Center for Advanced Multimodal Mobility Solutions and Education

Project ID: 2022 Project 02

REAL-TIME FREEWAY SPEED PREDICTION BASED ON DEEP LEARNING IN CONNECTED AND AUTONOMOUS VEHICLES ENVIRONMENT

Final Report

by

Wei Fan (ORCID ID: https://orcid.org/0000-0001-9815-710X)
Chengying Hua (ORCID ID: https://orcid.org/0000-0002-4691-7100)

Wei Fan, Ph.D., P.E.
Director, USDOT CAMMSE University Transportation Center
Professor, Department of Civil and Environmental Engineering
The University of North Carolina at Charlotte
EPIC Building, Room 3261, 9201 University City Blvd, Charlotte, NC 28223
Phone: 1-704-687-1222; Email: wfan7@uncc.edu

for

Center for Advanced Multimodal Mobility Solutions and Education
(CAMMSE @ UNC Charlotte)
The University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223

September 2022

ACKNOWLEDGEMENTS

This project was funded by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE @ UNC Charlotte), one of the Tier I University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT), under the FAST Act. The authors are also very grateful for all of the time and effort spent by DOT and industry professionals to provide project information that was critical for the successful completion of this study.

Table of Contents

List of Tables

List of Figures

EXECUTIVE SUMMARY

The development of connected autonomous vehicles (CAVs) has received a lot of attention in recent years, presumably as a result of the exponential rise in the use of artificial intelligence techniques in a wide range of fields. Utilizing cooperative adaptive cruise control (CACC), CAVs can significantly assist traffic engineers in managing traffic flow and reducing traffic congestion on road networks.

The Intelligent Transportation System (ITS) has been proven capable of effectively addressing traffic congestion issues. Timely traffic prediction is undoubtedly a crucial component for CAVs to operate more effectively, increase mobility, and reduce traffic congestion. A thorough review of the extant literature strongly implies that the focus of CAVs research has switched away from traditional statistics and optimization models toward adaptive machine learning methods. Due to the non-linear and complex correlations inherent in the spatiotemporal data collected from sensors, it is conceivable that existing machine learning models cannot be easily developed and directly applied in this situation.

This project builds a framework for predicting traffic speed based on multiple deep learning models using the Caltrans Performance Measurement System (PeMS) data. Additionally, this project establishes a simulation environment for CAVs, and compares the traditional car-following model with deep learning methods in terms of multiple performance metrics. The results indicate that both supervised learning and unsupervised learning are superior to the simulation-based model on the freeway. And the deep learning networks are almost identical to one another. Besides, it reveals that all models have their latent features for different time dimensions under the low traffic conditions, transition states, and heavy traffic conditions. Traffic engineers and other interested parties may benefit from the project’s findings in traffic operation and management, such as bottleneck identification, platooning control and route planning.

Chapter 1. Introduction

1.1 Problem Statement

The continuous growth in the number of vehicles leads to various mobility issues in the current transportation system, such as increased traffic congestion and high commuting time. Benefiting from the increasing popularity and deployment of artificial intelligence technology, CAVs (i.e, connected and autonomous vehicles) are supposed to alleviate traffic congestion. For CAVs to conduct effectively and improve mobility, real-time prediction of traffic speed is undoubtedly essential to the entire Intelligent Transportation System (ITS).

Accurate speed prediction can help efficiently control traffic in advance and short-term forecasting has gained popularity due to its adaptability (Liu et al., 2021). An overview of existing literature indicates that traffic prediction tasks have shifted from statistical models to adaptive machine learning (ML) methods (Miglani and Kumar, 2019). Furthermore, with enhanced data storage capacities, more historical data may be converted into meaningful information for data-driven models. However, due to the nonlinearity between the high-dimensional spatiotemporal data collected from sensors, shallow ML techniques may be unsatisfactory in the intelligent driving environment, especially as the forecasting horizon increases (Yu et al., 2017). This inspires researchers to address time series traffic prediction using deep learning (DL) methods and explore improved results (Lv et al., 2014; Tian and Pan, 2015; Fu et al., 2016).

Driving behavior on the road is determined by an individual's decision in current traffic circumstances using previous experience. In contrast to human-driven cars, where driving behavior is usually uncertain and can only be estimated via massive data from roadside units (RSUs), the control algorithms of CAVs may be predictable (Gora et al., 2020). However, the requirement for specialized infrastructure and robust algorithms during execution make traffic prediction costly in a real ITS environment. The Intelligent Driver Model (IDM) is a widely used car-following model to forecast the vehicle status in an intelligent collision-free manner and modify its behavior as desired (Helbing et al., 2009). Besides, the vehicle-road synergy is still in its initial phase, with fewer scenario-based, large-scale tests, and comprehensive frameworks in place (Do et al., 2019).

Hence, it is necessary to develop a simulated environment to control CAVs at various levels.

This project focuses on the traffic speed prediction task based on emerging deep neural networks (DNNs) using ground truth data. It also establishes a simulation environment for CAVs and compares the traditional IDM model with deep learning methods for their prediction accuracy in terms of multiple evaluation metrics. The findings can greatly help researchers and traffic engineers better improve dynamic traffic management. Platooning control, route planning and signal optimization are some of the potential applications for traffic speed prediction results.

1.2 Objectives

The main objective of this research project is to enhance the accuracy of traffic speed prediction in ITS. The objectives of this project include:

(1) Conducting a comprehensive review of traffic prediction techniques for CAVs.

(2) Identifying a potential freeway segment and collecting the features of the selected scenario.

(3) Developing and applying a logically intelligent car-following model that can describe CAVs on the highway mainline.

(4) Predicting the average traffic speed in the CAVs environment on the freeway, and comparing the performance of emerging deep learning technology with the existing car-following models.

1.3Expected Contributions

In this project, the expected contributions are summarized as follows:

(1) Reviewing the existing public datasets for traffic prediction. Also, traffic simulation models for CAVs and state-of-art traffic prediction techniques are introduced.

(2) Aiming at the spatiotemporal problems of deep learning technology, multiple Solution methods are proposed to realize real-time prediction.

(3) A simulation environment is established for CAVs to predict freeway speed and

compare the traditional car-following model with deep learning methods for their prediction accuracy in terms of multiple evaluation metrics.

(4) Investigating the latent features of different models during various time dimensions under the low traffic loads, transition states, and heavy traffic loads.

1.4 Report Overview

In this chapter, the background and motivation of the project have been discussed, followed by the research objectives and expected contributions.

Chapter 2 presents the systematic literature review. A keyword-based (connected and autonomous vehicles, traffic prediction, machine learning) search is conducted to identify existing highly relevant studies from the search results. A series of previous studies using traditional and deep learning methods for traffic prediction are investigated. Public datasets and microscopic simulation-based studies for CAVs are also presented.

Chapter 3 describes traffic prediction techniques used in this project. Given that traffic forecasting is a spatial and temporal process, deep learning (DL) techniques has been demonstrated to be an effective alternative to conventional methods. In order to reflect the car-following behavior of the CAVs, a model that can correctly characterize the driving behavior of the CAVs is developed, i.e., the Intelligent Driver Model (IDM), in which each parameter represents an actual meaning.

Chapter 4 identifies potential freeway segment and collects necessary traffic data related to the selected segment on the freeway. The California Department of Transportation (Caltrans) Performance Measurement System (PeMS) database is used as the data source for the potential freeway segment. In order to improve performance results, multiple evaluation metrics are used.

Chapter 5 shows the detailed outcomes of the suggested models. Two deep learning models, supervised deep neural network represented by Gated Recurrent Units (GRU) and unsupervised deep neural network represented by Stacked Autoencoders (SAEs), and the Intelligent Driver Model's prediction errors are discussed for various time dimensions under the low traffic loads, transition states, and heavy traffic loads.

Chapter 6 describes the conclusion of the prediction performance of different models. Additionally, instructions for future work are also given.

Chapter 2. Literature Review

2.1 Introduction

The current state-of-the-art and state-of-the-practice databases, microscopic traffic simulation models, and different approaches to predict the traffic speed under the connected and autonomous vehicles (CAVs) environment are all thoroughly reviewed in this chapter. This should give a comprehensive picture of the current CAV-based traffic speed prediction methods, potential freeway scenarios, and simulation environments. This is intended to provide a solid reference in formulating traffic data analysis and developing effective strategies for future tasks.

The most popular databases that can be used for research in CAVs, including PeMS and NGSIM, are introduced in Section 2.2. The models for microscopic traffic simulations that take connected and autonomous vehicles into account are described in Section 2.3. SUMO receives special consideration because it can measure the operational status of CAVs. Several prior studies that used simulation methods for modeling the CAVs, such as IDM, ACC, and CACC, are investigated and presented as well to gain a better understanding of the simulation methods. Section 2.4 presents conventional techniques and definitions of deep learning technologies, followed by the currently employed methodologies, allowing readers to compare various prediction techniques.

