PEAK-AWARE TRAFFIC PREDICTION WITH DEEP LEARNING MODELS AND A DRIVER SIMULATION METHOD WITH PROBABILISTIC HYBRID AUTOMATON

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATİH ACUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2022

Approval of the thesis:

PEAK-AWARE TRAFFIC PREDICTION WITH DEEP LEARNING MODELS AND A DRIVER SIMULATION METHOD WITH PROBABILISTIC HYBRID AUTOMATON

submitted by **FATİH ACUN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Assoc. Prof. Dr. Ebru Aydın Göl
Supervisor, **Computer Engineering, METU** ______________

Assoc. Prof. Dr. Sinan Kalkan
Co-supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU ______________

Assoc. Prof. Dr. Ebru Aydın Göl
Computer Engineering, METU ______________

Assist. Prof. Dr. Cemil Zalluhoğlu
Computer Engineering, Hacettepe University ______________

Date: 29.07.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Fatih Acun

Signature :

ABSTRACT

PEAK-AWARE TRAFFIC PREDICTION WITH DEEP LEARNING MODELS AND A DRIVER SIMULATION METHOD WITH PROBABILISTIC HYBRID AUTOMATON

Acun, Fatih
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Ebru Aydın Göl
Co-Supervisor: Assoc. Prof. Dr. Sinan Kalkan

August 2022, 43 pages

Accurately predicting traffic is crucial due to its impact on urban life in many aspects. Several statistical methods, machine learning, and deep learning approaches are applied to different traffic datasets. In general, traffic follows a stable behavior except for the morning and evening peaks which span a small period in time. To consider a prediction model to be accurate, it must demonstrate successful results during peak hours. In this thesis, a novel distance to mean weighting technique is presented that can be applied to any deep learning model by introducing a minor change in the loss function. The method also makes it possible to evaluate the performance of the models during peak hours in traffic. Traffic prediction is frequently used in estimating travel times and energy consumption (battery or fuel). In this thesis, in order to estimate both travel time and consumption, a driver simulation model based on Probabilistic Hybrid Automata is developed. The simulator generates speed-time data with respect to the given traffic and driver characteristics. The result is then used to estimate the consumption using an electric vehicle simulator.

Keywords: traffic prediction, deep learning, driver modeling, probabilistic hybrid automata

ÖZ

DERİN ÖĞRENME MODELLERİ İLE YOĞUNLUK YÖNELİMLİ TRAFİK TAHMİNİ VE OLASILIKSAL HİBRİT OTOMATA İLE BİR SÜRÜCÜ SİMULASYON YÖNTEMİ

Acun, Fatih
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Ebru Aydın Göl
Ortak Tez Yöneticisi: Doç. Dr. Sinan Kalkan

Ağustos 2022, 43 sayfa

Trafiği doğru bir şekilde tahmin etmek, birçok yönden kentsel yaşam üzerindeki etkisi nedeniyle önem arz etmektedir. Farklı trafik veri kümelerinde çeşitli istatistiksel yöntemler, makine öğrenimi ve derin öğrenme yaklaşımları uygulanmıştır. Genel olarak trafik, küçük bir zaman dilimine yayılan sabah ve akşam yoğunlukları dışında sabit seyreden bir davranış sergiler. Bir tahmin modelini yüksek doğruluklu olarak kabul etmek için yoğun saatlerde başarılı sonuçlar göstermesi gerekir. Bu tezde, kayıp fonksiyonunda küçük bir değişiklik ekleyerek herhangi bir derin öğrenme modeline uygulanabilecek yeni bir ortalamaya uzaklık ağırlıklandırma tekniği sunuyoruz. Yöntem, modellerin performansının yoğun saatlerde değerlendirilmesini de mümkün kılmaktadır. Trafik tahmini, yolculukların seyahat sürelerinin kestirimi ve araçların enerji (batarya ya da yakıt) tüketimlerini kestirmek amacıyla sıklıkla kullanılmaktadır. Bu tez kapsamında, seyahat süreleri ve yakıt tüketimini tahmin etme amacıyla Olasılıksal Hibrit Otomata tabanlı bir sürücü simülasyon modeli geliştirilmiştir. Bu simülatör, girdi olarak verilen trafik hızı ve sürücü özelliklerine göre hız-zaman verisini çıktı olarak üretir. Sonrasında bu çıktı elektrikli araç simulatöründe tüketimin kestirilmesi amacıyla kullanılır.

Anahtar Kelimeler: trafik tahmini, derin öğrenme, sürücü modelleme, olasılıksal hibrit otomata

To my family

ACKNOWLEDGMENTS

Firstly, I would like to thank to my supervisors Asst. Prof. Dr. Ebru Aydın Göl and Assoc. Prof. Dr. Sinan Kalkan for their guidance and support.

I am beyond grateful to our undergraduate research fellows Alp Eren Yılmaz and Mustafa Orhun Köse, for their considerable contribution to this study. I am thankful to my friends from the office A206 for their kind support and friendship.

This study is funded by the Research Council of Turkey (TUBİTAK) in the scope of the project titled "Modeling of Electric Vehicle Battery Use for Micro Grid Applications and Estimating the Energy Requirement (119E209)". I was also awarded the "TUBİTAK 2210-National Scholarship Programme for MSc Students" during my master's degree. I would like to remark my special thanks to TUBİTAK for giving me access to TUBİTAK ULAKBIM, High Performance and Grid Computing Center (TRUBA).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
GPS	Global Positioning System
EV	Electric Vehicle
V2G	Vehicle to Grid
5D	5 Dimensional
HA	Historical Average
ARIMA	Auto Regressive Moving Average
PeMS	Caltrans Performance Measurement System
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
GNN	Graph Neural Network
TGCN	Temporal Graph Convolutional Network
DTM	Distance to Mean
PHA	Probabilistic Hybrid Automata

CHAPTER 1

INTRODUCTION

Due to the increase in urbanization around the world, the problem of heavy traffic is becoming more critical in the last decades. Crowded cities are suffering from a variety of complications such as air and noise pollution due to high traffic loads. Local authorities and governments are taking actions to manage and attenuate the adverse effects of traffic. Besides its macro-scale effects, trip planning is getting essential for better time management and minimizing energy/fuel consumption for individual drivers. In addition to the aforementioned stakeholders, researchers also paid attention to this problem. In particular, computer science and related fields have found many research and engineering problems in the traffic domain, such as forecasting and simulating the traffic, driver characterization and modeling, autonomous driving, trip planning, and optimization.

Traffic information can be collected via different methods. Some traffic networks are equipped with loop detectors [3] that measure various metrics, and the collected data can be represented as density, flow, and average speed of traffic after preprocessing the raw measurements. Other than loop detectors, GPS data retrieved from a sufficiently large set of vehicles can be aggregated to macro-level traffic information such as average speed, flow, and density. Having the historical data in a processed form allows researchers to work on the problems such as traffic forecasting. In literature, various prediction methods are suggested to perform long and short-term predictions with a horizon from 30 minutes to a week. However, accurate predictions for peak hours in traffic are not addressed by the studies. In this thesis, we propose a novel peak-aware traffic prediction method specialized for accurate forecasts during peak hours in short-term predictions. Also, we developed statistical models to be able to make predictions in long-term periods.

Modeling individual drivers is another problem addressed in the traffic domain. Developing models that simulate drivers has challenges, such as representing the aggressiveness levels of drivers and covering the stochasticity due to the environment and the driver. Successful driver models are helpful for predicting energy consumption and estimating the arrival time of trips. The state of the traffic along a trip is highly decisive on a driver's driving boundaries. This relevance makes it inevitable to utilize traffic prediction within driver simulation models. In this thesis, we introduce a driver simulation model by considering the aggressiveness level of the driver and environmental conditions such as traffic speed and road types. The proposed driver simulator allows us to investigate the effect of traffic on the energy consumption of individual drivers.

We use our traffic prediction model and driver simulator to estimate the battery consumption of an electric vehicle (EV) in an interdisciplinary research project titled "Modeling of Electric Vehicle Battery Use for Micro Grid Applications and Estimating the Energy Requirement." The project is funded by the Research Council of Turkey (TUBİTAK), and conducted in cooperation with the Electrical and Electronics Engineering and the Civil Engineering Departments at Middle East Technical University. In the scope of the project, it is aimed to integrate the EV into a Vehicle to Grid (V2G) system of a micro-grid and allow the drivers effectively plan the battery consumption, sharing, and charging process. Figure 1.1 illustrates the components of the project.

1.1 Related Work

1.1.1 Traffic Prediction

Traffic prediction problem is considered one of the major time series prediction tasks. Initially, statistical prediction approaches are applied to this task. Historical average (HA), Auto Regressive Moving Average (ARIMA), and Kalman Filter are known to be the first models applied on traffic data and reveals successful results compared to

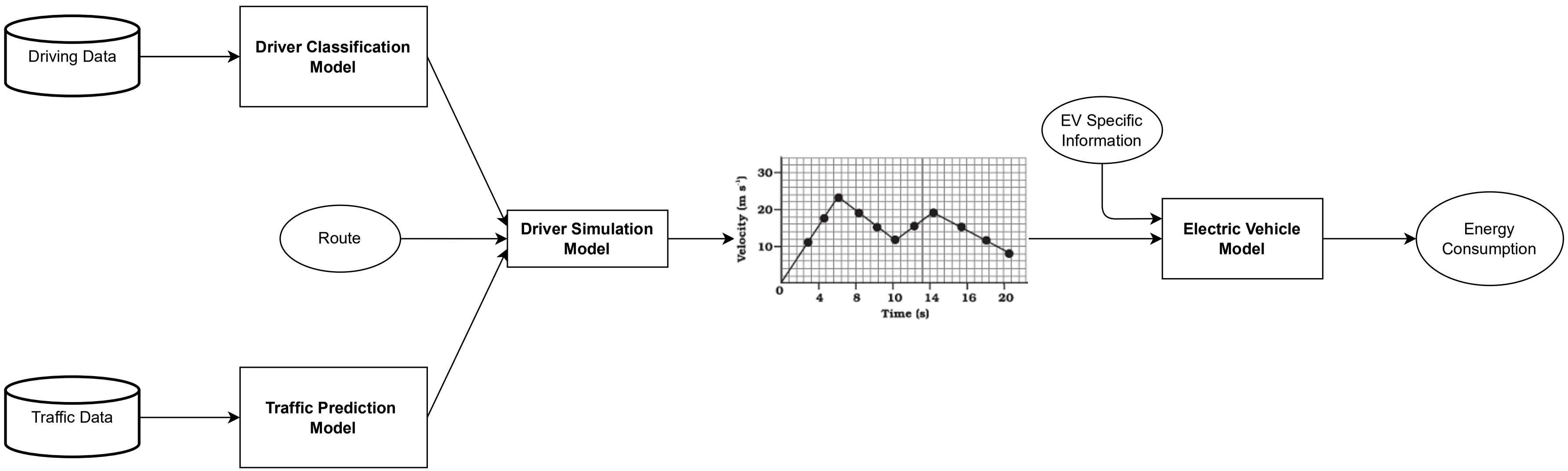


Figure 1.1: Modeling of Electric Vehicle Battery Use for Micro Grid Applications and Estimating the Energy Requirement

their simple nature [4], [5], [6].

