01
421_1	17.1	16.11	0.99	17.08	0.72
318_1	15.97	14.98	0.99	15.45	0.01
426_0	13.09	12.96	0.13	12.98	0.01
76_0	11.88	11.62	0.26	11.53	5.5
307_0	11.48	11.45	0.03	11.45	1.67
2094_1	11.28	11.21	0.07	11.27	13.31
327_1	10.53	9.99	0.54	10.40	0.01
34_0	9.73	9.48	0.25	9.61	0.6
287_0	9.41	9.33	0.08	9.31	0.8
505_0	8.95	9.27	-0.32	9.19	0.68
623_0	8.06	8	0.06	7.92	0.01
509_0	6.61	6.65	-0.04	6.58	3.18
510_0	6.21	6.16	0.05	6.18	0.73
Average Results	**14.60**	**14.06**	**0.54**	**14.52**	**3.19**

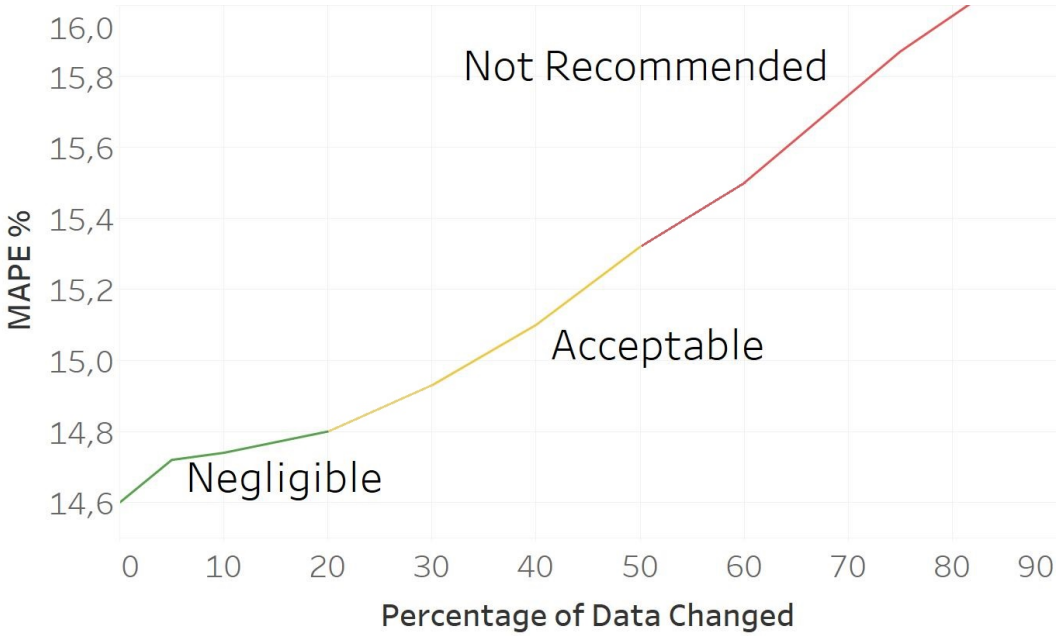


Figure 8.2 MAPE change regarding the missing value ratio. A percentage of 20% and lower is negligible while a percentage between 20% and 50% is acceptable and any percentage above 50% is not recommended.

8.1.2 Replacing Real Speed Values

We chose varying degrees of percentages of real data to replace in our dataset so that we could see exactly how much percentage of missing value we can replace without a substantial sacrifice for performance. In Table 8.3 and in Figure 8.2, we give the percentage of missing data and long-term speed prediction MAPE results. The results in Table 8.3 show that while the percentage of data replaced increases, MAPE also increases. This escalation can be explained by the destruction of the traffic characteristic at the corresponding segment due to missing values. Even though our model tries to find the best replacement, no artificial data can seamlessly mimic the sudden decreases in the dataset or changes in environmental circumstances. The disappearance of traffic characteristics might lower the success of the model in prediction, whereas however, the prediction error after filling gaps using the proposed method is negligible under 20% or lower missing value percentage and acceptable between 20% and 50% missing value percentages. Beyond the 50% percentage of missing value, the error rate of the prediction model increases and our model loses its efficiency.

8.1.3 Filling the Missing Values

The segments that we have chosen have missing data inherently, thus we test the proposed method with a more realistic scenario and fill in these missing values using our method. Since we use other segments to fill in missing values there could be some values that are missing in both two closest segments. After filling them we train a long-term speed estimation model and predict traffic speed one week ahead. In Table 8.4, we give the results of our test before and after missing data imputation. In this table results obtained with geographicaly closest segments is

also given. We can see that the improvement is much lesser when using forward and bacward segments for missing data imputation compared to the results obtained with the proposed method.

Analyzing the results shows that our method has a mostly decreasing effect on MAPE results. Most road segments have an improvement above 0.5 in terms of MAPE, whereas the MAPE values of only 2 road segments have increased. Following this outcome, we could safely recommend the researchers exploit our method to impute missing traffic speed data for long-term speed prediction models.

8.2 Data Augmentation Results

In this section, we evaluate the methods we have employed to mitigate the imbalance of traffic speed datasets. We inspect clustering techniques we have introduced in Chapter 6 and then show the results of the proposed method of prediction based k-NN.

8.2.1 Experimental Setup

In this section, we have used the dataset described in Chapter 4. This dataset is collected by Istanbul Municipality over the course of 2 years: 2018 and 2019. It contains traffic speed values collected from 441 main road segments in Istanbul with 1-minute intervals. Each of these road segments are on average 1 Km of length. We have changed the data intervals to 5-minute from 1-minute for reasons explained in Chapter 4. We have selected 106 segments from 441 main road segments. The selection of these segments involves looking at speed variation of the road segments which is one of the indicators of how hard to predict a road segment is. We chose these 106 segments so that the original dataset and selected dataset have similar average standard deviation of speed values. To make sure that spread of road segments are homogeneous in the entire map in our selection we used consecutive road segments. So the selection process is summarized as looking at consecutive road segments and selecting road segments that at the end results in similar average standard deviation to the original dataset.