2.2 Traffic Databases for CAVs

The field of transportation spans numerous academic disciplines. Only a tiny number of professional data sets have been made available to the general public due to issues including confidential data and uneven recognition in the industry. The two popular databases used in the field of intelligent transportation systems will be described in the paragraphs that follow.

2.2.1 Caltrans Performance Measurement System (PeMS)

2.2.1.1. The Introduction of PeMS

PeMS that gives customers access to both real-time and historical traffic data, was initially established in 1999 as a university research project. All of California's main urban cities' motorway systems have roughly 40,000 separate detectors that collect data every 30 seconds. The majority of PeMS's detecting tools are inductive loops. The inductive loops are placed at particular points on the motorway, and data is recorded by a controller in a cabinet on the side of the road. Other data sets can potentially provide information to the PeMS database. The Caltrans Districts offer the information on detector setup. Information on highway layout, such as the number of lanes, and incident data are provided by Caltrans Headquarters (i.e., number of collisions and type of collisions).

From the flow and occupancy information, the average vehicle speeds are determined using the g-factor (effective vehicle length). To produce precise speed estimations, PeMS computes the g-factor for each loop using an adaptive method. The technique has been examined and verified using "ground truth" data from floating automobiles and double loop detectors (Jia et al., 2001). Users must create an online account on the PeMS webpage in order to use PeMS. After that, users may access the PeMS database for free using a regular internet browser.

2.2.1.2. The Applications of PeMS

PeMS uses sensor data to calculate traffic amounts and other performance metrics. They include information on speed, vehicle-hours of delay, vehicle miles traveled, and trip time. For research projects and other goals related to transportation planning, the PeMS data can be used as an input into simulation models. Users can also utilize PeMS data to calibrate models, resulting in clear findings that take into account the actual traffic situation. Users may conduct both basic and complex traffic studies using PeMS, including highway operating assessments, bottleneck identification, calculating the Level of Service, evaluating advanced control measures, and assessing incident impacts. Managers can at any moment obtain a consistent, in-depth evaluation of the functioning of the highway. Having information about the status of the motorway network at the moment can help traffic engineers make informed operating choices. Planners can decide if congestion bottlenecks can be eliminated by making small capital upgrades or by enhancing operations. The most recent shortest route and estimated journey times are available to travelers. Researchers can calibrate simulation models and validate their theories.

Several versions of sub-data sets (PeMSD3/4/7(M)/7/8/-SF/-BAY) have appeared and have been widely used. The main difference lies in the scope of time and space, and the number of sensors.

- PeMSD3: This data set is processed by Song et al. It includes 358 sensors and flow information from September 1, 2018 to November 30, 2018.
- PeMSD4: It describes the San Francisco Bay Area and contains 3848 sensors on 29 roads from January 1, 2018 to February 28, 2018, for a total of 59 days.
- PeMSD7(M): Describes the 7 districts of California, with a total of 228 stations, and the time range is the working days of May and June 2012.
- PeMSD7: This version is publicly released by Song et al. It contains traffic flow information on 883 sensor stations, covering the period from July 1, 2016 to August 31, 2016.
- PeMSD8: It depicts the San Bernardino area and contains 1979 sensors on 8 roads from July 1, 2016 to August 31, 2016, for a total of 62 days.
- PeMSD SF: This data set describes the occupancy rate of different lanes of the San Francisco Bay Area highway, between 0 and 1. The time span of these measurements is from January 1, 2008 to March 30, 2009, with samples being taken every 10 minutes.
- PeMSD-BAY: It contains 6 months of traffic speed statistics, from January 1, 2017 to June 30, 2017, including 325 sensors in the Bay Area.

2.2.1.3 Other Datasets on the Freeway

- METR-LA: It recorded traffic speed statistics for the four months from March 1, 2012 to June 30, 2012, including 207 sensors on highways in Los Angeles County.
- LOOP: It was collected from loop detectors on four connected highways (I-5, I-405, I-90, and SR520) in the greater Seattle area. It contains the traffic status data of 323 sensor stations in the whole year of 2015 at 5-minute intervals.
- Los-loop: This data set was collected in real time by a loop detector on a highway in Los Angeles County. Including 207 sensors, its traffic speed collection time is from March 1, 2012 to March 7, 2012. These traffic speed data are summarized every 5 minutes.

Table 1 compares the forecast accuracies on various tasks including travel time, flow, speed, occupancy, and traffic demand. It is clear that, compared to other tasks, which have accuracy rates that are near to 80%, the speed prediction may achieve an overall accuracy rate of more than 90%.

Table 1. Statistics Prediction for Different Tasks

Task	Dataset	Time interval (min)	Prediction window (min)	MAPE (%)	RMSE
Flow	PeMSD3	5	60	16.78 (Song et al., 2020)	29.21
	PeMSD4	5	60	11.09 (Shi et al., 2020)	31
	PeMS07	5	60	10.21 (Song et al., 2020)	38.58
	PeMSD8	5	60	8.31 (Shi et al., 2020)	24.74
Speed	METR-LA	5	5/15/30/60	4.90 (Chen et al., 2019)/6.80/8.30/10.00 (Chen et al., 2020)	3.57/5.12/6.17/7.30
	PeMS-BAY	5	15/30/60	2.73 (Wu et al., 2019)/3.63 (Zheng et al., 2020)/4.31 (Zheng et al., 2020)	2.74/3.70/4.32
	PeMSD7 (M)	5	15/30/45	5.24/7.33/8.69 (Yu et al., 2018)	4.04/5.70/6.77
	Los-loop	5	15/30/45/60	-	5.12/6.05/6.70/7.26 (Zhao et al., 2019)
	LOOP	5	15/30/45/60	6.01 (Cui et al., 2019)	4.63
Occupancy	PeMSD-SF	60	7 rolling time windows (24 time-points at a time)	16.80 (Sen et al., 2019)	-

2.2.2. Next Generation Simulation (NGSIM)

2.2.2.1 The Overview of NGSIM

Even though a lot of the microsimulation models in use today are reliable and they offer a variety of analysis choices, there are still significant gaps and restrictions that can impact how accurate the results of these models are. The Next Generation Simulation (NGSIM) initiative was introduced by FHWA's Traffic Analysis Tools Program to help increase the usage of microsimulation systems and guarantee the tools produce correct findings (U.S. Federal Highway Administration, 2006).

The NGSIM datasets were initially gathered with the use of cameras and were then taken out of the resultant recordings. The NGSIM trajectory samples every 0.1 seconds, and each sample contains data such as the vehicle's length, type, longitudinal and lateral locations, and instantaneous speed and acceleration. The following list includes descriptions of the four datasets.

(1) The US-101 trajectory dataset was collected on a segment in the vicinity of Lankershim Avenue on southbound US-101 freeway in Los Angeles, California. The segment is approximately 640 m in length and contains 6 lanes.

(2) The I-80 trajectory dataset was collected on a segment of I-80 freeway in Emeryville (San Francisco), California. The segment is approximately 500 m in length and contains 6 lanes, where the median lane is a high occupancy vehicle (HOV) lane.

(3) The Peachtree trajectory dataset was collected on a segment of Peachtree Street in Atlanta, Georgia. The arterial segment is approximately 640 m in length, with five intersections (four are signalized and one is not) and two or three through lanes in each direction.

(4) The Lankershim trajectory dataset was collected on a segment of Lankershim Boulevard in the Universal City neighborhood of Los Angeles, California. The segment is approximately 488 m in length and contains three or four lanes and four signalized intersections.

Through the NGSIM program, FHWA will create a number of driver behavioral algorithms that form the core logic of microscopic traffic simulation models, together with supporting documentation and validation datasets. These algorithms will explain how multimodal passengers interact with automobiles, highway infrastructure, and other environmental factors including congestion, delineation, and traffic lights. Transportation practitioners will be able to use microsimulation software with more assurance knowing that traffic simulation experts developed the algorithms using high-quality, real-world datasets as a result of the core simulation algorithms developed through NGSIM ultimately being incorporated into commercial simulation models.

2.2.2.2 The Applications of NGSIM

The NGSIM datasets generally offer two key benefits for transportation research. The first is its high resolution, which enables studies of extremely specific driving behaviors and calibration or estimation of minute behavioral characteristics and variables. The second benefit is its accuracy in capturing traffic patterns, which gives researchers a comprehensive picture of actual traffic throughout the collecting time and at the collection sites. Making use of the benefits, these research studies primarily employed the NGSIM trajectory data in the methods listed below:

- Calibrating or training traffic flow models.
- Validating models.
- Demonstrating driving behaviors or traffic phenomena.
- Conducting analysis of samples.
- Building simulation environments or testbeds.

Coifman (2015) proposed a technique to estimate the fundamental diagrams without needing to look for stationary circumstances. The clue, it was discovered, was vehicle length. The suggested approach was used to examine the higher-resolution I-80 dataset as a supplement to the loop detector data, enhancing the method's veracity.

Siqueira et al. (2016) introduced a stochastic transport model with discrete speed spectrum and offered an alternative stochastic model for the fundamental diagrams. The model parameters and empirical fundamental diagrams were estimated using the I-80 dataset, and the estimated results were compared to those produced by the proposed model.