Due to the nonlinear characteritics of the traffic data, machine learning models with higher capacity are found to be suitable. Support Vector Regression model is used to predict travel time in Tapei, Taiwan [7]. *Feng et al.* [8] make use of Adaptive Multi-Kernel Support Vector Machines by modelling a system that accouts for spatial-temporal correlation information and perform short-term predictions on traffic flow data. *Zhang et al.* [9] propoese a K-nearest Neighbor nonparametric regression model for short term prediction in Shangai. Various applications with variants of Artificial Neural Networks on traffic data have been studied by researchers [10], [11], [12].

Variety of deep learning approaches are applied to traffic prediction problem since the traffic data shows complex spatial and temporal characteristics which can be adressed by deep learning architectures. One of the earliest applications by *Lv et al.* [13] use stacked auto encoders to predict traffic flow on the traffic data collected from the Caltrans Performance Measurement System (PeMS). Convolutional Neural Networks (CNN) are used in traffic prediction due to their ability to capture spatial dependencies. [14] converts the traffic data to image format and uses CNN's to perform traffic prediction. Another study with CNNs improves the performance by using additional short and long term temporal features to represent periodicity [15]. Recurrent models and their variants such as Long-Short Term Memory (LSTM) and Gated Recurring Unit (GRU) are applied to the traffic data to capture the temporal behaviour. *Tian et al.* [16] use LSTM to perform short term traffic prediction on well-known PeMS data. *Fu et al.* [17] apply GRU for the first time for traffic flow prediction and compares the results with LSTM and ARIMA models.

Recently, hybrid approaches that aim to capture spatio-temporal dynamics by combining convolutional and recurrent models have become common. An hybrid architecture named Spatio-Temporal Dynamic Network (STDN) that focuses on the dynamic spatial dependencies and shiftings in temporal periodicity is proposed in [18]. *Zonoozi et al.* [19] propose Periodic Convolutional Recurrent Network that is based on the Convolutional GRU and uses periodic representation dictionaries to represent the multiple periodic patterns in traffic data.

CNNs capability is proved on grid structures such as images but traffic networks form a graph structure and converting those to grid-like inputs causes an information loss. In order to overcome this difficulty, Graph Neural Networks (GNN) are used to represent the traffic networks using the connectivity information of road segments. A comprehensive survey of GNNs application on the traffic prediction problem is done by Jiang and Luo [20]. The survey includes variety of datasets and methods. *Yu et al.* [21] use spatial graph convolution with temporal gated convolution and propose a stacked architecture of spatio-temporal blocks. Another spatio-temporal deep architecture with graph convolution and GRU is proposed in [1] and this architecuture is extended by introducing attention mechanism at [22] to give more importance on different time points. A novel graph convolution method called Traffic Graph Convolution (TGC) is proposed by *Cui et al.* [23]. TGC is based on a free-flow reachable matrix. The authors combined TGC with LSTM in their final model.

Latest research incorporates Generative Adversarial Networks (GANs) for traffic prediction. In their study, *Jin et al.* utilize Wasserstein Generative Adversarial Nets for modelling road link features and use it together with RNN and graph convolution [24].

1.1.2 Modeling the Driver Behavior

Driver behavior modeling is needed for the scenarios such as lane changing, intersection decision making, driver profiling, and router choice modeling. The demand for the development of behavior models for drivers applied to different disciplines such as energy efficiency and driver assistance systems [25].

In their work, *Schwarze et al.* [26] propose a hybrid automata model to simulate different behaviors of drivers for a merge to the highway from an on-ramp junction. They designed an automaton for this scenario. The states the automaton represent different stages of the merging event and the authors optimized different pre-defined functions for each stage.

Several studies for vehicle speed prediction on a given route rely on non-parametric data-driven models. In [27], the authors provide a deep learning model called Neuro-Fuzzy Inference System. They trained different models based on driver classes such as conservative and aggressive. They also considered road geometry and weather conditions as input to their model. Another data-driven study proposes Hidden Markov Models to present the statistical relationship between individual vehicle speeds and traffic speed [28]. They've used 3000 vehicle traces to train and 2000 traces to test their model. Making short term predictions of vehicle speed is also studied in the literature to optimize the energy consumption and power management of vehicles. *Özgüner et al.* [29] introduce a lightweight and fast short term vehicle speed prediction method using Auto Regression and Markov models.

The problem of driver modeling fits in the Markov Decision Process framework in which an agent takes actions at discrete time steps based on the state of the environment. There are some studies in the Reinforcement Learning field for driver modeling. *Bacchiani et al.* [30] use the Deep reinforcement learning algorithm Asynchronous Advantage Actor-Critic with a reasonable reward shaping that manipulates the resulting behavior of the driver under the scenario of entering a roundabout.

1.2 Motivation and Problem Definition

1.2.1 Traffic Prediction

Various traffic datasets from different locations in the world represent similar characteristics, such as specific peak periods that occur one or two times a day. Those peaks in traffic allocate a small period in time, and this introduces a data imbalance problem. Apart from the limited peak hour intervals, traffic in most of the road segments stays steady around its free flow speed. Figure 1.2 shows one-day traffic speed data between 06:00 and 22:00 from a road segment in Ankara, Turkey. The evening peak is observed for a 2 hours interval. The daily data is 16 hours in total and the peak period corresponds to only 12.5% of it. Due to this imbalance problem in the traffic data, prediction models seem to reveal successful results even though the predicted speeds are not close to the actual ones during peak hours. Table 1.1 shows the peak performance, and overall performance results of the Temporal Graph Convolutional Network (TGCN) [1] trained with the traffic dataset of Ankara, Turkey. It is seen that the errors during peak hours are much higher than the overall results. To the best of our knowledge, current traffic prediction models are not specialized for accurate predictions for peak hours in traffic. However, peak predictions are extremely important and must be accounted for specially. Our method, distance to mean weighting, can be applied to the loss function of deep models, and it dynamically changes the value of the loss in peak hours and introduces more penalty for the prediction errors during these periods.

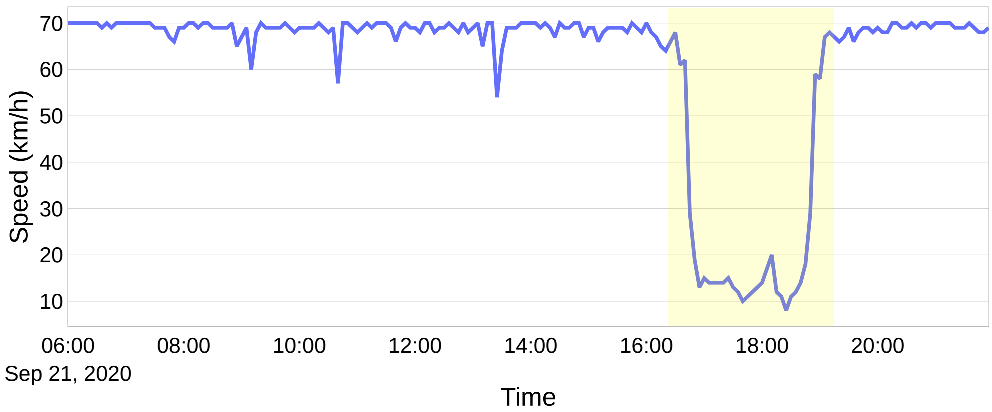


Figure 1.2: Imbalance problem in traffic data. One day traffic data is presented from a road segment. The peak period that occupies 12.5% of the day is highlighted with yellow.

Table 1.1: Results for the predictions of TGCN [1] in Ankara Dataset. "$(\cdot)^{p}$", stands for the errors during peak hours. There is a significant difference observed between the peak performance and overall performance.

Model	*Overall Performance*		*Peak Performance*	
	MAE $\downarrow$	**RMSE** $\downarrow$	**MAE**^{p} $\downarrow$	**RMSE**^{p} $\downarrow$
TGCN	4.9695	7.6333	28.3449	29.8887

1.2.2 Modeling the Driver Behavior

Driving simulation along a trip requires considering the driver behavior and environmental conditions that affect driving. Driver's travel speed, average traffic speed on the road, and the type of roads (highway, street, etc.) are some of the factors that play role on the acceleration/deceleration actions of drivers. Our approach produces the expected behavior as a time series speed data along a path. We use Probabilistic Hybrid Automaton to model the continuous and discrete dynamics of the driver and the environment. We represent driver speed as a continuous state and embed the discrete dynamics to the state space of the automaton that represents different acceleration/deceleration actions. To fit the probabilistic transitions between acceleration states, we used historical driving data that belongs to different driver classes.

1.3 Contributions

In the scope of this thesis, we propose novel methods for traffic prediction and driver simulation. Our contributions are as follows:

- Accurate traffic predictions for peak hours with distance to mean weighting to the loss function of deep learning models
- Peak aware traffic prediction in large scale traffic network of Ankara, Turkey
- Data driven driver simulation with Probabilistic Hybrid Automaton

Our study titled "Traffic Prediction with Peak-Aware Temporal Graph Convolutional Networks" was published at the conference 30^{th} Signal Processing and Communications Applications (SIU).

1.4 The Outline of the Thesis

The outline of the thesis is as follows. In Chapter 2, we present the details of Distance to Mean Weighting method for peak aware traffic prediction with deep learning models. In Chapter 3, we introduce our driver simulation method with Probabilistic Hybrid Automaton. Finally, we discuss the conclusions of this study in Chapter 4.