The models used in this section, is an LSTM model. We have chosen LSTM for its success in long-term traffic speed predictions. The model architecture includes one recurrent layer and two dense layers one of which is the output layer. LSTM layer has 256 nodes while the dense layer has 128 hidden nodes. Both LSTM layer and dense layer has dropout applied to them with 0.4 dropout coefficient also both of these layers are regularized with L2 regularizer with 0.002 regularization coefficient. The training process is stopped early if no improvement is observed for 10 to 100 epoch depending on the number of samples the training dataset includes. More samples training samples have less epoch is waited to stop training to limit the amount of time training takes. This does not decrease the prediction performance as the model reaches the local minimum with less epoch as with increase of samples the number of iteration each epoch contains also increases and this makes models learn similar amount of information with less number of epochs. For the worst-case

Table 8.5 MAPE scores obtained with clustering methods for 106 selected segments

Clustering Method Name	Cluster Number	MAPE
Regular Results	0	13.76
One Cluster	1	13.89
Base Clustering	2	14.00
Base Clustering	10	14.14
Base Clustering	220	14.16
DBScan Clustering	12	14.05
Feature Extraction	2	13.97
DTW Clustering	16	14.14
Traffic Hours	4	13.97
Period-wise Clustering	6 to 1	14.27

scenario where improvement is continued to be observed the training is stopped at 1500 epochs. From our observations training lasts on average 500 to 800 epochs before no improvement is achieved. To find similarity we have employed models that are trained for each segments. This process requires each segment to be tested with the models created for each segment. This means that it has the complexity of O(N^2) and it takes one hour to find the distance between one segment and rest of the segments. For 106 segments used for these experiments 4 days were spent for calculating the distances.

For our experiments, we used two servers with one server contains two A5000 GPUs with 128 Gb Ram while the other server contains one RTX 2080 with 64 Gb Ram. One training took on average 24 hours on either of the servers as the models trained on small datasets the capabilities of A5000 are not utilized full and performed at similar efficiency with RTX 2080.

8.2.2 Results of Clustering Methods

To solve the lack of data problem we have employed clustering techniques. These techniques are described in Chapter 6. In Table 8.5 we give the results obtained by using clustering techniques for data augmentation. In this Table, we can see that models trained for predicting segments within clusters performs worse compared to models trained for predicting each segment separately. This is caused by the fact that models trained with entire clusters predicts speed based on a general characteristics that is common in the cluster while this is not able to outperform the model that learns the characteristics that belongs to the segments itself. Even though these clusters contains 10 to 100 times of more data because of this generalization it performs worse than regular training process. One advantage of this type of clustering method has compared to regular training is the training time. The number of models trained is lower in clustering methods training time drops from 24 hours with training a model for each segment to 2 or 3 hours with clustering method.

These results coincide with our finding with missing data imputation experiments

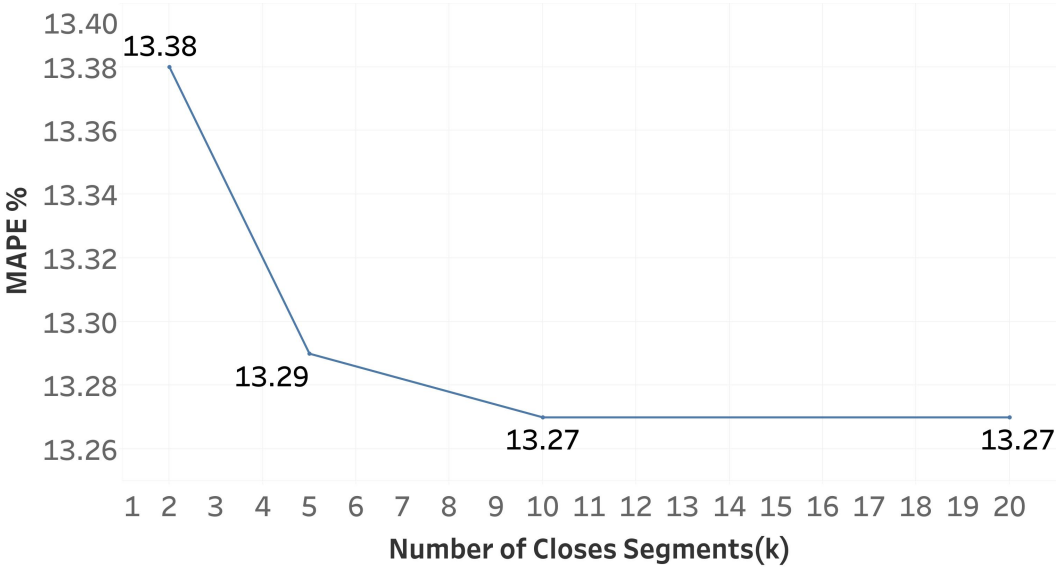


Figure 8.3 MAPE scores obtained with different number of k values using Prediction based k-NN method