In order to study the effects of driver variability on macroscopic traffic flow relations, Jabari et al. (2014) suggested a stochastic variant of the macroscopic traffic flow speed-density relation. The first 15 minutes of I-80 data (4:00 to 4:15 p.m.) were used to estimate the distributions of the model's parameters, and the next 30 minutes (5:00 to 5:30 p.m.) were used to draw an empirical speed distribution that was then contrasted with the outcomes of the simulation.

By taking into account vehicle acceleration in crowded traffic, Wu and Coifman (2014) suggested a length-based vehicle classification approach from dual-loop detectors. The I-80 dataset was utilized to test the effectiveness of the suggested vehicle categorization approach by establishing virtual loop detectors. The high-fidelity

NGSIM datasets' vehicle length information allows for the evaluation of the vehicle length-based study.

Past research has sought a better understanding of how to utilize the NGSIM in the macroscopic traffic variables estimation. Based on the literature review as presented above, Table 1 exhibits a summary of the existing studies using NGSIM data.

Table 2. Summary of Macroscopic Traffic Variables Estimation Using NGSIM

No.	Author, Year	Dataset	Main Usage of Data
1	Coifman, 2015	I-80	Being analyzed by using the proposed method as a complement of loop detector data
2	Siqueira et al., 2016	I-80	Calibrating model parameters and estimating the referred fundamental diagram
3	Jabari et al., 2014	I-80	Calibrating the distributions of model parameters using the (first 15 min) I-80 data, and validating model by comparing with speed distributions (other 30 min)
4	Wu and Coifman, 2014	I-80	Evaluating performance of the proposed method after setting virtual loop detectors

2.3 Traffic Simulation Models for CAVs

This section provides a wider analysis of studies focusing on the development or application of simulation models of traffic with CAVs, including the variable input and control methods that are relevant in describing the traffic dynamics and mutual interactions of CAVs.

2.3.1 Classification of Traffic Simulation Models

Simulation models can be distinguished from the macroscopic level (where traffic is represented in terms of relations between aggregated values such as speed, flow, and density), through the mesoscopic level (where cars travel in homogeneous packets and only alter their behavior in response to special events, like turning or stopping before a red signal), to the microscopic level. The wide variety of models that are accessible is warranted since traffic models are just approximate representations of extremely complicated and unpredictable real-world traffic systems, and different models may be appropriate for different demands and applications.

There are currently no actual field data for CAVs, several academics create micro-simulation or macro-simulation models to assess the implications of CAVs. This is mainly due to the lack of a car-following model that could accurately capture the car-following features of CAVs. Additionally, each simulation research used a different methodology and examined a specific performance indicator (e.g., micro stability, throughput, acceleration, headway profiles, macro link traffic volume, and link travel time). In this review, we primarily concentrate on the microsimulation-based research studies that take longitudinal dynamics into account.

2.3.2 Simulation-Based Car-Following Models for CAVs

Although some microscopic models are used more frequently than others, there are no standardized techniques, making it challenging to contrast various models. The absence of empirical CAVs data that might be utilized to precisely calibrate and validate models is one of the key factors. Additionally, CAV algorithms are still being developed. It could turn out that CAV control algorithms in actual traffic can be improved by using microscopic simulation models of CAVs.

The Microscopic Model for Intelligent Cruise Control Simulation (MIXIC) was created by Van Arem and De Vos (1997). The MIXIC is one of the models that have been most frequently used for cooperative intelligent vehicle simulations since it was an early intelligent vehicle model. Its broad use stems from the following factors:

- By transferring speed, acceleration, and/or braking capabilities between the antecedent and subsequent vehicles, the MIXIC model contains V2V communication. Better CACC characteristic simulations are made possible by such model capacity.
- The model is calibrated for various two, three, and four lane scenarios, producing a well-adjusted traffic flow model that is consistent with actual circumstances. Additionally, if a full calibration of a vehicle's performance was unavailable, the MIXIC results were proven to be trustworthy.

Treiber and Hennecke (2000) were the creators of the original intelligent Driver Model (IDM). The IDM is the most popular model for CAV simulations because it produces a realistic acceleration profile in a single lane environment and is one of the easiest and accident-free models available. The IDM's features are more similar to those of ACC cars than they are to those of a human-driven vehicle since it lacks an explicit response time and is provided as a continuously differentiable acceleration function. The IDM itself may be used as an ACC or a model for a human-driven vehicle by altering a few settings.

2.3.3 Microscopic Simulation-Based Studies for CAVs

Numerous studies considered a range of potential market-penetration rates for CAVs. A tiny number of studies exclusively modeled the extreme 100% penetration rate of CAVs without taking progressive growths into account. The research studies that examined this aspect with an emphasis on the simulation-based CAV modeling investigations are shown in Table 3.

A novel family of simulation models was created by VanderWerf et al. (2001), and they include the crucial aspects of driver behavior and control system architecture that have an impact on the dynamics and capacity of traffic flow. The impacts of new driver aid technologies, including adaptive cruise control (ACC), on traffic flow dynamics and capacity are predicted using a set of mathematical models. For the purpose of demonstrating that the models are delivering accurate results, example outputs from simulation validation test cases are presented and discussed.

An adaptive cruise control (ACC) technique was introduced by Kesting et al. (2007), in which the driving style and acceleration characteristics automatically adjust to changing traffic conditions. The concept consists of three parts: the actual ACC, which is implemented as a car-following model; an algorithm for automatically determining the traffic situation in real-time using local data; and a driving strategy matrix to modify the driver’s characteristics, or the ACC controller’s parameters, to the traffic conditions.

To test system performance with various AV ratios, Zhou et al. (2017) created a cooperative intelligent driver model. The findings demonstrated that an increasing proportion of AVs would cut down on overall travel time and smooth out traffic oscillations with the use of an appropriate vehicle-to-vehicle regulating mechanism.

Talebpour and Mahmassani (2016) provided a framework for simulating various vehicle kinds with varying communication capabilities using various models with technology-appropriate assumptions. This framework’s stability study of the ensuing traffic stream behavior for various connected and autonomous car market penetration rates is shown. According to the investigation, string stability may be raised by linked and autonomous cars.

A microscopic simulation framework was developed by Rios-Torres and Malikopoulos (2017) to evaluate the effects of CAVs on traffic flow at merging highways and the consequences for fuel consumption and journey time. The simulation findings demonstrated that CAVs may significantly reduce fuel consumption and trip time for a variety of traffic circumstances under scenarios of average and severe congestion.

In order to include novel algorithms that are critical to describing the interactions between CACC cars and manually driven vehicles in mixed traffic, Liu et al. (2018) expanded a state-of-the-art CACC modeling framework. To create the high volume traffic flow that is anticipated to emerge as a result of the CACC string operation, the upgraded modeling framework implements a new vehicle dispatching model. To provide realistic CACC vehicle behaviors in highway on/off-ramp zones where traffic disruptions could regularly interfere with the CACC string operations, the framework additionally includes new lane change rules and automatic speed control algorithms.

Table 3. Simulation-Based CAV Modeling Studies

No.	Author, Year	Models	Objectives
1	VanderWerf et al., 2001	An error-based ACC and CACC. The lane change is human control.	Develop the ACC and CACC car-following models and estimate their impact
2	Kesting et al., 2007	IDM	Propose the ACC-based traffic-assistance system intended to improve traffic flow and road capacity
3	Zhou et al., 2017	The Full Velocity Difference Model (FVDM) and IDM	Develop a cooperative IDM (CIDM) to examine the system performance under different proportions of the AVs.
4	Talebpour and Mahmassani, 2016	MIXIC for AV. IDM for CAV.	Propose an acceleration framework to address the limitations of micro-simulation models in capturing the changes in driver behavior in a mixed environment
5	Rios-Torres and Malikopoulos, 2017	Optimal control for CAVs. Gipps model for manual vehicles	Develop a micro-simulation framework for CAVs to analyze the impact on fuel consumption and travel time.
6	Liu et al., 2018	CACC and anticipatory lane change (ALC)	Extend the CACC modeling framework to incorporate new algorithms describing the interactions between the CACC and manual vehicles in mixed traffic.

2.4 Traffic Prediction Methods

The CAV system’s key component for addressing issues with traffic congestion is traffic prediction. To avoid a breakdown in traffic flow, the forecasted traffic information may be distributed to traffic control towers, RSUs, drivers, and CAVs. Furthermore, an ideal routing strategy and traffic management may be carried out using traffic estimations. Other important applications for traffic prediction include vehicle route planning, traffic signal optimization, and real-time congestion control.

Traffic prediction is affected by various factors such as – forecasting horizon, sampling frequency, algorithms, type of dataset, type of area, data source etc., which is shown in Table 4.