CHAPTER 2

PEAK-AWARE TRAFFIC PREDICTION WITH DISTANCE-TO-MEAN WEIGHTING FOR DEEP LEARNING MODELS

Having accurate predictions for traffic is critical for trip planning and traffic management systems for stakeholders to make micro- and macro-level decisions. In particular, peaks in traffic cause undesirable transportation delays and excessive energy consumption for vehicles. Therefore, developing successful models that are exceptionally accurate during peak hours is vital.

Traffic data is represented with different metrics such as density, average speed, and flow. Those metrics define the traffic state from different modeling perspectives and demonstrate both divergent and similar characteristics from case to case. The association between those metrics is investigated by various studies that originate from the fundamental relation of Greenshield [2], who is considered as the founder of traffic flow theory. Figure 2.1 shows the relations of traffic information metrics in different planes [31].

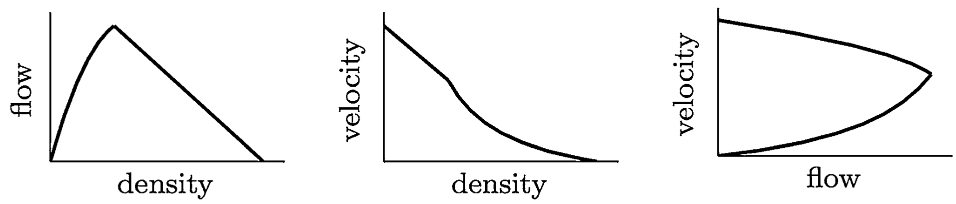


Figure 2.1: Fundamental relations of traffic in different planes. The relation between the traffic information metrics is demonstrated with respect to each other. Figure source: [2]

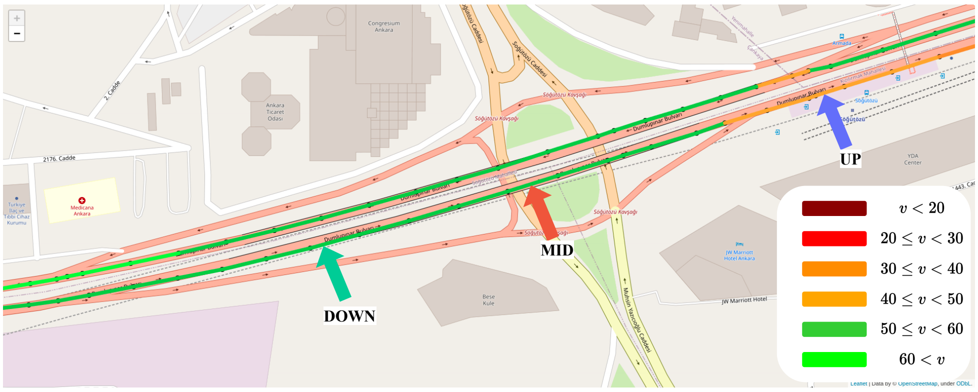
Traffic data consists of univariate time series with one of the metrics above. This series data shows certain characteristics such as seasonalities in daily and weekly periods, peaks in certain intervals, and unusual anomalies. Thus the prediction problem has a temporal aspect that shouldbe taken into account to make accurate predictions. On the other hand, traffic networks consist of many road segments that are connected to each other. This connectivity defines an implicit spatial correlation since traffic flow strictly depends on the road infrastructure. Queue formations and congestion propagations spread through road segments such that the neighbouring road segments are affected stepwise. For this reason, there is a spatial dependence between the road segments, and therefore, the models used for traffic prediction should capture the spatial characteristics of traffic data. Figure 2.2 and Figure 2.3 show how the traffic congestion propagates between segments close to each other in a 40 minutes period, in Söğütözü, Ankara.

One of the most important problems in traffic prediction is making predictions with high accuracy during peak periods. Since peaks span a small period in time compared to off-peak periods, an imbalance problem occurs in data as we illustrated in Figure 1.2, in Chapter 1. Also, error metrics that are calculated over the whole dataset are not reliable due to the mentioned imbalance problem. In this study, we've used a spatio-temporal deep learning model specialized for traffic prediction named Temporal Graph Convolutional Network (TGCN), proposed by *Zhao et al.* [1]. We introduce a Peak-Aware TGCN (pTGCN) model by proposing an extension on the loss function. We apply distance to mean weighting to the loss function so that the loss is dynamically increased during peak hours. Correspondingly, the prediction accuracy is enhanced during peaks.

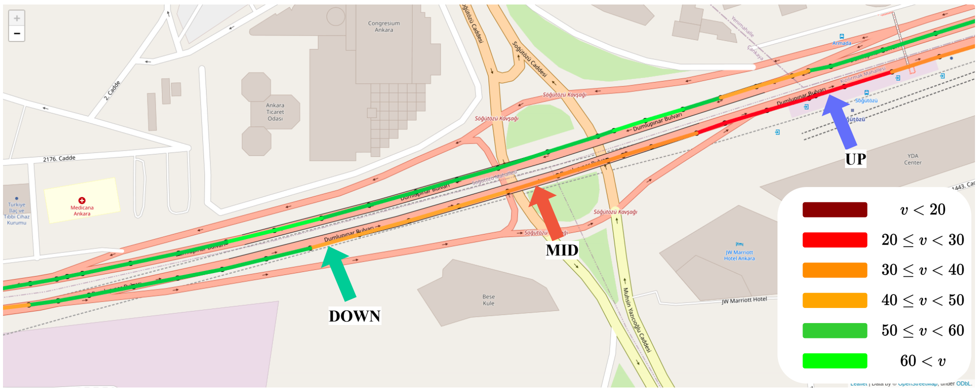
2.1 Background Information and Problem Definition

2.1.1 Traffic Prediction Problem

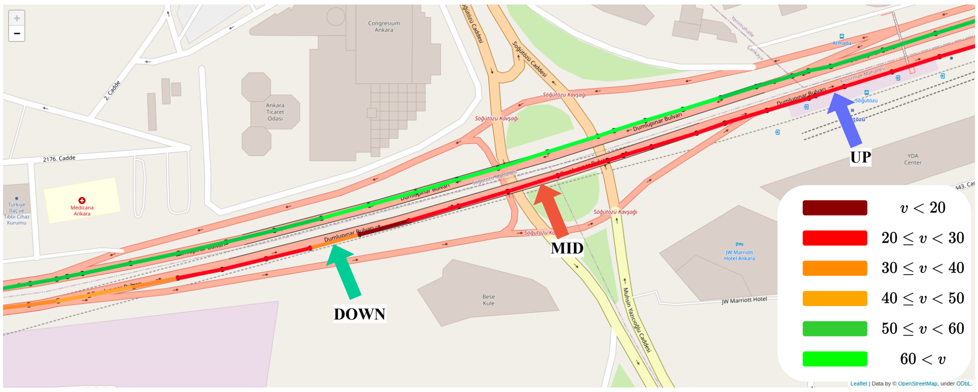
Traffic networks can be considered as a graph data structure. We define a graph over the traffic network $G = (V, E)$ such that V is the set of road segments and E is the set of connections between road segments. In particular, each vertice is considered as a road segment such that $V = \{v_1, v_2, ..., v_N\}$, where N is the number of road segments



(a) 8:00 AM



(b) 8:20 AM



(c) 8:40 AM

Figure 2.2: Spatial dependence of nearby road segments. Congestion propogation is observed in a 40 minutes period.

in the traffic network. Figure 2.4 demonstrates the representation of an example traffic network as a graph. In our study, we interpret the traffic data over the graph G as the

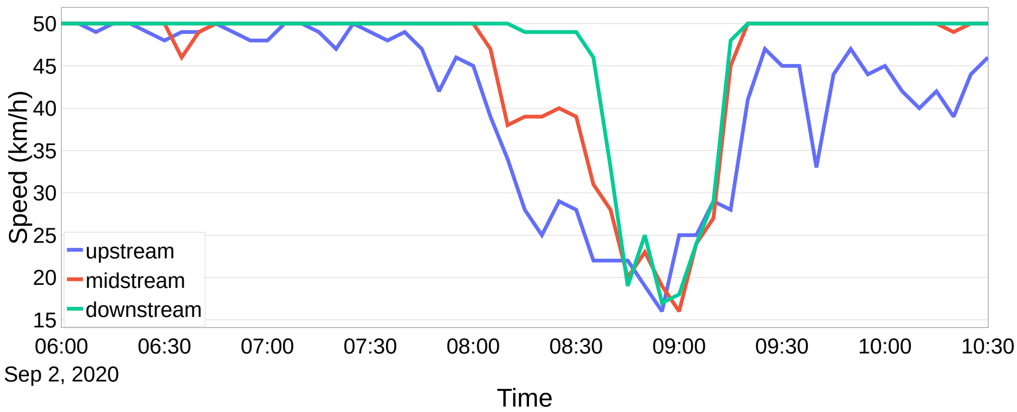


Figure 2.3: Traffic speeds of nearby road segments. Upstream, midstream and downstream road segments are labeled with UP, MID and DOWN in Figure 2.2. The peaks occur with lags in time in order with their spatial position.

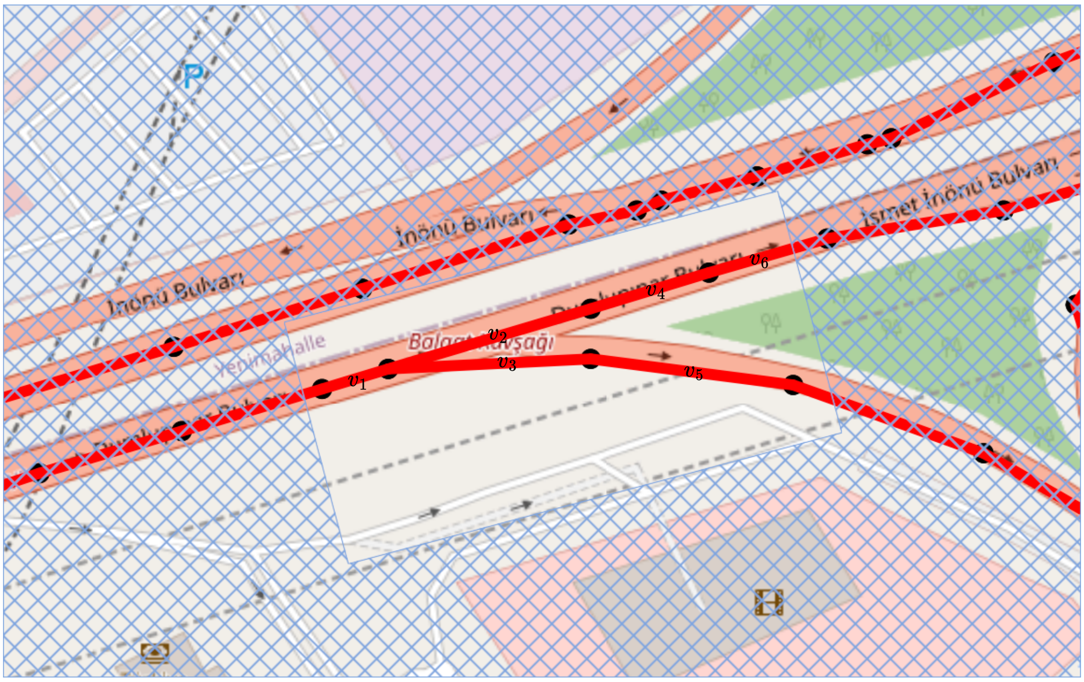
time series data of average speed values for each vertice. Therefore, the data matrix is $X^{N \times L}$, where N is the number of road segments, and L is the length of the time series data.