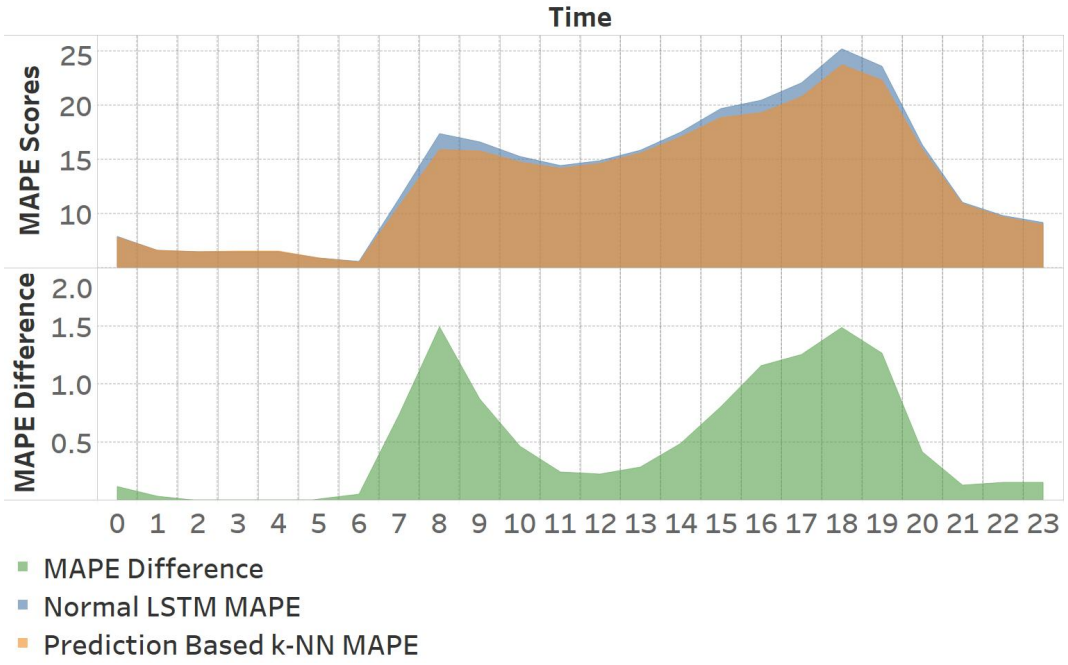


Figure 8.4 MAPE difference between regular long-term prediction and data augmented long-term prediction.

where clustering techniques also performed worse compared to more specialized solutions. We infer from these results that general solutions like clustering does not perform well for a problem like traffic speed prediction which contains substantial variation between each other. More specialized solutions like k-NN performs better as with these types of specialized solutions each segment is evaluated separately.

Table 8.6 MAPE scores obtained with Prediction based k-NN Method

Results with LSTM Model	
Method Name	**MAPE**
Regular Results	13.76
Geographicly Closest Segments	13.66
Weighted k-NN Augmentation*	13.63
Prediction Based k-NN	13.27
Speed-wise Pattern Organization + Prediction Based k-NN	12.86
Results with GRU Model	
Method Name	**MAPE**
Regular Results	13.48
Prediction Based k-NN	13.06
Speed-wise Pattern Organization + Prediction Based k-NN	12.80

* Method that is proposed in Missing Data Imputation

8.2.3 Results of Prediction-Based k-NN Method

We have tested prediction based k-NN method with an LSTM method and compared the MAPE scored obtained with a model that was trained with data augmentation method applied. Before this comparison though optimal number of segments that will be used for data augmentation must be determined. To determine the optimal k number we have experimented with multiple number of segments with our augmentation method. In Figure 8.3 we show the results for different number of k values which represents the number of closest segments used in data augmentation process. In this Figure we can see that no improvement is observed with k values that are bigger than 10 so for any comparison we will examine further we have used 10 for k value. These results were taken from a long-term traffic speed prediction model. In Table 8.6 we see the MAPE scores obtained from long-term prediction models and observe that prediction based k-NN have managed to lower the MAPE score by 0.51 points which is a 3.7% increase in prediction performance. In this table we can also see that our proposed method of prediction based k-NN has outperformed both k-NN using distance between speed values which is the method we have proposed for missing data imputation problem and using two closest segments based on geographical position. Looking closely at the results of prediction based k-NN by looking at MAPE change based on hour in Figure 8.4 we can see that data augmentation has managed to increase the prediction performance 1.5 point on two rush hours which contain the lowest traffic speed values in the dataset. This is a 8.6% increase in morning rush hours and 6% increase in evening rush hours. We also experimented with Speed-wise Pattern Organization technique alongside Prediction based k-NN method, the results of which can be seen in the same Table 8.6. Combining these two methods, we were able to lower the MAPE score from 13.76 to 12.86 which is a 6.6% prediction performance improvement.

From these results, we determine that unlike clustering techniques which uses the similarity between segments in the same fashion as our method, it was able to