The span of time in the future during which traffic prediction is conducted is known as the forecasting horizon. According to Ishak and Al-Deek, the models' accuracy decreases as predicting horizon increases (2002). Additionally, the prediction becomes more difficult as the predicting horizon grows shorter. According to the Highway Capacity Manual, the optimal forecast interval is a 15-minute interval (Smith and Demetsky, 1997). The impact of forecast time horizon on the precision of short-term traffic prediction was studied by Larry (1995). However, the majority of methods in the literature were created for short-term forecasting.

Sampling frequency and aggregation rate play a key role in data resolution. The inaccuracy decreases as the amount of aggregation increases. However, compared to historical values, more recent traffic data observations could serve as superior forecasters (Polson and Sokolov, 2017).

The two categories of algorithms are univariate and multivariate. While a multivariate technique uses numerous sites for input and output, a univariate approach just monitors traffic characteristics from a single site. Multivariate models, as demonstrated by Kamarianakis and Prastacos in 2003, better capture observations made at many times and places than univariate models.

The type of area that describes the area of data collection for carrying out experiments of traffic flow prediction is another important consideration. Examples include freeways, urban arterials, highways, etc. In this situation, sensor-based traffic data from an arterial route has data values that are significantly longer than data from a local road.

Highway traffic flow has a cyclic and dynamic character, according to Li and Liu's (2014) research. Additionally, compared to the highway, it is more challenging to estimate traffic flow in urban areas due to signalization limitations (Vlahogianni et al., 2000; Zhang and Huang, 2018).

Whether a real or simulated dataset is utilized for experimentation depends on the type of dataset. The source of the data used to calculate traffic flow numbers is represented by the data source, for example, loop detectors, sensors, GPS, crowdsourcing, social media, and floating car data.

Table 4. Factors Affecting the Traffic Prediction

Factors	**Descriptions**
Forecasting horizon	Range of time ahead to which traffic prediction is carried out.

Sampling frequency	General about 5 minutes
Algorithms	Univariate and multivariate
Type of area	The area of data collection for implementing experiments
Type of dataset	Real or simulated
Data source	Loop detectors, sensors, GPS, crowdsourcing, social media, floating car

2.4.1 Traditional Methods

One of the first techniques for anticipating traffic demand is based on the analysis and forecasting of time series of observed historical data. Techniques used in time-series models include non-linear regression, smoothing, averaging algorithms, seasonal ARIMA (SARIMA) models, and others. The most popular time series approach for predicting traffic is the autoregressive integrated moving average (ARIMA) model, which assumes that traffic conditions are a static process with unaltered mean, variance, and auto-correlation.

The autoregressive-moving average (ARMA) model has been expanded to create the ARIMA model. To forecast next series points, this model is specifically used to analyze the time-series data. The acronym for the ARIMA model is ARIMA (p, d, q), where p, d, and q stand for the moving average, integrated, and autoregressive polynomial orders, respectively. In order to forecast short-term highway traffic flow, Mohammed et al. introduced the ARIMA model in 1979. After ARIMA, numerous variations of ARIMA were proposed to increase prediction accuracy for traffic flow prediction, including SARIMA (Williams and Hoel, 2003), Kohenen ARIMA (KARIMA) (Van Der Voort et al., 1996), ARIMA with explanatory variable (ARIMAX) (Williams, 2001), and Vector ARIMA (Gallego, 2009).

The historical average (HA) approach, which uses an average of historical traffic data to forecast future traffic flow, is another way to deal with traffic flow data. This approach is based on how traffic flow is cyclical. However, this approach does a poor job of adjusting to unanticipated situations like accidents. For instance, Stephanedes et al. (1981) enhanced traffic flow prediction and real-time control using the HA approach with fewer calculations and less data.

The Kalman filtering (KF) method is another widely used parametric technique for time-series models and is typically used in nonstationary stochastic environments. The benefit of KF is that it allows for seamless updating of certain state variables. However, erratic traffic patterns could cause a shift in how traffic is moving. The estimation of an accurate traffic flow prediction is made harder by these unsteady flow characteristics and also by environmental elements that are not explicitly stated. Additionally, the KF technique produces predictions that are either overpredicted or underpredicted.

For handling data on traffic flow, Bayesian networks are helpful since they give matching variances in addition to the mean value of prediction. Additionally, when fresh data becomes available, this model may update the prediction findings. The probability distribution between the input and output of traffic flow data is likewise seen by the Bayesian network. Sun et al. (2006) used geographical historical traffic data from a nearby road link to create a Bayesian model to forecast traffic flow for a segment.

For forecasting traffic flow, researchers frequently employed the non-parametric methodologies k-NN and SVR. They both fall under the category of shallow ML approaches. K-NN was used by Davis and Nihan to forecast short-term highway traffic (Davis and Nihan, 1991). The suggested k-NN model, according to the authors, performs similarly to the linear time series technique but not better.

Support Vector Regression (SVR) is a supervised machine learning technique that is mostly used for classification and regression. It is trained to learn a function to transfer input features to output. The goal of SVR is to map input data to a high-dimensional feature space, and then use that same space to conduct linear regression. Here, the dataset is initially displayed with each item as a point in n-dimensional feature space. The next step in classification is to find the hyperplane that categorizes the input. Compared to NN, SVR uses the Structural Risk Minimization (SRM). Additionally, it ensures localization of global minima. However, despite the fact that both of these approaches are nearly identical, they differ in the kind of value they provide as a result (SVR outputs a real number whereas SVM outputs either 0 or 1). The accuracy of traffic flow prediction is examined using a supervised online support vector regression technique by Neto et al. (2009) under both typical and exceptional traffic scenarios.

A comparison of the above discussed models has been presented in Table 5.

Table 5. Traditional Models for Traffic Prediction

Model	**Strengths**	**Weakness**
ARIMA (Box-Jenkins model)	• Linear patterns	• Focus on means and miss the extremes
		• Bad on rapid fluctuations
		• Software dependent
		• Requires sufficient data
HA	• Easy to implement	• Bad on unexpected events
	• Fast execution	
	• Longer horizon	• Computationally expensive
KF	• Updated continuously	• Assumes dependent and independent variable
	• Multivariate environment	• Gaussian hypothesis
	• Need limited data	

Bayesian Network	• Density function	• Can’t handle high dimensional data
	• Update new information	
k-NN	• Noisy data	• Can’t handle spatial and temporal modeling simultaneously
	• Large data	
	• Fast execution	
SVM	• Don’t assume any underlying relationship about data form	• Not good for linear patterns
	• Even unstructured data	• Time consuming
	• High dimensional data	

2.4.2 Machine Learning Terminology

Supervised Learning and Unsupervised Learning are the two primary categories into which machine learning techniques fall. Input data for supervised learning algorithms must include labels that are specific about the naming of the data. It intends to carry out the two main objectives of classification and regression. Unsupervised learning, in contrast, identifies patterns and distributions within the provided data sets. It also entails the challenges of density estimation and grouping.

Self-learning traffic prediction algorithms fall generally into two categories: parametric and non-parametric. Researchers, however, chose non-parametric approaches over parametric methods due to the stochastic, indeterministic, and non-linear nature of traffic flow data. The taxonomy of the traffic flow prediction model is displayed in Figure 1.

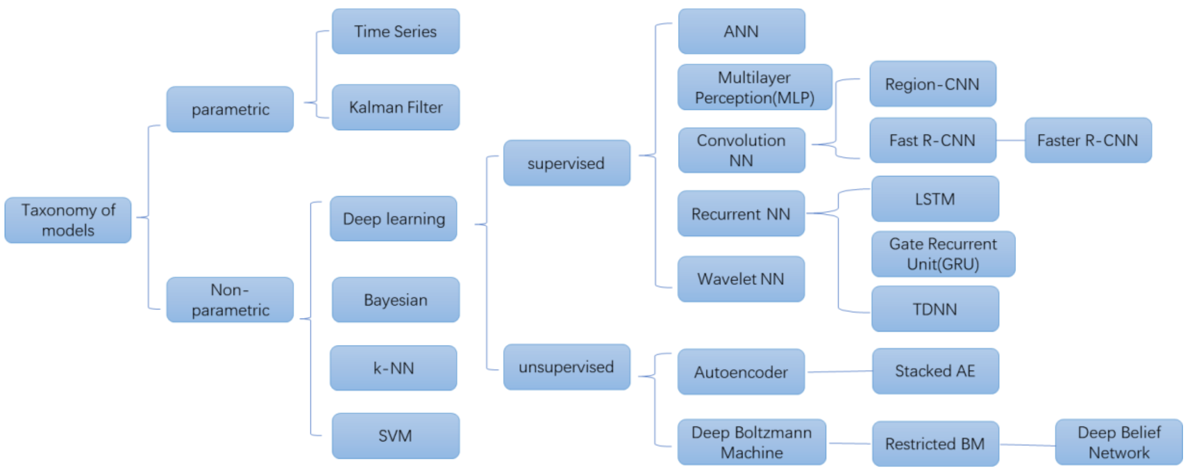

Figure 1. Taxonomy of the Traffic Prediction Models

A subset of machine learning called deep learning (DL) tries to build a computational model with numerous processing layers to accommodate high-level data abstraction. Without human intervention, DL can automatically extract features from data to discover latent relationships between various data set properties (Shickel et al., 2018). DL models have shown reliable results when compared to conventional ML techniques. The human nervous system serves as an inspiration for DL concepts. In light of this, the bulk of DL architectures are created utilizing the ANN framework. A single neuron termed a perceptron serves as the structural foundation of NN. It accepts several inputs, analyzes them using a weighted summation of inputs, and then passes the processed data to an activation function to produce the output.