Let $X_t =< v_1, v_2, ..., v_N >$ be a vector that represents the state of the network as average speed values for all road segments at time t. Traffic prediction problem at time t is described as forecasting the future steps $(X_{t+1}, ..., X_{t+h})$ with a prediction horizon of h, using the input data $(X_t, X_{t-1}, ..., X_{t-s})$ of past time steps with a window of length s. Then, we define the traffic prediction problem with the function f that maps the input to the output with the weight parameters θ as follows:

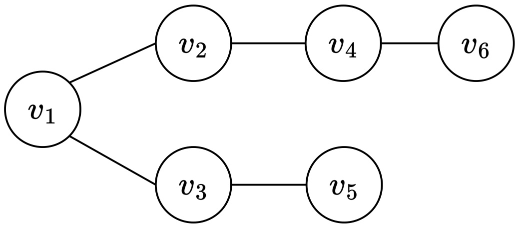
$$X_{t+h}, \ldots, X_{t+2}, X_{t+1} = f(X_t, X_{t-1}, \ldots, X_{t-s}; \theta). \tag{2.1}$$

2.1.2 Graph Convolutional Networks

Convolutional Neural Networks (CNNs) are one of the most popular deep learning architectures and are applied to grid-structured data such as images. However, data on a graph has irregularities and is not structured enough compared to grid-like data or any other data represented in a rectangular form. To address this problem, graph convolution is proposed, which is an extension of CNNs on graphs. There exists various methods, such as Graph Convolutional Networks (GCN) [32], Graph Attention



(a) Traffic Network



(b) Graph Representation

Figure 2.4: An example representation of roads for a small portion of a traffic network in 2.4a as a graph in 2.4b. Road segments shown in red lines are represented as the nodes of the graph. The connection between the road segments are demonstrated with the edges of the graph.

Networks (GAT) [33], Graph Sample and Aggregate (GraphSAGE) [34], and Graph Isomorphism Network (GIN) [35] that implement graph convolution with different approaches. In this study, we use GCNs to apply to the traffic data. GCN model is defined as stacked convolutional layers of the following:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}\theta^{(l)}), \tag{2.2}$$

where $\tilde{A}$ is the adjacency matrix with self connections, $\tilde{D}$ is the degree matrix, $H^{(l)}$ is the output of the layer l, $\theta^{(l)}$ is weights of layer l, and σ is the activation function sigmoid.

2.1.3 Recurrent Neural Networks

To capture the temporal dependencencies in the time series data, Recurrent Neural Network (RNN) architectures are utilized. Due to the problem of gradient vanishing in RNNs, Long Short Term Memory (LSTM) [36] and its variants, e.g. Gated Recurrent Units (GRU) [37], are preferred. Gated mechanisms in LSTM and GRU allow to memorize long-term information and reveal more successful results than vanilla RNNs. In our study, GRU is preferred to model temporal dependencies in traffic data due to its advantage of shorter training time compared to LSTM. The GRU model consists of reset and update gates and a candidate hidden state to decide which information is to be kept or forgotten. GRU is defined as follows [38]:

$$u_t = \sigma(X_t W_{xu} + H_{t-1} W_{hu} + b_u), \tag{2.3}$$

$$r_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r), \tag{2.4}$$

$$c_t = \tanh(X_t W_{xc} + (r_t \odot H_{t-1}) W_{hc} + b_c), \tag{2.5}$$

$$h_t = u_t \odot H_{t-1} + (1 - u_t) \odot c_t, \tag{2.6}$$

where u_t and r_T are the update and reset gates, c_t is the candidate hidden state and h_t is the hidden state. W represents the weight parameters, b represents the bias terms. σ is the sigmoid activation function and $\odot$ is the elementwise multiplication operator.

2.1.4 Temporal Graph Convolutional Network (TGCN)

TGCN [1] is a spatio-temporal deep neural network that consists of GRU and GCN models. In the architecture, the data over the traffic network graph at time t is given as input to the 2 layers of GCN defined as follows:

$$f(X, A) = \sigma(\hat{A} ReLU(\hat{A} X W_0) W_1), \tag{2.7}$$

where X is the input, A is the adjacency matrix, $\hat{A}$ is the preprocessing step such that $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, W is the weight matrix, $ReLU$ is the Rectified Linear Unit [39] activation function. The output of this 2 layer GCN is then fed to the recurrent model built with GRUs. Figure 2.5 illustrates the model architecture.

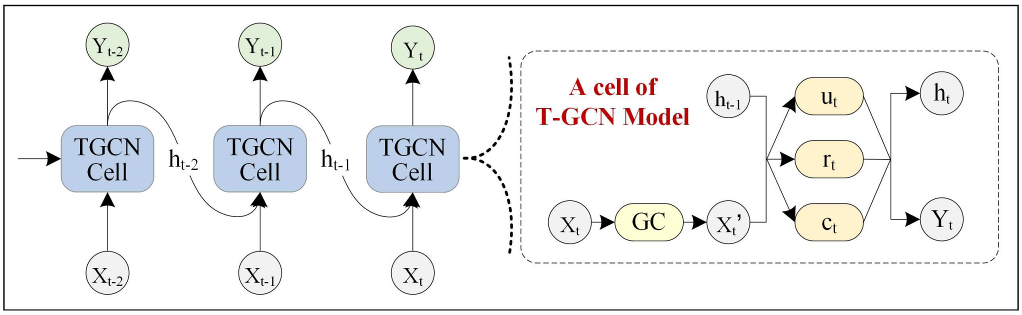


Figure 2.5: TGCN model architecture [1]. The utilization of Graph Convolution (GC) and GRU are illustrated. Figure source: [1]

2.2 Temporal Embedding

Peaks occur at certain time intervals in traffic. The day of the week, hour, and minute are specific indicators of dense traffic periods, and similar traffic behavior seasonally repeats itself. Therefore, we use time information as an additional feature along with the univariate time series data to address this dependence on the time and represent the seasonal characteristics in the traffic data. The indices of day, hour, and minute are processed with periodic sine and cosine functions as follows to encode the time data;

$$\phi_{sin} = \sin\left(2\pi \times t_i \times \frac{1}{l}\right), \tag{2.8}$$

$$\phi_{cos} = \cos\left(2\pi \times t_i \times \frac{1}{l}\right), \tag{2.9}$$

where t_i is the timestep index, l denotes the period. Figure 2.6 shows one week normalized train data versus the generated temporal features for consequent days such that distinct days are represented with unique sine-cosine pairs.

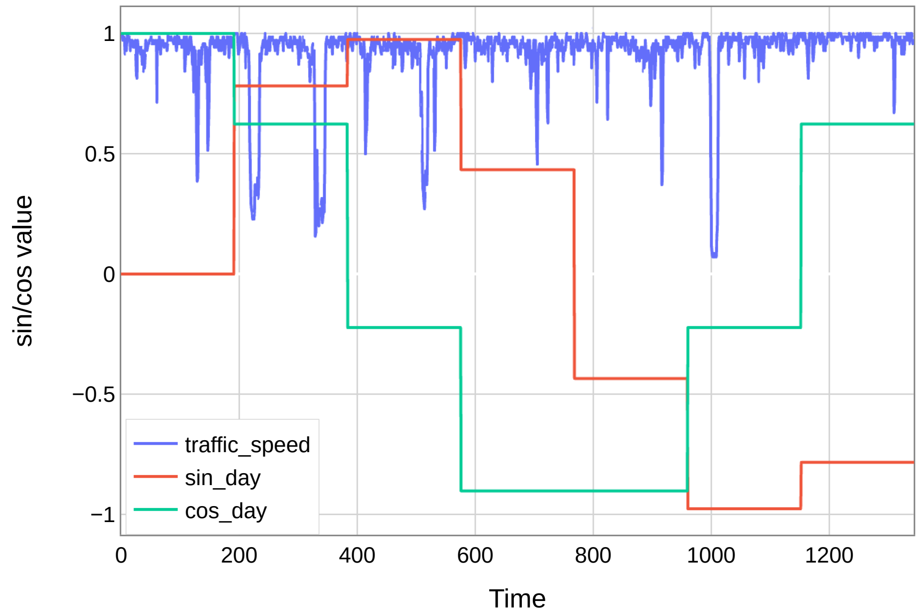


Figure 2.6: Temporal features generated for day indices. One week traffic data is shown in correspondance with the sine and cosine encodings of temporal information.

2.3 Distance-to-Mean Weighting for Improved Peak Prediction

Making accurate predictions for peak hours in traffic is crucial for prediction models. During the whole day, except for the morning and evening peak hours, the traffic speed for a road segment is mostly around its free-flow speed, that is, the average speed of the cars when the traffic flow rate is low to moderate [40]. Therefore, a data imbalance problem occurs as we illustrated in Figure 1.2 in Chapter 1, and models tend to learn the mean of traffic speed corresponding to the free-flow speed.