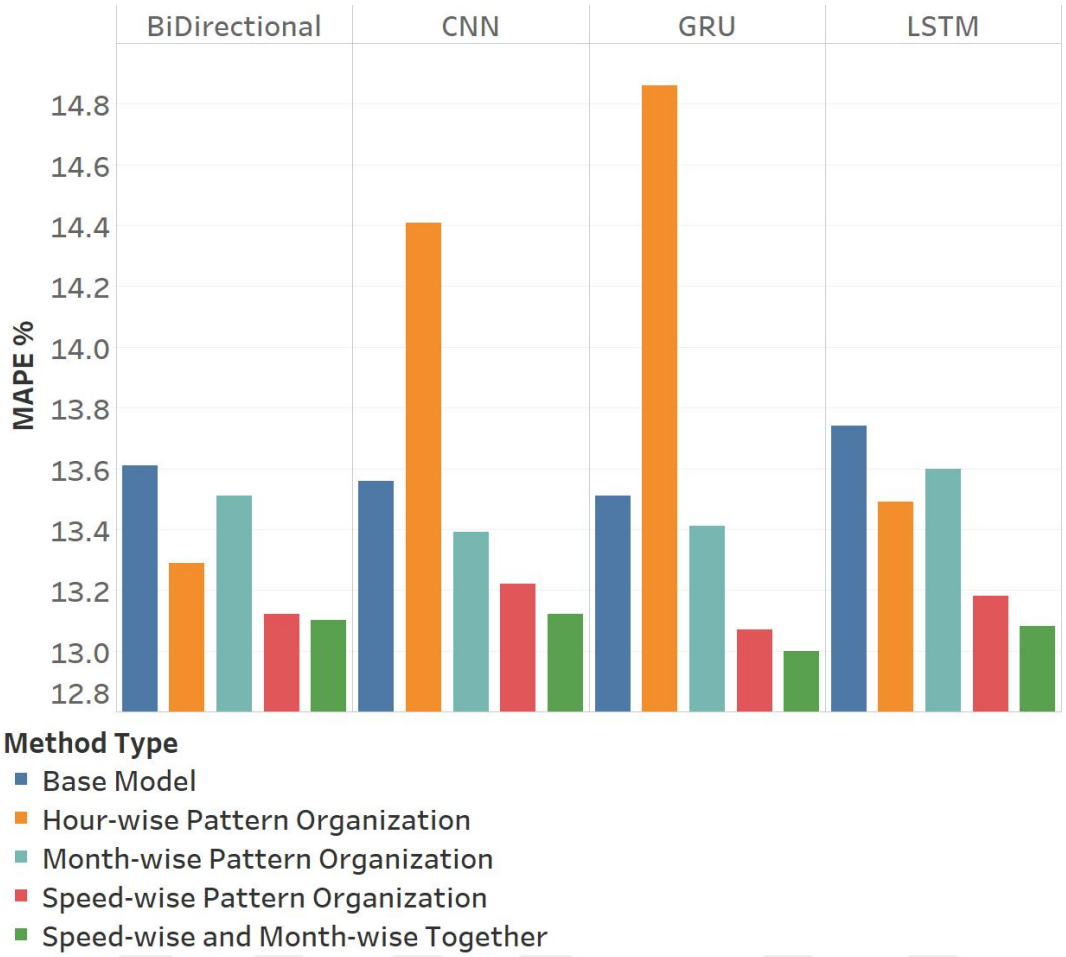


Figure 8.5 MAPE scores obtained with 4 base models. Base models show MAPE scores obtained with no additional method applied. Hour-wise, Month-wise and Speed-wise Pattern Organization shows the MAPE obtained with methods proposed for imbalanced dataset problem. Speed-wise and Month-wise shows the results obtained using the two methods together.

improve the prediction performance. These results also shows that more specialized solutions are more successful with traffic speed prediction problem.

8.3 Imbalanced Dataset Results

In this section, we evaluate our three methods for data imbalance and a combination of Speed-wise and Month-wise Pattern Organization methods. To evaluate our methods we compared the MAPE scores of a normal training process where no additional steps has been applied to the dataset and training process of our methods. We have experimented with 4 different model types to illustrate the flexibility of methods suggested. To our knowledge we are the first research to balance the traffic speed dataset so unlike the previous chapters our comparisons does not include a benchmark method from literature.

Table 8.7 MAPE scores obtained with our method with LSTM, Bi-Directional LSTM, GRU, and CNN models

	LSTM	Bi-Directional	GRU	CNN
No Method	13,74	13,61	13,51	13,56
Month-wise Pattern Organization	13,6	13,51	13,41	13,39
Hour-wise Pattern Organization	13,49	13,29	14,86	14,41
Speed-wise Pattern Organization	13,18	13,12	13,07	13,22
Speed-wise and Month-wise	**13,08**	**13,1**	**13**	**13,12**

Table 8.8 Experimental setup and server spesifications

Training Spec	Spec value
Gpu	2xA5000
Epoch	800 to 1500
Total Training Time	Less than 24 Hours
Batch Size	2048
Number of Segments	111

8.3.1 Experimental Setup

To demonstrate the contribution of our methods we used 4 different base models: CNN, LSTM, Bi-Directional LSTM, and GRU. Our CNN model contains 2 convolution layers with Relu activation both of which are batch normalized and applied dropout method. Following these convolution layers is an average pooling layer then a dense layer which is also batch normalized and applied dropout method. Each of the convolution layers has 128 kernels while the dense layer also has 128 nodes.

LSTM, Bi-Directional LSTM, and GRU all have one recurrent neural network layer of their respective types and followed by a dense layer and an output layer. While LSTM has 256 units, Bi-Directional and GRU have 512 units. In all of these base model types Dropout and kernel regularization are used. After every dense layer batch normalization is also applied.

Models in this thesis were trained with a server running an A5000 GPU. On average, each model took nearly a day to train for 111 road segments. This means each segment on average trained for 13 minutes. Each segment trains a model for every month a year so one model training took 1 minute on average. Early stopping was utilized so each model trained a different amount of epochs but on average each model trained for 800 to 1500 epochs with a batch size of 2048. Testing of a sample takes less than a second so it can be ignored. These information is summarized in Table 8.8. The methods themselves are applied before training and does not use GPU for calculations. Separation of the training dataset is O(1) complexity and only takes few seconds.