Data on traffic flow are measured using both geographical and temporal methods. While spatial correlation describes the correlation between traffic flow of the target road segment and simultaneous sample values of its nearby and distant areas for the same time interval, temporal correlation defines the correlation between current and historical traffic flow samples collected with a temporal span. For instance, incidents on a road that occurred two hours ago might result in gridlock on nearby connecting roads for the next three to four hours.

2.4.3 Deep Learning Methods

Most traditional ML-based traffic prediction techniques cannot uncover deep correlation in traffic data. Non-parametric approaches are increasingly frequently used for prediction since traffic flow has complicated and non-linear patterns.

The flowchart for building DL architecture (Miglan and Kumar, 2019) for traffic flow prediction is shown in Figure 2. Real-time raw data that was recorded is first transformed into a standard format. The input data is then split into a training set and a testing set. The DL model is then trained, and the parameters are changed until the evaluation parameters (which measure the cost and performance requirements) are below a predefined threshold. The evaluation of the forecast outcomes and testing of the testing data comes last.

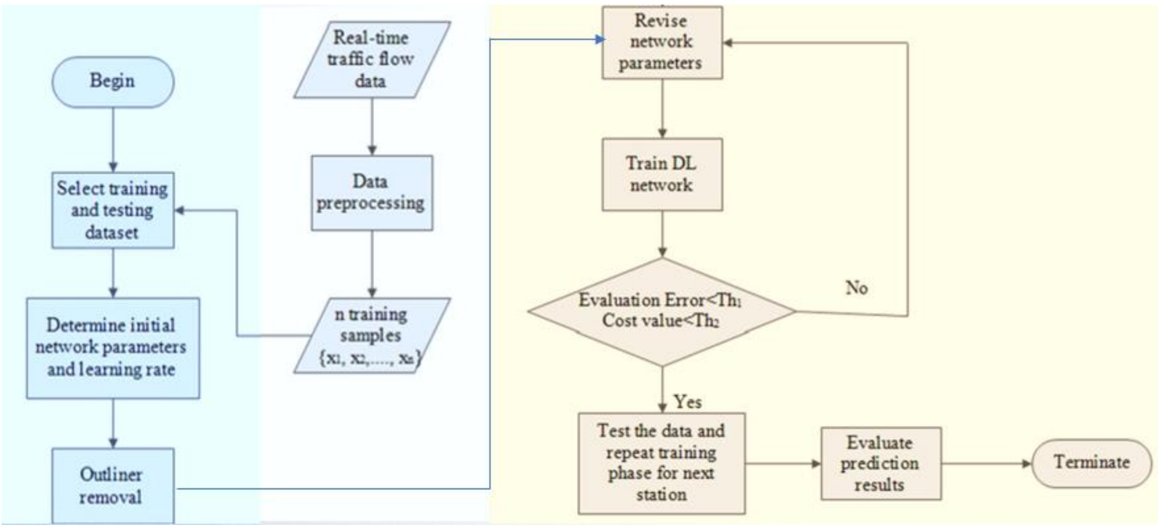

Figure 2. Flowchart of Implementing DL Models

2.4.3.1 Supervised DL Techniques for Traffic Prediction

Hua and Faghri (1994) proposed the concept of traffic information prediction using NNs, where ANNs were utilized to estimate vehicle journey time. Since then, many NN models for forecasting traffic data have emerged. Smith and Demetsky created a NN model in the early 1990s, for which they compared to the conventional approach of traffic prediction. The findings showed that NNs outperform conventional ML models at peak periods (Smith et al., 1994).

A cooperation-based ANN model for predicting urban traffic flow was suggested in the study of Ledoux (1997). The model forecasts traffic flow for the following 60 seconds using past traffic flow data that has been simulated. First, the traffic patterns on a signalized connection were modelled using a single ANN. The data was then shared across linked local NN to simulate traffic flow at a junction.

By combining the prediction output from an online KF and NN with a fuzzy rule-based system (FRBS), Stathapoulos et al. (2008) created a hybrid NN model. According to the findings, hybrid prediction performs better when urban traffic flow becomes more non-linear, unpredictable, and highly variable.

Different situations from incident and typical regions are represented by the work zone area. For such planned work zone locations, Hou et al. (2015) offered 4 alternative models for both short-term and long-term traffic prediction. These four models are non-parametric regression, regression tree, MLFFNN, and RF. However, these ideas marked the first-time traffic prediction that had been done utilizing RF and regression tree approaches. For both long-term and short-term forecasts, RF method yields the highest degree of precision.

Parmula (2018) sought to investigate the use of NN in traffic flow prediction in the event of input data loss. They computed the difference between the sensitivity of data loss for the MLP and AE models in their investigation. The results of the experiment showed that MLP provides superior accuracy than AE in case of data loss.

Table 6 provides a comparison of the above discussed research studies. Notably, all these studies are based on short-term prediction horizon.

The field of vision-based traffic flow prediction makes use of CNNs. Historical data is shown as a picture in issues of traffic flow prediction based on CNN. Additionally, CNNs are able to represent topological locality, i.e., they can identify patterns between inputs that are close to one another.

In a recent work, Chung et al. (2018) used video footage and a deep CNN technique to count the number of automobiles on a specific road stretch. CNNs may extract spatial correlation of traffic flow by employing a multi-layer convolutional structure.

Liao et al. (2018) developed a model that combines ensemble learning and random subspace learning on deep CNN to address the issue of incomplete data. They showed that CNN, SAE, and DBN outperform NN and SVR in comparison to their suggested model and SAE, DBN, NN, and SVR.

Conventional CNNs can only be modeled for grid-based data (image and voice, etc.). Additionally, each network's traffic flow data may always be translated to a certain graph structure. Graph Convolution NN (GCNN), which extends the convolution operator from regular to irregular data, is devised in this context (Duvenaud et al., 2015; Defferrard et al., 2016). Similar to this, Yu et al. (2017) used simply CNN structure to simulate the spatial-temporal properties of a traffic dataset that was structured as a graph. The proposed graph convolution network outperforms the RNN-based model during the training phase.

Table 6. ANN and MLP- Based Traffic Prediction Studies

No	Author	Year	DL	SF (min)	Source	Area	Comparison
1	Smith et al.	1994	ANN	15	Sensors: Capital Beltway, Virginia	Freeway	HA, ARIMA
2	Ledoux	1997	MLP	-	Simulation Semi Macroscopique traffic	Urban	-

3	Stathapoulos et al.	2008	ANN + KF	3	LD: Alexandras Avenue in Athens, Greece	Urban arterial	ANN, KF
4	Hou et al.	2015	MLFFNN	60	Sensors: I-270, MO-141 in St.Louis, MO, USA	Work Zone, signalized arterial	–
5	Parmula	2018	MLP + AE	5	LD, VD: Gliwice Traffic Control Centre	Urban	–

LD: Loop Detectors, VD: Video Detectors

RNNs provide a feedback loop that runs from the next input to the interim output and is only suitable for temporal modeling. The issue of vanishing gradient and inflating gradient causes conventional RNN to fail when attempting to anticipate traffic over a lengthy period of time. Variants of RNN, including as LSTM, GRU, and TDNN, are frequently employed in forecasting short-term traffic flow in the network to address these challenges. Ma et al. (2015) employed LSTM NNs for the first time in the field of transportation.

Qiao et al. (2017) suggested another LSTM-based approach with the aim of obtaining periodic features as well as geographical and temporal characteristics of traffic flow. Repeated conduct by an individual is one of a person’s periodic traits.

When combined with meteorological data from Beijing, the LSTM model’s prediction accuracy was tested by Zou et al. (2018) utilizing GPS tracked data from cabs. It has been demonstrated that when training time is increased, the RMSE of the model drops. Additionally, the introduction of fine-grained and high-resolution data might make LSTM model training more difficult due to the rise in model parameters.

Abbas et al. (2018) presented a method to address this problem in which the road network was first divided into smaller segments before an LSTM model was used to train the data gathered inside those segments. It has been demonstrated that 2-layer stacked LSTM enhances model accuracy when compared to conventional design.

Fu et al. (2016) applied GRU for the first time in the field of traffic flow prediction. Results showed that GRU NNs outperforms LSTM NNs in terms of performance.

Zhang et al. (2018) carried out a research that takes the weather into account while analyzing traffic statistics for a certain time period. Therefore, a model based on GRU and DNN was utilized to enhance the predictions of traffic flow. However, the dataset that was utilized to draw conclusions was rather limited.