We mitigate this behavior by giving more importance to peak hours by multiplying the loss with distance-to-mean (DTM) weights that have higher values when the absolute distance of the traffic speed to its mean is high. We integrate the distance to mean weight matrix w_i^{DTM} to the loss function as follows:

$$\mathcal{L} = \frac{1}{h \times N} \sum_{i=1}^{h} w_i^{DTM} (\hat{y}_i - y_i)^2, \tag{2.10}$$

and we define w_i^{DTM} as follows:

$$w_i^{DTM} = \lambda \left(\delta + \frac{|y_i - \mu_s|}{y_{max}} \right)^{\tau}, \tag{2.11}$$

where y_i is the target for prediction step i, and μ_s is the mean speed for the corresponding road segment, and y_{max} is the maximum traffic speed value for the train dataset. The parameters λ, δ, and τ are scale, shift, and exponential factors, respectively. By introducing these parameters, we analyze the different forms of DTM function and have better control over the loss function. Figure 2.7 shows three different configurations of DTM parameters and presents the effect of different values for the exponential factor τ.

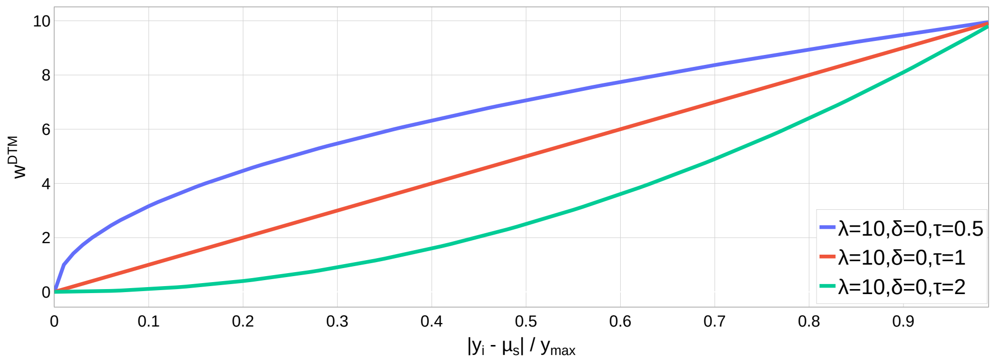


Figure 2.7: DTM parameter analysis. Values of w^{DTM} is shown with respect to $\frac{|y_i - \mu_s|}{y_{max}}$ with the different settings of DTM parameters λ, δ, and τ.

2.4 Experimental Setup

2.4.1 Datasets

2.4.1.1 Ankara Dataset

The dataset spans five weeks of average speed data in Ankara, Turkey, from August 24 to September 27, 2020. Data is collected for each minute from a large-scaled traffic network that consists of 6646 road segments having a length of 255 kilometers in total. We used linear interpolation to fill the missing values and aggregated the traffic data into 5-minute intervals by taking the average. Also, we filtered out the data between hours 22 and 6 since the traffic speed is simply constant at night and out of interest to perform predictions. In addition, we constructed a 6646×6646 adjacency matrix to represent the connections between the road segments. Highlighted parts in Figure 2.8 show the aforementioned traffic network. We reserved the last week of 5-week length Ankara dataset for the test set and separated the remaining 4 weeks to train, and validation splits with 80% and 20% ratio, respectively.

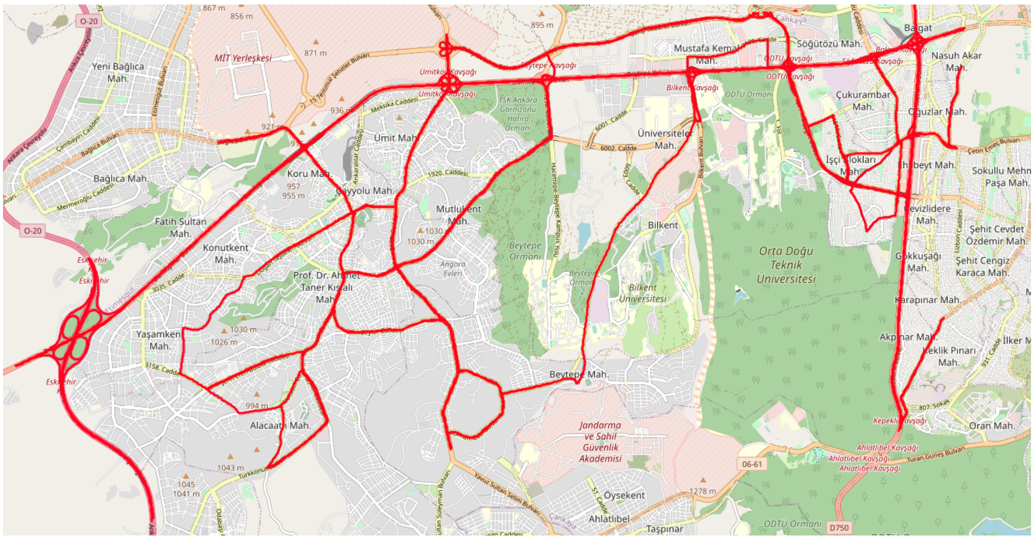


Figure 2.8: Traffic network of Ankara.

2.4.1.2 Los-loop Dataset

This dataset consists of average speed values in 5-min aggregates collected from the highway of Los Angeles County between March 1 to March 7, 2012. The data is collected from loop detectors at 207 points. In comparison to the Ankara dataset, this dataset is smaller in terms of its network size and time length. This dataset is provided in [1]. The dataset is splitted to train and test with a percentage of 80% and 20%, respectively.

2.4.2 Implementation and Training Details

To observe the impact of DTM weights in training, we've used the same configurations with the baseline method TGCN. The implementation by the authors can be found at (https://github.com/lehaifeng/T-GCN/). In their experimental setup, they set the learning rate to 0.001, the batch size to 32, and the number of hidden units to 64. They've trained their models for 5000 epochs. In order to observe the isolated impact of the DTM weights on the prediction accuracy, we've kept the parameter settings the same with TGCN. However, we've run experiments for learning rate tuning since we introduce a change in the loss function of the model. We set the prediction horizon to 30 minutes and the historical window to 1 hour for all experiment configurations.

2.5 Experiments

2.5.1 Experiment 1: Analysis with a Large-scale Dataset (Ankara Dataset)

Analysis of DTM values is the first stage of the experiments to verify and consolidate our hypothesis. We expect DTM weights to have higher values during peak hours to magnify the loss for those periods. Figure 2.9 shows the daily data from one road segment. During the evening peak, it is observed that values of DTM are higher compared to off-peak hours. Further analysis of DTM values over the train dataset of the whole traffic network is demonstrated in Figure 2.10. The histogram shows that DTM values are accumulated between 0-0.1 since the traffic data is close to its mean in most examples of the dataset. In other words, there is a significant imbalance between on-peak and off-peak samples.

The hyperparameter tuning experiments consist of tuning the learning rate for the base model TGCN and our method Peak-Aware TGCN (pTGCN) with the DTM extension. To see the isolated effect of DTM in the results, we've kept other parameters the same with the TGCN's configuration. The scale, shift and exponential parameters for DTM are set to $\lambda = 1$, $\delta = 0$, and $\tau = 1$. This setting of DTM parameters acts as a unit function since it does not change the value of normalized distance to mean values, i.e. $w_i^{DTM} = \left(\frac{|y_i - \mu_s|}{y_{max}}\right)$. For both models, results are presented for

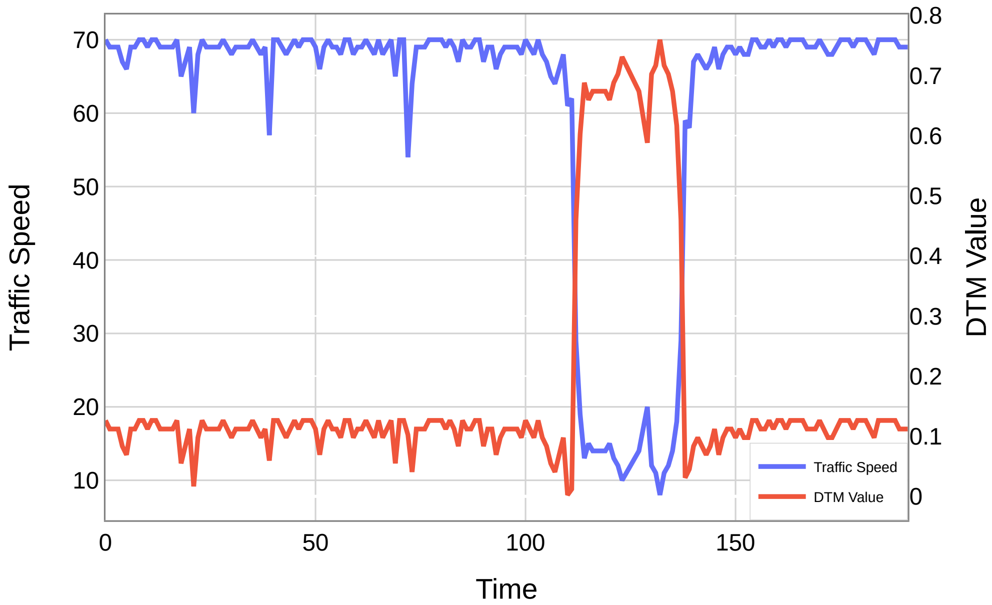


Figure 2.9: Distance to mean weights in the daily data of a road segment. Higher values for DTM are observed during the peak hours.

the optimal learning rates. The pTGCN models used are trained for 1500 epochs with a learning rate of 0.001, and the TGCN model is trained for 5000 steps with a learning coefficient of 0.01. Table 2.1 shows the results for the baseline models and our proposed method. The predictions of the deep learning models are compared with the ARIMA models (total of 6646 models), which are trained separately for each traffic segment, and the Historical Average (HA) models, which work by averaging the data over the same time periods in the past. The reason why the weekly forecasts of ARIMA and HA models are included in the study is that these models are more successful in long-term forecasting. In the short-term predictions of deep learning models, it is expected to outperform these long-term predictions. Consequently, the peak performances indicate that our model has significantly improved the results of the base model TGCN up to 14% and performs better than all the other models.

We calculated the peak errors "$(\cdot)^{\mathrm{p}}$" by filtering the dataset based on the distance to mean values such that $w^{DTM} \geq 0.2$, for $\lambda = 1$, $\delta = 0$, and $\tau = 1$. In consequence, we presume that 0.2 is a threshold for an instance to be accepted as in the peak period as also observed within Figure 2.9.