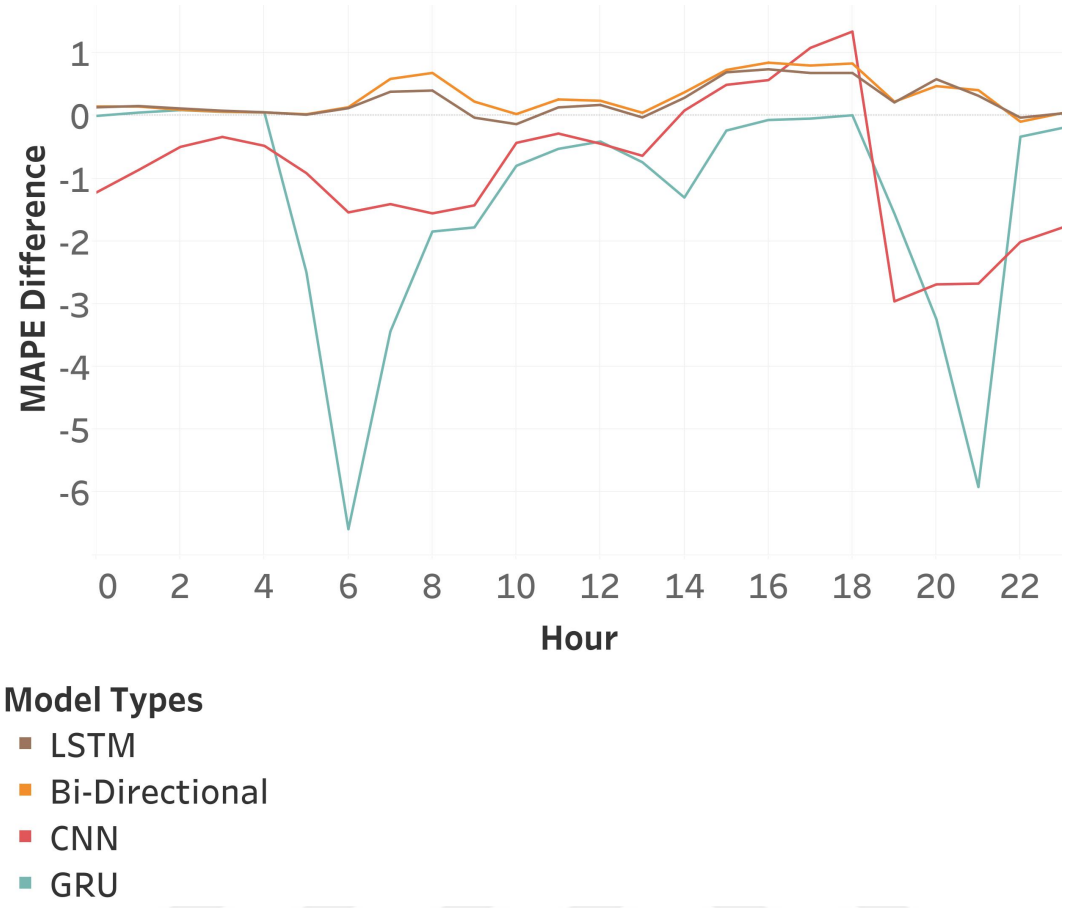


Figure 8.6 Hourly MAPE difference between models trained with no additional process and models trained with Hour-wise Pattern Organization method.

8.3.2 Results For Model Training Based on Hour-wise Pattern Organization

In Figure 8.5 we show the MAPE results obtained while using the method that separates the values based on the time of day data was recorded. These results are compared to a model trained with no additional process applied to test its effectiveness.

Results can be seen in the Figure 8.6. In this figure it can be observed that LSTM and Bi-directional LSTM have improved every hour while CNN model has improved only in evening hours. Prediction performance of GRU model has not improved with Hour-wise Pattern Organization method.

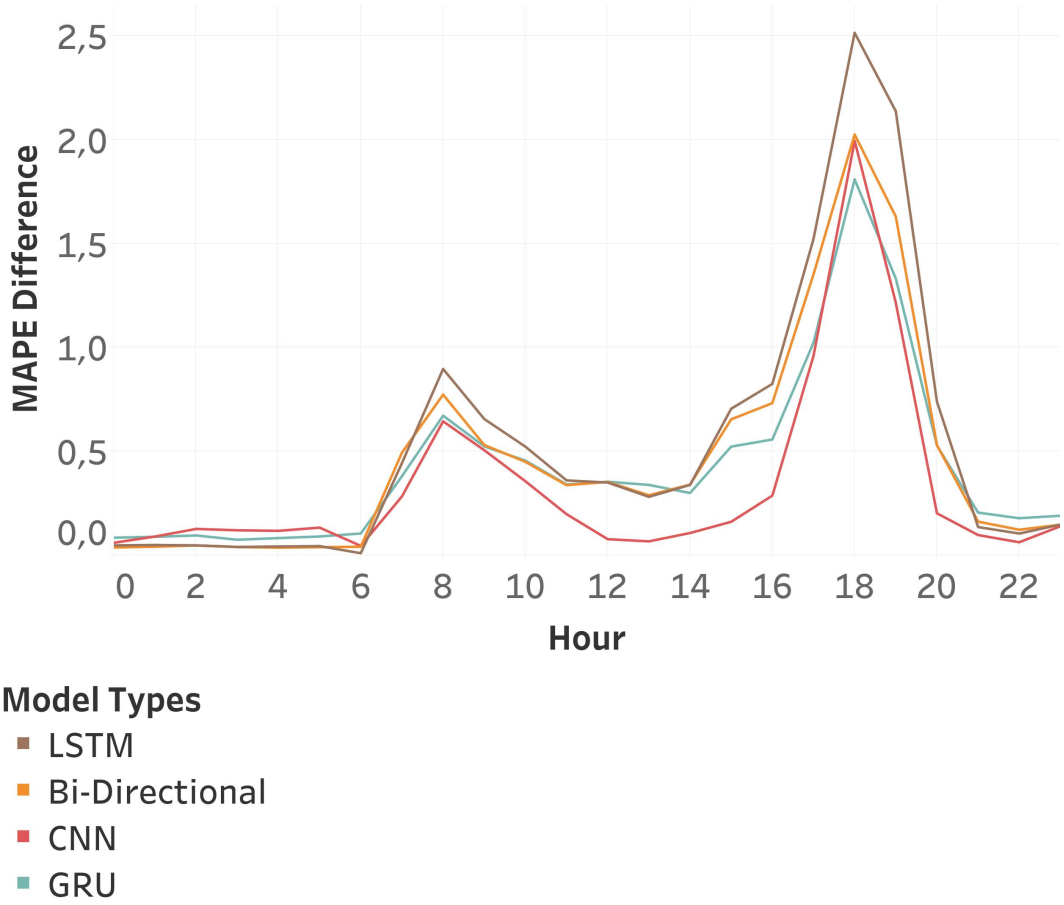


Figure 8.7 Hourly MAPE Difference between models trained with no additional process and models trained with Speed-wise Pattern Organization method.