The deterioration problem suggests that not every model is equally simple to optimize. ResNet is discovered to provide a solution to the degradation issue in this scenario. In order to estimate traffic flow, Zhao et al. (2018) suggested a variant of GRU called PARALLEL-RES GRU. In this case, authors aimed to create a parallel architecture for converting a deep model into a multi-shallow residual model that effectively avoids deterioration.

Another short-term traffic flow system based on TDNN and optimized using Genetic Algorithm was proposed by Abdulhai et al. in 1999. (GA). AI-powered GA is capable of searching across very complicated areas with several local minima.

Gao et al. (2013) suggested a technique that combines WA and ANN in order to concentrate on the characteristics of time-variation and uncertainty of urban arterial traffic flow. To address the drawbacks of sluggish convergence, they adopted momentum factor as a training procedure in this instance.

Feng et al. (2017) suggested a traffic prediction system employing wavelet function and extreme machine learning to increase forecast accuracy (EML). ELM is an enhanced single hidden layer FFNN that offers quick learning and strong generalization capabilities.

Table 7 provides a comparison of the previously described approaches based on CNN, RNN, LSTM, GRU, TDNN, and WNN.

Table 7. NN-Based Traffic Prediction Studies

No	Author	Year	DL	SF (min)	Source	Area	Comparison
1	Chung et al.	2018	CNN	–	VD	–	–
2	Liao et al.	2018	CNN	–	LD: California PeMS	Freeway	SVR, NN, SAE,
3	Yu et al.	2017	CNN	5	LD + Sensors: BJER4, Caltrans PeMS (Beijing)	Highway, freeway	HA, LSVR, ARIMA, FFNN, LSTM, FC-LSTM
4	Ma et al.	2015	LSTM	2	RTMS Detector: Beijing Ring Road	Expressway	Elman NN, TDNN, NARX, NN, SVR, ARIMA, KF
5	Qiao et	2017	LSTM	1	Sensors:	Urban	SVM, ARIMA

	al.				Qingdao	arterial	
6	Zou et al.	2018	SLSTM	–	GPS: TaxiBJ, BikeNYC	Urban arterial	–
7	Abbas et al.	2018	SLSTM	1	Sensors: Motorway control system in Stockholm	Highway	–
8	Fu et al.	2016	LSTM, GRU	0.5	Sensors: PeMS	Freeway	ARIMA
9	Zhang et al.	2018	GRU	60	Sensors: Caltrans PeMS	Freeway	–
10	Zhao et al.	2018	PARALLEL–RES GRU	10	Sensors: UCI's PEMS-SF	Freeway	GRU
11	Abdulhai et al.	1999	TDNN+ GA	30s	LD: Interstate-5 Orange county, California	Freeway	MLFFNN
12	Gao et al.	2013	WNN	5	Traffic flow data,Qingdao	Urban	BP
13	Feng et al.	2017	WA, Eetreme ML	20s	Canadian Whitemud Drive data	Highway	R^2 value

RTMS: Remote Traffic Microwave Sensor

2.4.3.2 Unsupervised DL Techniques for Traffic Prediction

For the first time, Lv et al. (2015) employed layered AE models to exhibit temporal and geographical correlations in traffic flow data. In this model, supervised traffic flow prediction was accomplished by adding a logistic regression layer on top of the network. This concept did not suit well, though, with less information on traffic flow.

Traffic flow was predicted by Duan et al. (2016) both during the day and at night. A total of 250 tests were conducted in this study to train an SAE model. Additionally, a regression layer was added on top of SAE in this research. It was shown that MAE and RMSE are more valuable during the day than at night, and MRE is less valuable during the day than at night.

Using a layer-by-layer feature granulation unsupervised learning methodology, Yang et al. (2017) created an optimal network topology based on the Taguchi method and trained a deep SAE LM model to learn traffic feature. Even while employing more AE might enhance the accuracy of AE-based prediction models, the training time may rise as a result.

A two-level DL model in DBN, which includes a regression layer at the top and a DBN at the bottom, was employed by Huang et al. (2014). Multitask learning (MTL), in which many tasks are combined and the model is trained concurrently, is made possible by this method.

Koesdwiady et al. (2016) developed a rainfall integrated DBN and LSTM model that takes into account the influence of rainfall component in traffic flow data in order to study DL models with multi-source data inputs. The outcomes showed that taking weather into account greatly increases forecast accuracy.

The dynamic nature of traffic statistics means that the flow of traffic does not always follow the same pattern throughout the day. For instance, daytime traffic density is higher than nighttime traffic density. Zhang et al. (2018) used GA in this situation to derive the ideal hyperparameter for the DBN for various time intervals.

Table 8 compares the unsupervised DL methods for forecasting traffic that were previously covered. A comparison of following discussed DL models for traffic prediction is provided in Table 9.

Table 8. AE and DBN-Based Traffic Prediction Studies

No	Author	Year	DL	SF (min)	Source	Area	Comparison
1	Duan et al.	2016	SAE	15	PeMS	Freeway	–
2	Yang et al.	2017	SAE (optimal structure)	1	M6 freeway, UK	Freeway	EXP-LM, PSONN, RBFNN
3	Huang et al.	2014	DBM	15	PeMS, Highway system of china	Freeway, Highway	–
4	Koesdwiady et al.	2016	DBM	5	PeMS	Freeway	ARIMA, ANN
5	Zhang et al.	2018	DBN, GA	0.5	PeMS	Freeway	ANN, SAE, RNN, RW

EXP-LM: Exponential smoothing and the LM algorithm with NNs (EXP-LM), PSONN: Particle swarm optimization algorithm with NNs

Table 9. Comparison of DL Methods

Model	Strengths	Weakness

ANN	• Non-linear	• Falls in local optimization
	• No assumption	• Bad on complexed problems
MLP	• Fault tolerance	• Bad on sequence and time series data
	• Self-learning	• Long training time
CNN	• Image recognition	• Need large amount of training
RNN	• Dynamic	• Updates multiple parameters
	• Temporal dependency	• Exploding gradient
WNN	• Self-learning	• Lacking basis on selecting parameters
	• Fault tolerance	
	• Wavelet transform time-frequency localization	
AE	• Feature extraction	• Time complexity
	• Reduces dimensional feature space	
DBN	• Classification	• Time complexity due to parameter initialization
	• Uses of hidden layers effectively	

2.4.3.3. Hybridizations of DL Methods for Traffic Prediction

LSTM and CNN were coupled by Wu et al. (2016) to forecast traffic flow. While LSTM obtains temporal information from the traffic data, CNN in the model captured spatial patterns. As a next step, characteristics from the LSTM and CNN modules were combined to enhance traffic flow prediction. Conv-LSTM is a model created by Liu et al. (2017) that also combined CNN and LSTM.

Another DNN that combines CNN and LSTM was proposed by Duan et al. (2018) to extract spatial and temporal characteristics from GPS trace data in metropolitan areas. The method was trained using a greedy reinforcement method to cut training time and increase network accuracy.

Fouladgar et al. (2017) suggested a scalable, decentralized traffic flow prediction based on a hybrid CNN/LSTM technique to address the issue of centralization. This design was suggested to use the 2D input structure in the field of image processing. Additionally, it was found in this research that, in the absence of historical data, traffic flow for a junction may be predicted using traffic data measurements from nearby nodes. However, as the depth of the network increases, the training of such hybrid networks becomes difficult and time-consuming.

Du et al. (2017) suggested a hybrid framework based on CNN and RNN to manage the non-linear and non-stationary characteristics of traffic flow data. The latter model, which has an LSTM unit, captured short and long term temporal dependencies whereas the earlier model collected local trend aspects.

A model that incorporates stacked LSTM and AE was proposed by Yu et al. (2017) to assist in bridging the gap between supervised and unsupervised learning. With the use of a linear regression layer, these two models were integrated. The latent representation of static characteristics across accidents was extracted using AE. The idea was assessed using data that has been pooled every five minutes.

A hybrid model for predicting traffic based on decomposition was proposed by Zhong et al. (2018). This model used a mode decomposition and mode combination technique to first examine the periodic and random characteristics of the traffic flow data. Next, prediction models were adjusted to the subsequence's complexity. According to the complexity of the subsequence, authors established integration of the BP, ϵ-SVR, and LSTM models in this work.

A comparison of the above discussed hybrid models is provided in Table 10.