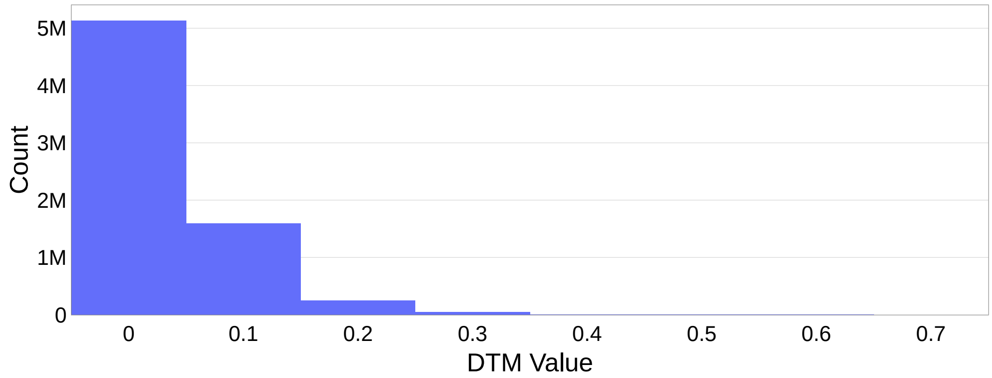


Figure 2.10: The histogram of distance to mean values for the Ankara dataset.

Table 2.1: Results for the prediction models. "$(\cdot)^{p}$" stands for the errors during peak hours.

Model	DTM	Temporal Embedding	*Overall Performance* MRE ↓	MAE ↓	RMSE ↓	*Peak Performance* MRE^p ↓	MAE^p ↓	RMSE^p ↓
ARIMA			**0.1599**	**4.7513**	7.6712	1.8467	28.6745	30.7418
HA			0.2664	6.7017	9.9407	2.0314	28.4094	31.3782
TGCN			0.1652	4.9695	**7.6333**	1.7887	28.3449	29.8887
		✓	0.1628	4.9550	7.6556	1.7908	28.5527	30.1190
	✓		0.2002	6.8128	8.9452	1.5531	24.4146	26.8029
pTGCN (Proposed Method)	✓	✓	0.2023	6.9218	9.0187	**1.5432**	**24.2929**	**26.6907**

In Figure 2.11, prediction results of different models for one road segment are visualized for one day from test data. The predictions for the peak period between 16:30 and 18:30 show that our pTGCN model approximates the actual traffic speed more precisely than the others. For the off-peak hours, it is observed that other models perform better.

2.5.2 Experiment 2: Analysis with a Small-scale Dataset (Los-loop Dataset)

The Los-loop dataset, which is a small-scale dataset compared to the Ankara dataset, is used in the experiments of the study TGCN [1]. In this section, we present our experiments on the Los-loop dataset with DTM extension on the model. In this dataset, our results for DTM can not outperform the TGCN model. However, our results are promising and considered to be successful enough to be compared with the TGCN

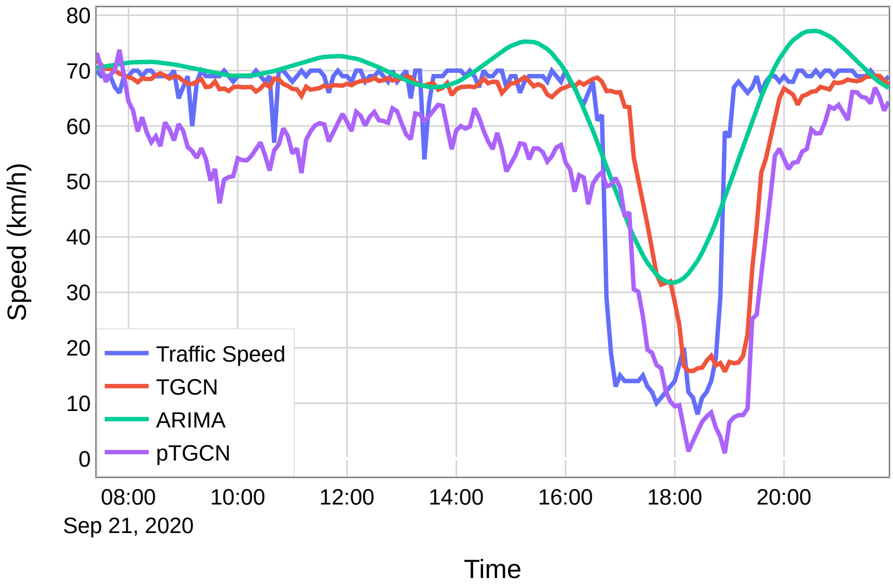


Figure 2.11: Prediction results for different models. It is shown that proposed method pTGCN performs better during peak hours.

model. Different from the experiments for the Ankara dataset, we present an ablation analysis for the parameters of DTM to outperform the TGCN model. For each experiment setup with different settings of DTM parameters λ, δ, and τ, we've analyzed the different forms of DTM function on the prediction accuracy. We present the results for optimal learning rates in Table 2.2.

2.6 Discussion

Experiments in Ankara and Los-loop datasets with our proposed method pTGCN show promising results. Especially in the Ankara dataset, we achieved a significant improvement for the predictions during peak hours. However, the overall prediction performance is decreased for the errors during off-peak hours. Although this is not a critical problem, we think this can be addressed by training ensemble models to have accurate predictions on both peak and off-peak hours. The other possible solution will be optimizing the DTM funciton to have a balanced impact on the peak and off-peak

Table 2.2: Results for the predictions of TGCN [1] in Los-loop Dataset. "$(\cdot)^{\text{p}}$" stands for the errors during peak hours. λ, δ, and τ are the scale, shift and exponent parameters of DTM, respectively.

Model	λ	δ	τ	**RMSE ↓**	**RMSE$^{\text{p}}$ ↓**
TGCN	-	-	-	**6.4155**	**7.7491**
pTGCN	1	0	0.5	7.5601	8.4070
	1	0	1	7.5548	8.4661
	1	0	2	7.5099	8.8513
	1	1	0.5	7.4479	8.1078
	1	1	1	7.1878	8.0554
	1	1	2	7.5531	8.2798
	10	0	0.5	7.5697	8.3853
	10	0	1	7.5424	8.3322
	10	0	2	7.1224	8.3157
	10	1	0.5	7.5846	8.4863
	10	1	1	7.2112	8.1337
	10	1	2	7.1571	8.2823

hours.

For the Los-loop dataset, our method pTGCN has revealed results that are close to the TGCN model. We presume this is encountered by the different characteristics and size of the dataset. The sharpness, distribution, and importance of the peaks over the datasets might differ. Further optimizations in pTGCN can be done after analyzing and having more powerful insights into the different datasets.

CHAPTER 3

A DRIVER MODELING AND SIMULATION APPROACH WITH PROBABILISTIC HYBRID AUTOMATA

The impact of the traffic state on a driver is enormously high. Environmental conditions such as heavy traffic and rainy weather introduce limitations and specific behavioral patterns in driving. In particular, a driver's actions are highly affected by the preceding driver in heavy traffic. This phenomenon is investigated in the literature with the car following models [41]. On the other hand, driving style is more decisive during off-peak hours. Aggressive drivers can drive at high speeds and/or make frequent acceleration and deceleration actions, while coutious drivers prefer to cruise at a constant speed with less bold moves.

One of the direct impacts of different driving practices is observed in the energy consumption of a vehicle. Aggressive driving with frequent acceleration and deceleration actions significantly increases consumption. On the contrary, cruising around at a constant speed ensures energy saving. Therefore, accurate simulations that approximate a driver's actual driving empower predicting the energy consumption.

In this chapter, we propose a driver simulation method based on Probabilistic Hybrid Automata (PHA). Our approach encapsulates the average traffic speed and road types to model the effects of traffic on the trips. Furthermore, it fits the driving characteristics from historical driving data to accurately emulate the driving behavior.

3.1 Background

3.1.1 Probabilistic Hybrid Automaton

PHA [42] is a stochastic automaton model with continuous and discrete variables. Continuous variables are stored in the locations (discrete states) of the PHA, and their change in time is expressed with differential equations.

Probabilistic edges of PHA allow the implementation of stochastic events. The state transitions are based on a probability distribution over the probabilistic edges. The formal definition of PHA is given with a tuple as follows;

$$PHA = (Loc, I_{init}, Act, X, flow, inv, prob, L), \tag{3.1}$$

where;

- Loc is the finite set of locations (states)
- $I_{init} \in Loc \times \mathbb{R}^X$ is the initial condition
- Act is a finite set of actions
- X is a finite set of continuos variables
- inv is the invariant condition, such that $Loc \rightarrow 2^{\mathbb{R}^X}$
- $flow$ is the flow condition, such that $(Loc \times R^X) \rightarrow \mathbb{R}^X$
- $prob$ is the probabilistic edge relation, such that
  $prob \subseteq Loc \times 2^{\mathbb{R}^X} \times Dist(Loc \times \mathbb{R}^X)$
- L is the labeling function, $Loc \rightarrow AP$, where AP is a set of atomic propositions

3.2 Driver Modeling and Simulation

In this section, we describe the utilization of the PHA model in the driver simulation problem.

The most critical metrics in the driving action are speed and acceleration. The change of those continuous variables in time constitutes a trip from the start to the destination. Using PHA, we store the speed of the driver as a continuous variable and acceleration as a discrete variable.

Each location of the PHA represents traffic speed (s), road class (rc), driver speed (v), and acceleration type (acc). In our traffic network, we have 3 different road classes that represents the different road types of a scale from the motorway to the roads with less capacity. In this study, we separated the traffic speed s and driver speed v into discrete intervals. After the discretization of those variables, the number of locations in the PHA is expressed as follows;

$$|Loc| = |s| * |v| * |rc| * |acc|, \tag{3.2}$$

where $|s|$, $|v|$ denote the number of discrete intervals of s and v, respectively. $|rc|$ is the number of road classes and $|acc|$ is the number of accleration types.

We map the locations of the PHA to the different acceleration types (acc) of drivers. Acceleration types can be defined as a scale of Hard Acceleration (HA), Acceleration (A), Cruising (C), Deceleration (D), and Hard Deceleration (HD). These acceleration types are built in a finer discretization in different experimental setups as follows;

$$acc = \{acc_0, acc_1, acc_2, ..., acc_n\}, \tag{3.3}$$

where acc_0 and acc_n refer to HD and HA, respectively, and the others are the intermediate steps of acceleration types, such that $acc_{n/2}$ refers to C. At time t, the PHA model calculates the values of acceleration and speed of the vehicle at $t+1$, depending on location of the PHA that stands for the current acceleration type.