Table 8.9 MAPE scores received with using Month-wise Pattern Organization(MPO) for training with LSTM, Bi-Directional LSTM, GRU and CNN models and Base model which uses no other method for training

	LSTM Base	LSTM MPO	Bi-Directional Base	Bi-Directional MPO	CNN Base	CNN MPO	GRU Base	GRU MPO
Jan	15,57	15,52	15,41	15,47	15,33	15,29	15,39	15,35
Feb	12,82	12,78	12,68	12,68	12,69	12,68	12,78	12,74
Mar	13,6	13,42	13,5	13,32	13,43	13,34	13,48	13,3
Apr	14,61	14,45	14,49	14,33	15,32	14,3	14,63	14,42
May	14,63	14,68	14,45	14,57	14,41	14,61	14,56	14,56
Jun	15	14,82	14,76	14,63	14,67	14,6	14,92	14,68
Jul	11,68	11,67	11,53	11,58	11,63	11,67	11,67	11,7
Aug	11,48	11,41	11,32	11,3	11,36	11,36	11,5	11,49
Sep	16,81	16,51	17,06	16,72	15,63	15,39	15,55	15,38
Oct	12,91	12,58	12,64	12,44	12,85	12,46	12,58	12,47
Nov	13,85	13,61	13,65	13,44	13,48	13,29	13,39	13,27
Dec	12,72	12,55	12,56	12,45	12,74	12,56	12,57	12,47

8.3.3 Results For Model Training Based on Speed-wise Pattern Organization

In the graphic 8.5 we show the MAPE obtained with our method of grouping based on speed along with other MAPE scores. We can see in this graphic that in every type of model, there is an improvement while the margin of this improvement is different for different model bases. LSTM improves more than the other model types with 4%, while CNN improves least with 2.5%. We believe this is because of the inability of CNN to learn with smaller datasets. While GRU has only improved by 3.2% the improvement happened at higher MAPE thus we can say it learned more important information. In the graphic 8.7 we see that even though improvement to MAPE seems little when inspected closely we see that improvement is much more in high traffic points thus proving the effectiveness of our method. We can also observe that although the value of improvement changes, similar improvement patterns are observed in all of the base models trained. This means that our method learns an aspect of traffic that is hard to learn in all of these models. Our method is thus suitable for different models as it helps this aspect of traffic to be learned.

8.3.4 Results For Model Training Based on Month-wise Pattern Organization

In Figure 8.5 and Table 8.7 we show the MAPE obtained with our method of using similar months for training rather than previous months. Although the calculated average improvement is less than 1%, Table 8.9 shows that monthly improvement varies depending on the dissimilarity of the tested month to the previous months. Predicting dissimilar months like September or March, obtained decrement in MAPE score is close to 0.3 which corresponds a 1.7% improvement in performance. Another aspect is that although some months do not seem to improve, no significant decrease is observed.

8.3.5 Results For Model Training Based on Speed-wise and Months-wise Pattern Organization

In the previous Chapter, we discussed the possibility of using multiple methods at the same time. In this section, we show the results of using model training based on both speed and similar months. These two methods do not hinder each other as one uses speed values of the training dataset while the other changes which months would become the training dataset. In the graphic 8.5 we show the MAPE obtained with these two methods compared with other results. In Table 8.7 We see that the best results obtained thus far are by combining these two methods. The best overall success is achieved by GRU with 13 MAPE score and using both speed grouping and similar months methods has an improvement of 0,51 MAPE score compared with a model trained with no methods. This improvement is a 3,77% improvement while the biggest improvement belongs to the LSTM model which improved 0,66 MAPE score or 4,8%. CNN and bi-directional models both improved with 3,2% and 3,7% respectively. The difference between this combination and speed-based training is similar to the difference between the base model and training based on similar months. We concur that this combination works because both methods help the model learn different patterns thus combining them does not overlap the benefits.

9
CONCLUSION

In this thesis, we explore dataset problems faced in traffic speed prediction domain. We categorize the problems in three different groups: Missing values, Lack of data and Imbalanced dataset. We try various methods and developed novel techniques for each of them.

For missing values, we introduced half a dozen of missing value imputation techniques and tested their effectiveness on long-term traffic speed prediction models. To the best of our knowledge, it is the first study, which evaluates the performance of missing speed value imputation techniques in terms of long-term traffic speed prediction error rates. Specifically, we proposed a robust method of relying on the idea of finding similar road segments and using the weighted average values of the two most similar segments for filling in missing speed values. The proposed method can increase the prediction success of long-term traffic speed estimation problems with an improvement of 3.6%. We also found that 20% or a lower amount of missing data can be effectively replaced by the values of similar segments, whereas any segment that has missing values below 50% can be filled in with an acceptable deterioration ratio. In future work, we will be analyzing different clustering techniques and novice features like spatiotemporal relation of city-wide speed values occurring on other segments at particular time zones.

For the second problem, namely Lack of data, we experimented with different methods of using road segments with similar traffic characteristics together. We developed a novel augmentation method that uses prediction results obtained with models trained using different road segments data. The proposed method used the MAPE score obtained from these models predicting target segment data as distance measure and found k similar road segments to use together for training. This method is used for long-term prediction problems as the models used for finding distance values were trained for long-term prediction. With our method we improved the performance of prediction model 3.7%. In the future, better merging techniques of similar segments or similar segments based on monthly similarities will be explored.

For imbalanced dataset problem we propose three novel techniques that doesn't change the dataset size, which is the usual method for imbalanced dataset, but rather emphasize certain patterns for models to learn easier. These patterns are aspects of traffic characteristics and each carry different information of these characteristics. These patterns are based on hours, traffic speed and months. Our developed

techniques were tested on 4 popular models types that are used frequently in traffic speed predictions. We found that our methods improved performance of each of these model types. Hour-wise Pattern Organization's improvement to performance is around 2%, Speed-wise Pattern Organization's improvement is around 4% and for Month-wise Pattern Organization's improvement is around 1%. The real strength of our methods though may not be apparent on average improvement as the goal of these method is to make sure certain patterns are learned and predicted better. Spesifically these patterns are the high traffic, low speed periods. If we look at these moments the improvements of proposed methods are more obvious. For Hour-wise Pattern Organization the goal is to learn the rush hours and the performance improvement in these hours are around 4%, for Speed-wise Pattern Organization the goal is to learn low speed values and the performance improvement in low speed values is around 10% and for Month-wise Pattern Organization the goals is to learn transition months which have different characteristics than months that comes prior to them and the performance improvement in these months is around 2.5%. Our methods specifically month based emphasis could be used with other two methods together improving performance more than when they are used separately. Using month and speed based emphasis together we improved model success by 5% on average. Also combining data augmentation with Speed-wise Pattern Organization we have increased the aveage improvement to 6.6% which is much higher than Speed-wise Pattern Organization utilized separately.