Table 10. Comparison of Hybrid DL Based Traffic Flow Prediction Proposals

No	Author	Year	DL	SF (min)	Data Source	Area	Comparison
1	Wu et al.	2016	CNN + LSTM	5	PeMS	Freeway	LSTM, SAE, Gradient boosting regression
2	Liu et al.	2017	CNN + LSTM	0.5	PeMS	Freeway	CNN-LSTM
3	Duan et al.	2018	CNN + LSTM	30	GPS: Xian taxis trajectory data	Urban arterial	Linear model, CNN, LSTM
4	Fouladgar et al.	2017	CNN+ LSTM	5	Sensor: PeMS	Freeway	–
5	Yu et al.	2017	AE + LSTM	5	California highway patrol, LA Department of transportation	Highway, Arterial streets	ARIMA, RW, HA, SARIMA
6	Zheng et al.	2018	NN+SVR +LSTM	5	PeMS	Freeway	–

Chapter 3. Freeway Speed Prediction Methods

3.1 Introduction

Traffic speed prediction is a regression issue related to time series data which can be stated as follows. Let X_i^t represent the observed traffic speed at i th point during the t th time interval on a freeway. Providing a sequence $\{X_i^t\}$ of observed speed, $i = 1, 2, \ldots,$ N, $t = 1, 2, \ldots, T$, the task is to predict the traffic speed at time $(t + \Delta)$ for horizon size Δ. Without any assumptions, deep neural networks (DNNs) are a type of Artificial NNs inspired by human neurons. It can mine traffic data by extracting features generated by hierarchical and distributed architecture. CAV should be able to anticipate its future speed based on the current state of its leading vehicle. Deep learning techniques, a relatively new field of technology, have proven to have highly sophisticated computational capabilities. In this chapter, the suggested deep learning approach's prediction accuracy will be measured, and its findings will be compared with the conventional method.

This chapter is organized as follows. Section 3.2 presents the supervised deep learning model for traffic speed prediction. Section 3.3 presents the unsupervised deep learning model for traffic speed prediction. Section 3.4 describes the Intelligent Driver Model. Finally, in section 3.5, a summary concludes this chapter.

3.2 Supervised Deep Learning Method

Given that Recurrent NNs can remember long-term dependencies, it was well-known to handle sequential traffic data. However, it encounters the problem of vanishing gradient when timesteps increase. To solve it, the variant Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) were developed. LSTM was first introduced by Hochreiter and Schmidhuber (1997) for language processing and used in traffic flow prediction by Ma et al. (2015). Different from RNNs, LSTM regards the hidden layer as a memory cell, which makes it outperform RNNs due to its ability to flexibly memorize patterns for longer durations. To make the training process more effective and concise, GRU was introduced by Chung et al. (2014). It removed the separate memory unit without reducing the performance compared to LSTM. Figure 3 shows the structure of GRU.

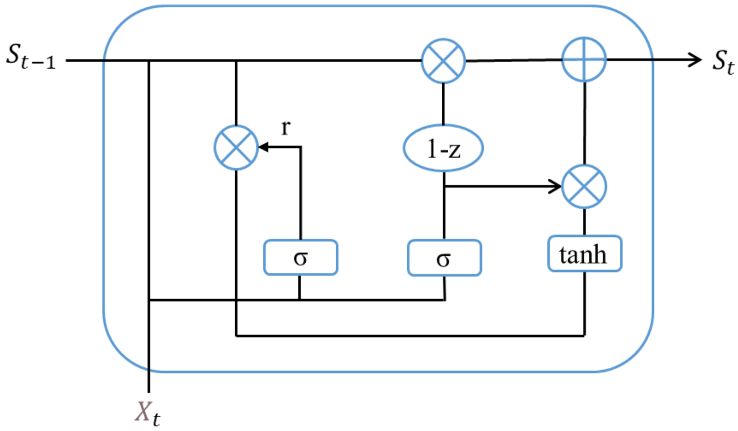

Figure 3. Structure of GRU

In GRU, the memory unit comprises two gates, namely the reset gate and the update gate, which decide what information should be sent to the output layer. It merges the input gate and the update gate into the reset gate, which performs similarly to the LSTM forget gate in that it selects whether to integrate previous and present information, while the update gate determines how much previous information to retain. Equations are given below:

$$r = \sigma(X_t U_r + S_{t-1} W_r) \tag{1}$$
$$z = \sigma(X_t U_z + S_{t-1} W_z) \tag{2}$$
$$h = \tanh(X_t U_h + (S_{t-1} * r) W_h) \tag{3}$$
$$S_t = (1 - z) * S_{t-1} + z * h) \tag{4}$$

Where X_t is input, r is reset gate, z is update gate, h is hidden state output, S_t is output, all of them are vectors U and W are corresponding weight parameter matrices for them. GRU uses the sigmoid function σ to activate reset and update gate. It outputs a value from 0 to 1, where 0 denotes no information go through while 1 denotes all information go through the cell state. The *tanh* function is used to activate the hidden state and outputs a number from −1 to 1.

After the hyperparameter tuning by a manual search, this study designs a 2-hidden layers architecture GRU with 32 neuron units. To avoid the overfitting problem, dropout regularization (Srivastava et al., 2014) is set as 0.2. RMSprop (Hinton et al., 2012) is selected as the optimizer, which is a modification of Stochastic Gradient Descend with adaptive learning rates and better in RNNs to prevent local minimum. Mean square error is utilized as the loss function and the goal is to minimize it. Datasets are classified with 128 batch sizes and trained with 100 epochs.

3.3 Unsupervised Deep Learning Method

Auto-Encoders (AEs) are a typical unsupervised learning method using unlabeled training (Liou et al., 2014). AEs are made up of two basic parts: encoder and decoder, where the encoder compresses the input x whereas the decoder reconstructs the input x'. Similar to the neural network, it also owns one or more hidden layers, and the number of units in the input layer and output layer are the same. They can be used for data compression and fusion since they generate comparable input at the output layer. Backpropagation (BP) algorithms are also used to minimize the error function by adjusting the weight parameters, and return a target value that is equal to the input.

Stacked AEs (SAEs) are the most prevalent AEs variants, in which numerous AEs are stacked into hidden layers using greedy layer-wise training (Bengio et al., 2007). Each AE receives bottleneck activation vector output from lower layers as input. The mechanism of it is to encode the feature vector extracted from the input via an encoder layer, and next, the feature from the previous layer is sent to the following layer until the training process finishes. Last, the input is reconstructed in the decoder layer. Equations are given below:

$$y = f(Wx + b) \tag{5}$$
$$x' = g(W'y + b') \tag{6}$$
$$\theta = \arg\min \frac{1}{2}\sum_{i=1}^{N} \|x - x'\|^2 \tag{7}$$

Where f and g are sigmoid functions used to activate the encoder and decoder layer, b and b' are the encoder and decoder bias vector respectively, W and W' are weight matrices for encoding and decoding. The parameters are trained by minimizing the error between reconstructed and actual input, which are defined as θ.

This study first designs 3 independent AEs and SAEs that utilize the same hidden layer with 128 neuron units. Dropout regularization is set as 0.2. Adam (Kingma and Ba, 2012) is selected as the optimizer, which is a combination of RMSprop and Momentum and used for Backpropagation Through Time. Mean square error is utilized as the loss function. To ensure the same iterations, datasets are also classified with 128 batch sizes and trained with 100 epochs.

3.4 Simulated Car-Following Model

The Intelligent Driver Model (IDM) is a conventional car-following model based on the present state of the object vehicle. The core principle of it involves comparing the object vehicle’s desired velocity to its real velocity collected from the sensors, as well as comparing its desired headway to its true headway to determine the vehicle's acceleration rate. Equations are given below:

$$a = a_m \left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*(v,\Delta v)}{s}\right)^2\right] \tag{8}$$

$$s^*(v, \Delta v) = s_0 + s_1\sqrt{\frac{v}{v_0}} + vT + \frac{v \cdot \Delta v}{2\sqrt{a_m b}} \tag{9}$$

Where the values of all the parameters in this study are adapted from (Treiber et al., 2000; Liu and Fan, 2021). a is the acceleration rate of the object vehicle, a_m is the maximum acceleration rate and equals 0.73 m/s^2, v is the current speed of the object vehicle, v_0 is the desired velocity and equals the speed limit m/s, δ is the acceleration exponent and equals 4, $s^*(v,\Delta v)$ is the desired minimum headway, Δv is the velocity difference between the object and the leading vehicle, s is the current headway between the object and the leading vehicle; s_0 is the linear jam gap and equals 2 m; s_1 is the non-linear jam gap and equals 3 m, T is the desired headway and equals 1.0 s, b is the comfortable deceleration rate and equals 1.67 m/s^2. It is worth mentioning that there are five parameters, v_0 desired velocity, a_m maximum acceleration rate, b comfortable deceleration rate, T desired headway, s_0 linear jam gap can be calibrated in the simulation according to various scenarios.

The IDM car-following model is applied in the microscopic "Simulation of Urban MObility" (SUMO) to predict the traffic speed, which is an open access platform developed by the German Institute of Transportation Systems. It supports multiple car-following models including Wiedemann 99, Krauss, IDM, Adaptive Cruise Control (ACC), and Cooperative Adaptive Cruise Control (CACC). It contains numerous default parameters to describe traffic flow characteristics and driver behavior. But it also allows users to input other values for the parameters. SUMO also provides a Traffic Control Interface (TraCI) for users to retrieve and change the attribute values of network objects during the simulation, e.g., vehicles, roads, traffic lights, and pedestrians. In addition, it includes other supporting tools handling network import, operational performance measurements, and emission calculations. The default lane-changing model is LC2013.