The deterministic transitions between the locations of the automaton are controlled based on the invariants of traffic speed (s), driver speed (v), and road class (rc). s is the average speed of the traffic, v is the speed of the ego vehicle, and rc is the type of road. Those criteria directly influnce the acceleration characteristic and introduce limitations on a vehicle. To illustrate, it is not physically possible to perform HA when the driver speed v is high. Figure 3.1 demonstrates a slice from the state space of the PHA.

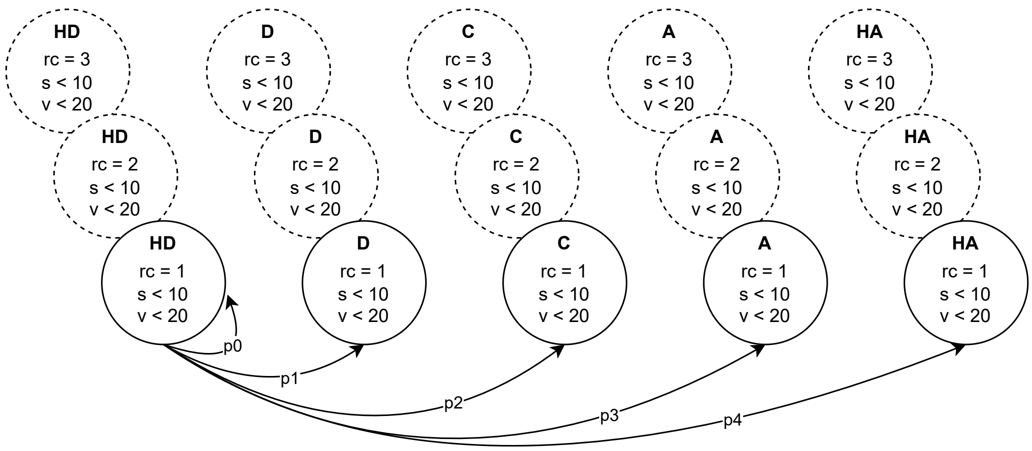


Figure 3.1: Visual illustration of a slice of the PHA. (For simplicity, probabilistic transitions outgoing from only one location are visualized)

The probabilistic transition of the PHA is defined with the function P as follows;

$$p = P(acc'|s, v, rc, acc), \tag{3.4}$$

such that p is the probability of the successor acceleration state (acc'), given the (s, v, rc, acc) tuple. The probability distributions for probabilistic edges are calculated using the sample observations in the driving dataset. The transition instances observed in the dataset are counted and normalized into the range of $[0, 1]$ to represent the probability distribution from one acceleration state to others. We store a vector for the probabilistic edge from each location. Stacking those vectors for the different values the invariants, in total, builds a 5D transition matrix with dimensions defined by (s, v, rc, acc, acc'). The transition matrix stores the probabilities of a transition for any given source-target acceleration states. Figure 3.2 shows a transition probability matrix from the source to destination acceleration states with given (s, v, rc).

3.3 Dataset Description

The driving dataset is GPS track data of trips in Ankara, Turkey. Those trips are anonymous and have no driver information. The attributes of the data are latitude, longitude, speed, and timestamp logged at irregular frequencies by vehicles. The frequency of data varies in different trips and also may change along one trip.

Figure 3.2: Sample transition probability matrix with given (s, v, rc). Matrix shows the transition probabilities from the source to destination acceleration type.

3.3.1 Data Preprocessing

We applied preprocessing steps to standardize the dataset and eliminate the spatial noise in GPS points. Firstly, to align the noisy points that are spread around 5-10 meters off from the roads, we applied an HMM-based map matching algorithm implemented by Open Source Routing Machine [43]. After this map matching operation, we achieved to fetch the related traffic speed of road segments that the GPS point is inside, based on time and location information.

Due to the missing GPS points in the trips and GPS loggers with varying irregular frequencies, the data need to be standardized along the time dimension. In order to have a standardized dataset with a GPS point every second, we applied a linear interpolation on the trips' path to fill the missing parts.

In addition to these preprocessing steps, we filtered out the too short and less reliable trips with a low percentage of GPS points. After having the data in a cleaned form, it is feasible to match the GPS points with the traffic data of the road segments. Using the timestamp and matched road segment information, we unified the traffic average speed data and GPS track data to be used in the PHA model. Finally, we have 58,091 trips after all preprocessing steps. In total, this corresponds to a 421 days length driving data.

3.4 Experiments

In this section, we use a driver classification model and EV model integrated with our PHA model. Those models are developed by İven Güzel and Berkay Sağlam from METU Electrical and Electronics Engineering department, under the supervision of Assoc. Prof. Murat Göl as a part of the TUBİTAK project mentioned in Chapter 1. Figure 1.1 explains the relation between those models and our PHA based driver simulation model. The driver classification model classifies the trips in our dataset in terms of their driving chracteristics. The EV model calculates the battery consumption of the electric vehicle for a trip, given the speed as time series input.

We implement the PHA model using PRISM model checker [44]. PRISM is a probabilistic model checking tool for formal modelling and analysis of systems that demonstrates random or stochastic characteristics. We build a PHA model for each driver class obtained by the driver classification model to distinguish between different driver characteristics in our simulations. The classification model identifies 6 different driver classes in our dataset. Therefore, we develop 6 different PHA models for those classes. We also split the dataset to train, validation, and test splits for each of those classes with the percentages of 70%, 15%, and 15%, respectively.

We divide the traffic speed into 5 intervals. The traffic speeds over 80 km/h are considered to be in the same discretization level as it indicates the free flow speed.

The discretization is defined by the following function D;

$$D(s) = \begin{cases} s_0, & s < 20 \\ s_1, & 20 \leq s < 40 \\ s_2, & 40 \leq s < 60 \\ s_3, & 60 \leq s < 80 \\ s_4, & 80 \leq s \end{cases} \tag{3.5}$$

The driver speed (v) is discretized into the 7 intervals defined by the function D as follows;

$$D(v) = \begin{cases} v_0, & v < 20 \\ v_1, & 20 \leq v < 40 \\ v_2, & 40 \leq v < 60 \\ v_3, & 60 \leq v < 80 \\ v_4, & 80 \leq v < 100 \\ v_5, & 100 \leq v < 120 \\ v_6, & 120 \leq v \end{cases} \tag{3.6}$$

where, v_i indicates the discretization level.

We apply two different discretization setup on the acceleration types. We present experiments with the number of acceleration types 5 and 9 to see the effect of having finer discretizations of the acceleration on the simulations. Those set of accelerations are referred as acc_f, and acc_c in our experiment setups, that represent the fine and coarse sets of acceleration types, such that $|acc_f| = 9$, and $|acc_c| = 5$. The acceleration values (m/s) for those sets are as follows;

$$acc_c = \{-2, -1, 0, 1, 2\}, \tag{3.7}$$

$$acc_f = \{-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2\}. \tag{3.8}$$

Our model outputs a speed data versus time and the resulting simulation is not necessarily to be the same length with the original trip. Depending on the driving speed, the duration of the simulation can vary and might be longer or shorther than the trip. This prevents to use error metrics that compare the simulation results with the original data as a one-to-one correspondance. Since we aim to approximate the energy consumption of the trip at the final step such an error metric is not informative. Therefore, we use a histogram distance metrics based on the acceleration histograms of trips and simulations to evaluate our model on the acceleration values of the simulation and trips. We use Earth Mover's Distance (EMD) [45] and and the Normalized Histogram Distance (NHD) to compare trip and simulation acceleration histograms. We define NHD as follows;

$$NHD = \Sigma_i |h_i^t - h_i^s|, \tag{3.9}$$

where h^t and h^s are the normalized histograms of acceleration values in the actual trip and simulation respectively. We normalize the histograms to make our error metrics independent from the length of the trips.

We present the results for the experiments with acc_f, and acc_c settings using the NHC, EMD error metrics. We also provide metrics considering the error in the energy consumption of the trips. We specify Mean Absolute Error (MAE), which is the absolute difference between the trip and simulation energy consumption values in kWh per 1 km. The last error metric Capacity Percentage Error (CPE) represents the percentage of the MAE with respect to the 37.9 kWh total usable battery capacity of the EV BMW i3. CPE metric is given for better comparision of the error in MAE as a proportion to the battery capacity. The results show that acc_f setting exposes better results for the NHD and EMD metrics. On the other hand, with acc_c configuration, the battery consumption errors MAE and CPE are improved.

Table 3.1: Error metrics for different acceleration configurations. It is shown that with finer accelereation setup acc_f, more successful results are obtained.

Configuration	**NHD** $\downarrow$	**EMD** $\downarrow$	**MAE (kWh)** $\downarrow$	**CPE (%)** $\downarrow$
acc_f	**0.349**	**0.798**	0.05	0.133
acc_c	0.477	0.923	**0.04**	**0.106**

Finally, we present our simulation results with respect to the actual trips and traffic average speeds. We demonstrate a sceneario from a trip that we selected from the 0^{th} driver class. We run 3 different simulations with the PHA models trained for the 0^{th}, 1^{st}, and 3^{rd} driver classes. The experiments show that the PHA model trained for the 0^{th} class better approximates the actual trip. This verifies our approach for modeling different driver classes with different PHA models.