In this thesis, we have inspected the major problems concerning the dataset used for traffic speed prediction problem and more spesifically long-term traffic speed prediction. Proposed methods have improved the prediction accuracy long-term prediction models by considerable measure and produced more robust models.

REFERENCES

[1] L. Qu, W. Li, W. Li, D. Ma, Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with applications*, vol. 121, pp. 304–312, 2019.

[2] A. Zhang, Q. Liu, T. Zhang, "Spatial–temporal attention fusion for traffic speed prediction," *Soft Computing*, pp. 1–13, 2022.

[3] A. Nasser, V. Simon, "A novel method for analyzing weather effect on smart city traffic," in *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2021, pp. 335–340. doi: 10.1109/WoWMoM51794.2021.00061.

[4] S. Kwoczek, S. Di Martino, W. Nejdl, "Predicting and visualizing traffic congestion in the presence of planned special events," *Journal of Visual Languages & Computing*, vol. 25, no. 6, pp. 973–980, 2014.

[5] J. An, L. Fu, M. Hu, W. Chen, J. Zhan, "A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information," *Ieee Access*, vol. 7, pp. 20 708–20 722, 2019.

[6] B. Yao, B. Yu, "Short-term traffic speed prediction for an urban corridor," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 2, pp. 154–169, 2017.

[7] Z. Wang, J. Hu, G. Min, Z. Zhao, J. Wang, "Data-augmentation-based cellular traffic prediction in edge-computing-enabled smart city," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4179–4187, 2020.

[8] Z. Zhang, F. He, X. Lin, Y. Wang, M. Li, "High-performance traffic speed forecasting based on spatiotemporal clustering of road segments," *IET Intelligent Transport Systems*, vol. 15, no. 2, pp. 225–234, 2021.

[9] N. Kim, S.-W. Kim, "Apots: A model for adversarial prediction of traffic speed," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, IEEE, 2022, pp. 3353–3359.

[10] I. Laña, J. L. Lobo, E. Capecci, J. Del Ser, N. Kasabov, "Adaptive long-term traffic state estimation with evolving spiking neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 126–144, 2019.

[11] M. S. Ahmed, A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 1979.

[12] M. Van Der Voort, M. Dougherty, S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[13] S. Lee, D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transportation research record*, vol. 1678, no. 1, pp. 179–188, 1999.

[14] B. M. Williams, L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[15] W.-H. Lin, "A gaussian maximum likelihood formulation for short-term forecasting of traffic flow," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, IEEE, 2001, pp. 150–155.

[16] G. A. Davis, N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.

[17] X. Gong, F. Wang, "Three improvements on knn-npr for traffic flow forecasting," in *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, IEEE, 2002, pp. 736–740.

[18] L. Zhang, Q. Liu, W. Yang, N. Wei, D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[19] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[20] D. Xu, Y. Wang, P. Peng, S. Beilun, Z. Deng, H. Guo, "Real-time road traffic state prediction based on kernel-knn," *Transportmetrica A: Transport Science*, vol. 16, no. 1, pp. 104–118, 2020.

[21] A. Ding, X. Zhao, L. Jiao, "Traffic flow time series prediction based on statistics learning theory," in *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, IEEE, 2002, pp. 727–730.

[22] C.-H. Wu, J.-M. Ho, D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[23] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, L. D. Han, "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[24] R. Yasdi, "Prediction of road traffic using a neural network approach," *Neural computing & applications*, vol. 8, no. 2, pp. 135–142, 1999.

[25] M. S. Dougherty, M. R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *International journal of forecasting*, vol. 13, no. 1, pp. 21–31, 1997.

[26] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, no. 2, pp. 253–261, 2001.

[27] F. Moretti, S. Pizzuti, S. Panzieri, M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.

[28] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] Y. Tian, L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, IEEE, 2015, pp. 153–158.

[30] D. Kang, Y. Lv, Y.-y. Chen, "Short-term traffic flow prediction with lstm recurrent neural network," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, IEEE, 2017, pp. 1–6.

[31] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[32] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[33] R. Fu, Z. Zhang, L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, 2016, pp. 324–328.

[34] W. Cheng, Y. Shen, Y. Zhu, L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[35] G. Zheng, W. K. Chai, V. Katos, "A dynamic spatial–temporal deep learning framework for traffic speed prediction on large-scale road networks," *Expert Systems with Applications*, vol. 195, p. 116 585, 2022.

[36] Z. Wang, X. Su, Z. Ding, "Long-term traffic prediction based on lstm encoder-decoder architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561–6571, 2020.

[37] G. Khodabandelou, W. Kheriji, F. H. Selem, "Link traffic speed forecasting using convolutional attention-based gated recurrent unit," *Applied Intelligence*, vol. 51, pp. 2331–2352, 2021.

[38] M. Xu, H. Xiong, "Spatial-temporal transformer networks for traffic flow forecasting," *arXiv preprint arXiv:2001.02908*, 2020.

[39] J. Zhao, Z. Liu, Q. Sun, Q. Li, X. Jia, R. Zhang, "Attention-based dynamic spatial-temporal graph convolutional networks for traffic speed forecasting," *Expert Systems with Applications*, vol. 204, p. 117 511, 2022.

[40] C. Zheng, X. Fan, C. Wang, J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 1234–1241.

[41] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[42] J. Zhang, Y. Zheng, D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[43] D. Zang, Y. Fang, Z. Wei, K. Tang, J. Cheng, "Traffic flow data prediction using residual deconvolution based deep generative network," *IEEE Access*, vol. 7, pp. 71 311–71 322, 2019.

[44] Z. He, C.-Y. Chow, J.-D. Zhang, "Stcnn: A spatio-temporal convolutional neural network for long-term traffic prediction," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, IEEE, 2019, pp. 226–233.

[45] D. Zang, J. Ling, Z. Wei, K. Tang, J. Cheng, "Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3700–3709, 2018.

[46] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[47] T. N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[48] B. Yu, H. Yin, Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[49] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[50] J. James, C. Markos, S. Zhang, "Long-term urban traffic speed prediction with deep learning on graphs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7359–7370, 2021.

[51] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, P. Hellinckx, "A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.

[52] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, H. Xiong, "Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 547–555.

[53] Z. Zhang, Y. Li, H. Song, H. Dong, "Multiple dynamic graph based traffic speed prediction method," *Neurocomputing*, vol. 461, pp. 109–117, 2021.

[54] W. Huang, G. Song, H. Hong, K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[55] G. Zheng, W. K. Chai, V. Katos, "The sequence-to-sequence architecture with an embedded module for long-term traffic speed forecasting with missing data," in *2021 26th International Conference on Automation and Computing (ICAC)*, IEEE, 2021, pp. 1–6.

[56] M. Amiri, R. Jensen, "Missing data imputation using fuzzy-rough methods," *Neurocomputing*, vol. 205, pp. 152–164, 2016.

[57] Y. Tian, K. Zhang, J. Li, X. Lin, B. Yang, "Lstm-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[58] Z. Cui, L. Lin, Z. Pu, Y. Wang, "Graph markov network for traffic forecasting with missing data," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102 671, 2020.

[59] P. Wang, T. Hu, F. Gao, R. Wu, W. Guo, X. Zhu, "A hybrid data-driven framework for spatiotemporal traffic flow data imputation," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 343–16 352, 2022.

[60] X. Qiu, Y. Zhang, "A traffic speed imputation method based on self-adaption and clustering," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, IEEE, 2019, pp. 26–31.

[61] X. Chen, Z. He, J. Wang, "Spatial-temporal traffic speed patterns discovery and incomplete data recovery via svd-combined tensor decomposition," *Transportation research part C: emerging technologies*, vol. 86, pp. 59–77, 2018.

[62] J. Liu, P. Musialski, P. Wonka, J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2012.

[63] X. Chen, M. Lei, N. Saunier, L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 301–12 310, 2021.

[64] G. Chang, Y. Zhang, D. Yao, "Missing data imputation for traffic flow based on improved local least squares," *Tsinghua Science and Technology*, vol. 17, no. 3, pp. 304–309, 2012.

[65] Y. Li, Z. Li, L. Li, "Missing traffic data: Comparison of imputation methods," *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 51–57, 2014.

[66] I. Laña, I. I. Olabarrieta, M. Vélez, J. Del Ser, "On the imputation of missing data for road traffic forecasting: New insights and novel techniques," *Transportation research part C: emerging technologies*, vol. 90, pp. 18–33, 2018.

[67] J. Yoon, J. Jordon, M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International conference on machine learning*, PMLR, 2018, pp. 5689–5698.

[68] Y. Wang, D. Li, X. Li, M. Yang, "Pc-gain: Pseudo-label conditional generative adversarial imputation networks for incomplete data," *Neural Networks*, vol. 141, pp. 395–403, 2021.

[69] R. S. Ramhormozi, A. Mozhdehi, S. Kalantari, Y. Wang, S. Sun, X. Wang, "Multi-task graph neural network for truck speed prediction under extreme weather conditions," in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2022, pp. 1–11.

[70] N. Ronzoni, P. Goatin, "Road traffic data analysis: Clustering and prediction," Ph.D. dissertation, Inria; Unniversité Ctote d'Azur; CNRS; I3S, 2021.

[71] Z. Song, Y. Guo, Y. Wu, J. Ma, "Short-term traffic speed prediction under different data collection time intervals using a sarima-sdgm hybrid prediction model," *PloS one*, vol. 14, no. 6, e0218626, 2019.

[72] D. Jiang, C. Yao, Z. Xu, W. Qin, "Multi-scale anomaly detection for high-speed network traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 3, pp. 308–317, 2015.

[73] H. Rong, A. Teixeira, C. G. Soares, "Data mining approach to shipping route characterization and anomaly detection based on ais data," *Ocean Engineering*, vol. 198, p. 106 936, 2020.

[74] E. Joelianto, M. F. Fathurrahman, H. Y. Sutarto, I. Semanjski, A. Putri, S. Gautama, "Analysis of spatiotemporal data imputation methods for traffic flow data in urban networks," *ISPRS International Journal of Geo-Information*, vol. 11, no. 5, p. 310, 2022.

[75] W. C. Ku, G. R. Jagadeesh, A. Prakash, T. Srikanthan, "A clustering-based approach for data-driven imputation of missing traffic data," in *2016 IEEE forum on Integrated and sustainable Transportation Systems (FISTS)*, IEEE, 2016, pp. 1–6.

[76] D. Martin, A. Serrano, A. Bergman, G. Wetzstein, B. Masia, "Scangan360: A generative model of realistic scanpaths for 360° images," Mar. 2021.

[77] Y. Chen, X. M. Chen, "A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 143, p. 103 820, 2022.

[78] T. Ayar, F. Atlinar, M. A. Guvensan, H. I. Turkmen, "Long-term traffic flow estimation: A hybrid approach using location-basedtraffic characteristic," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 3, pp. 562–578, 2022.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. WoWMoM 2023 - Missing Traffic Speed Data Imputation Using Road Segment Characteristics for Long-Term Traffic Speed Prediction

   M. Mert KARA, H. İrem TÜRKMEN, M. Amaç GÜVENSAN

Projects

1. Tübitak 1001 Project - Uzun Vadeli Trafik Akış Hızı Tahmini için Çoklu Özellik Uzayı Tabanlı Bir Modelin Oluşturulması ve Gerçeklenmesi