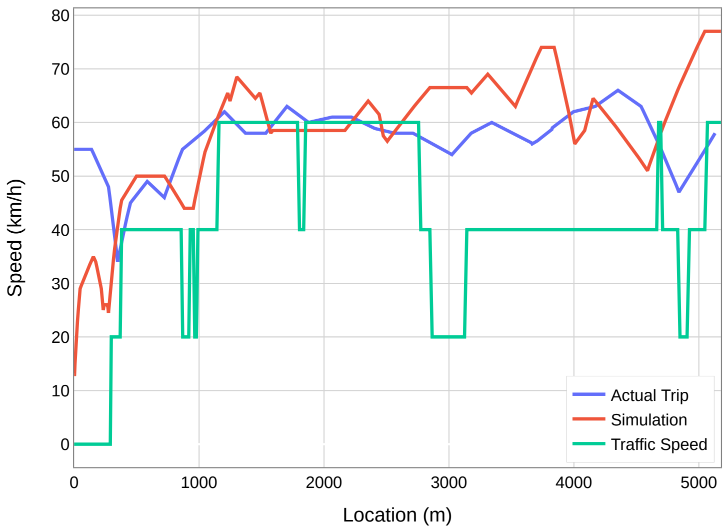


Figure 3.3: Simulation results for driver class 0

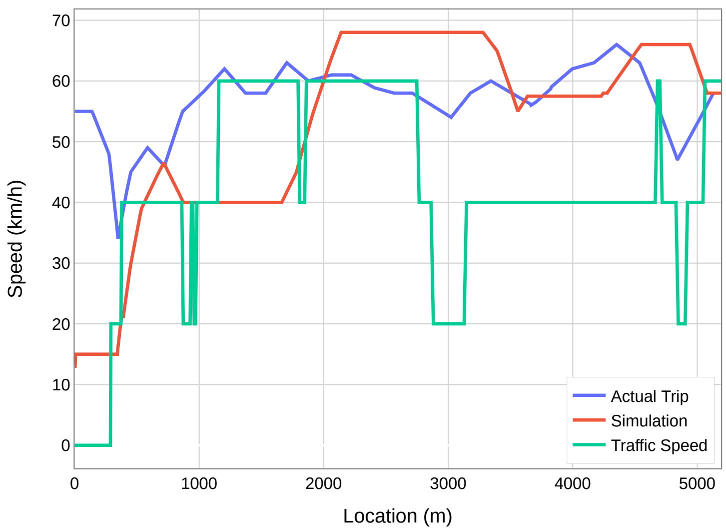


Figure 3.4: Simulation results for driver class 1

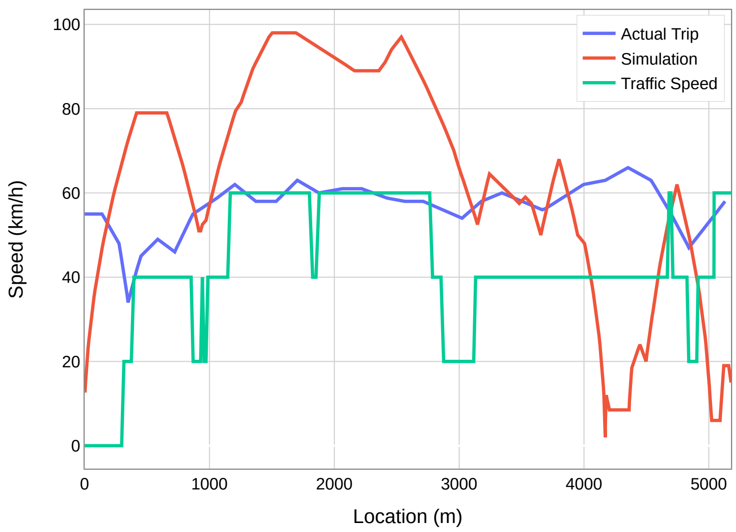


Figure 3.5: Simulation results for driver class 2

3.5 Discussion

In this section, we propose a data driven PHA model for driver simulation by considering the environment dynamics such as road types, and traffic speed. We use historical driving data of 6 different driver classes to fit probabilistic acceleration state changes of the drivers and represented those with the probabilistic edges of the PHA. To represent the continuous aspects in driving such as speed and acceleration, we utilize the continuos variables of PHA. We build 6 PHA models for each driver class to distinguish between the different driving characteristics. We evaluated our results based the errors in the acceleration histograms of trips and simulations, and the energy consumptions using the EV model.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

In this thesis, we proposed a peak-aware traffic prediction method, named pTGCN, for deep learning models, and a novel driver simulation technique with Probabilistic Hybrid Automaton.

We underlined that traffic prediction models must reveal accurate results during the peak hours and pointed out evalutaion of the models might be misinterpreted due to the imbalance problem in traffic data. Out method concentrates on making accurate predictions on peak hours by signifying the values of the loss function for those periods during training with DTM weighting. Our approach also empowers to evaluate the prediction quality during the peak hours in traffic.

Our driver simulation method with PHA outputs the speed of a vehicle along a trip. Being a data driven model makes it able to reflect the real-life driver characteristics. On the other hand, using a PHA as the method brings in a transparent model with explainable state transitions. Our model also comprises the environmental factors in traffic such as traffic speed and road type. The output of the model is then used to estimate the energy consumption of a trip.

4.2 Future Work

We observed that, the prediction accuracy of the pTGCN model is decrased during the off-peak hours. To prevent this, further optimizations on the loss function is found to be a possible solution. Another alternative approach to solve this problem will be using an ensemble model to perform better on both peak and off-peak hours.

Since we have presented promising results with pTGCN model, we found it necessary to apply DTM to different traffic datasets and also other time series prediction tasks from different domains.

For some simulations with PHA model, we observed that unrealistic behaviour can occur such as too high driving speeds. As a future work, reasons and results of this unrealistic cases should be investigated. Also, the PHA model is open to further advancements and extensive experiments with different discretization setups.

REFERENCES

[1] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[2] B. D. Greenshields, J. Thompson, H. Dickinson, and R. Swinton, "The photographic method of studying traffic behavior," in *Highway Research Board Proceedings*, vol. 13, 1934.

[3] J. H. Kell, I. J. Fullerton, and M. K. Mills, "Traffic detector handbook," tech. rep., 1990.

[4] Y. J. S. EDES, P. G. Michalopoulos, and R. A. Plum, "Improved estimation of traffic flow for real-time control," *Transportation Research Record*, vol. 95, pp. 28–39, 1980.

[5] J. Whittaker, S. Garside, and K. Lindveld, "Tracking and predicting a network traffic process," *International Journal of Forecasting*, vol. 13, no. 1, pp. 51–61, 1997.

[6] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches," *Transportation Research Record*, vol. 1857, no. 1, pp. 74–84, 2003.

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

[8] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2018.

[9] L. Zhang, Q. Liu, W. Yang, N. Wei, and 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.

[10] A. Csikós, Z. J. Viharos, K. B. Kis, T. Tettamanti, and I. Varga, “Traffic speed prediction method for urban networks—an ann approach,” in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 102–108, IEEE, 2015.

[11] T. Man-Chun, F. Luo-Bin, and X. Jian-Min, “Traffic flow prediction based on hybrid arima and ann model,” *China Journal of Highway and Transport*, vol. 20, no. 4, p. 118, 2007.

[12] S. Dunne and B. Ghosh, “Weather adaptive traffic prediction using neurowavelet models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 370–379, 2013.

[13] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[14] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and 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.

[15] D. Yang, S. Li, Z. Peng, P. Wang, J. Wang, and H. Yang, “Mf-cnn: traffic flow prediction using convolutional neural network and multi-features fusion,” *IEICE TRANSACTIONS on Information and Systems*, vol. 102, no. 8, pp. 1526–1536, 2019.

[16] Y. Tian and 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)*, pp. 153–158, IEEE, 2015.

[17] R. Fu, Z. Zhang, and 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)*, pp. 324–328, IEEE, 2016.

[18] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, “Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 5668–5675, 2019.

[19] A. Zonoozi, J.-j. Kim, X.-L. Li, and G. Cong, “Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns.,” in *IJCAI*, pp. 3732–3738, 2018.

[20] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” *arXiv preprint arXiv:2101.11174*, 2021.

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

[22] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, “A3t-gcn: Attention temporal graph convolutional network for traffic forecasting,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, p. 485, 2021.

[23] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.

[24] J. Jin, D. Rong, T. Zhang, Q. Ji, H. Guo, Y. Lv, X. Ma, and F.-Y. Wang, “A gan-based short-term link traffic prediction approach for urban road networks under a parallel learning framework,” *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[25] N. AbuAli and H. Abou-zeid, “Driver behavior modeling: Developments and future directions,” *International journal of vehicular technology*, vol. 2016, 2016.

[26] A. Schwarze, M. Buntins, J. Schicke-Uffmann, U. Goltz, and F. Eggert, “Modelling driving behaviour using hybrid automata,” *IET intelligent transport systems*, vol. 7, no. 2, pp. 251–256, 2013.

[27] Z. Cheng, M.-Y. Chow, D. Jung, and J. Jeon, “A big data based deep learning approach for vehicle speed prediction,” in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 389–394, IEEE, 2017.

[28] B. Jiang and Y. Fei, “Vehicle speed prediction by two-level data driven models in vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1793–1801, 2016.

[29] J. Jing, D. Filev, A. Kurt, E. Özatay, J. Michelini, and Ü. Özgüner, “Vehicle speed prediction using a cooperative method of fuzzy markov model and auto-regressive model,” in *2017 IEEE intelligent vehicles symposium (IV)*, pp. 881–886, IEEE, 2017.

[30] G. Bacchiani, D. Molinari, and M. Patander, “Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning,” *arXiv preprint arXiv:1903.01365*, 2019.

[31] F. van Wageningen-Kessels, H. Van Lint, K. Vuik, and S. Hoogendoorn, “Genealogy of traffic flow models,” *EURO Journal on Transportation and Logistics*, vol. 4, no. 4, pp. 445–473, 2015.

[32] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[33] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *stat*, vol. 1050, p. 20, 2017.

[34] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.

[35] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?,” *arXiv preprint arXiv:1810.00826*, 2018.

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

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

[38] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dive into deep learning,” *arXiv preprint arXiv:2106.11342*, 2021.

[39] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Icml*, 2010.

[40] M. Kyte, Z. Khatib, P. Shannon, and F. Kitchener, “Effect of weather on free-flow speed,” *Transportation Research Record*, vol. 1776, no. 1, pp. 60–68, 2001.

[41] B. Cao and Z. Yang, “Car-following models study progress,” in *2009 Second International Symposium on Knowledge Acquisition and Modeling*, vol. 3, pp. 190–193, IEEE, 2009.

[42] M. W. Hofbaur, *Hybrid estimation of complex systems*, vol. 319. Springer Science & Business Media, 2005.

[43] D. Luxen and C. Vetter, “Real-time routing with openstreetmap data,” in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS ’11, (New York, NY, USA), pp. 513–516, ACM, 2011.

[44] M. Kwiatkowska, G. Norman, and D. Parker, “Prism 4.0: Verification of probabilistic real-time systems,” in *International conference on computer aided verification*, pp. 585–591, Springer, 2011.

[45] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.