Chapter 4. Experimental Settings

4.1 Introduction

As discussed in the literature review conducted in Chapter 2, this chapter will identify potential freeway segment and collect necessary traffic data related to the selected freeway segment. The study site used for the conduct of the case study in this paper is a basic freeway segment that is selected through the Caltrans Performance Measurement System (PeMS) database. This potential freeway has a total length of about 1 mile.

The following sections are organized as follows. Section 4.2 presents information on the selected freeway segment. Section 4.3 presents the data collection and processing. Finally, section 4.4 shows the performance evaluation metrics that are used for those three models.

4.2 Potential Freeway Segment

The freeway segment is around the city of San Francisco, a large population area. The site is selected because of its preexisting congestion issues during the peak hour, and because it is the major interstate freeway with high traffic volumes. Figure 4 shows the San Francisco part in the PeMS. The study area is a mainline segment of I-80 freeway eastbound in the west of Berkeley district (marked with a blue rectangle area).

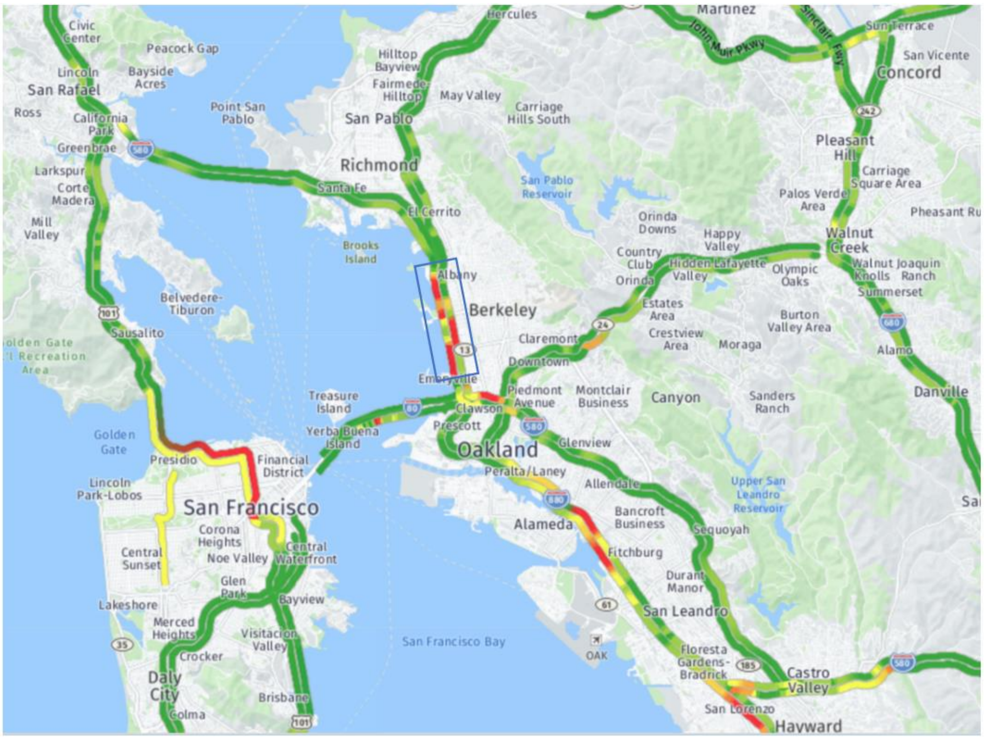

Figure 4. San Francisco Part in the PeMS

Figure 5 provides the global view of the specific study area. It is a total length of 1 mile section around 10.41 postmile in a two-way interstate road with 5 lanes in each direction.

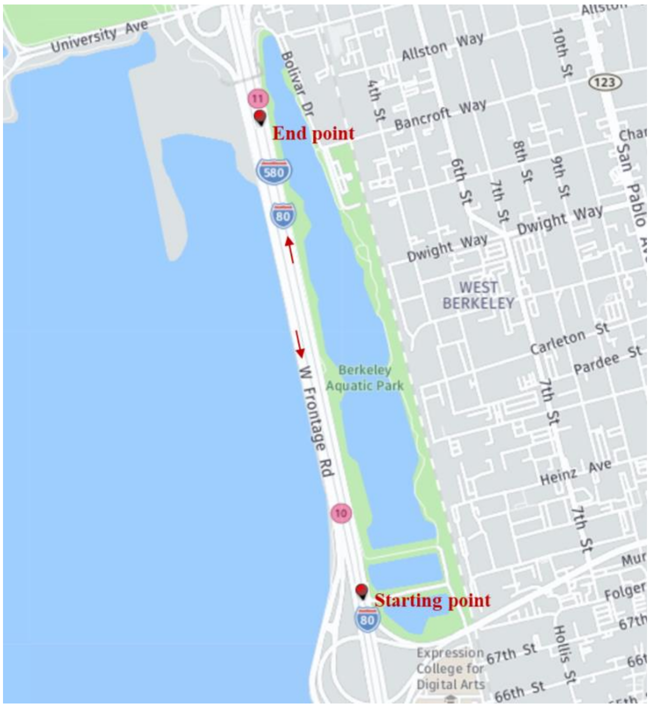

Figure 5. Global View of the Study Area (Source: PeMS)

Figure 6 provides a detailed configuration of the freeway segment. The starting point of the traffic flow is inside the red area. The blue lines in the freeway segment are vehicle detector stations (VDS), including vehicle detectors on each lane of the freeway. These vehicle detectors collect, store, and process real-time traffic data and sent it to PeMS. Table 11 shows an example of the roadway information provided by the vehicle detector station VDS 405589.

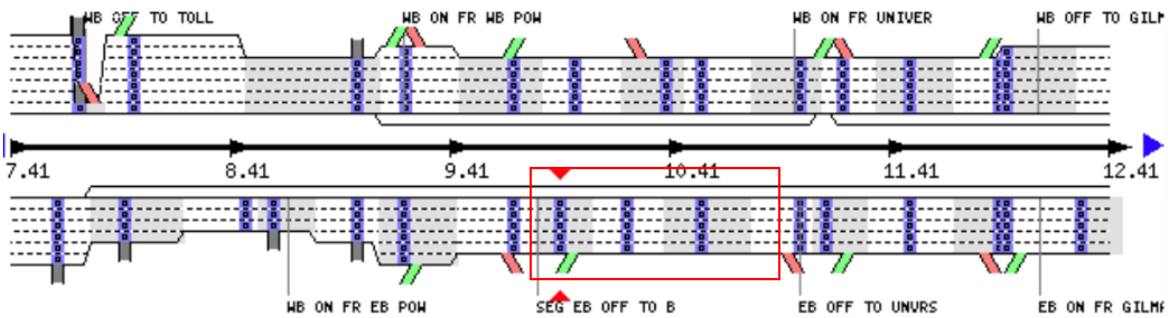

Figure 6. Configuration of the Freeway Segment

Table 11. Roadway Information of VDS 405589

Roadway Information (from TSN)	
Road Width	60 ft
Lane Width	12.0 ft
Inner Shoulder Width	3 ft
Inner Shoulder Treated Width	3 ft
Outer Shoulder Width	10 ft
Outer Shoulder Treated Width	10 ft
Design Speed Limit	70 mph
Functional Class	Principal Arterial W/ C/L Prin Arterial
Inner Median Type	Separate Grades
Inner Median Width	36 ft
Terrain	Flat
Population	Urbanized
Barrier	Guardrail in Median left Roadway
Surface	Base & Surface >= 7" Thick
Roadway Use	Median Lane is HOV Lane

4.3 Data Collection and Processing

The data is derived from the Caltrans Performance Measurement System (PeMS), which contains data from about 40,000 inductive loop detectors across the highway network in California. Each vehicle detector station collects data every 30 seconds and is aggregated into 5-minute time intervals. Due to the unique patterns of various sequential traffic speed data and that no single pattern can match all the time series data, this study uses the information gathered by a unitary detector.

The experimental scenario is a mainline segment of the I-80 freeway eastbound, Berkeley. It is a two-way road with five lanes in each direction, and the average traffic speed from south to north is selected. Since the traffic speed data is periodic and its pattern can differ between weekdays and weekends. This study collects data from March 1st to April 29th on the weekdays of 2022. According to Chen et al. (2012), 5-minute traffic is more suitable and predictable. In this experiment, the past 1 hour which is a time sequence of 12 data points is used to predict the coming average traffic speed in the next 5 minutes. Incorporating the periodicity of traffic data over weeks, the whole dataset is divided into training and testing sets. The first 33 days (75%) are used as the training set, and the last 11 days (25%) are used as the testing set.

Before training the dataset, normalization is a necessary step to accelerate the gradient descent speed (Zhang and Kabuka, 2018). This study first implements a feature scaler by the training set, then uses the MinMaxScaler to normalize the training set and test set separately. After scaling, data are normalized from 0 to α, where α is a standardized factor that is set as 1 for simpleness. The equation is given below:

$$s = \alpha \times \frac{x - \min(x)}{[\max(x) - \min(x)]} \tag{10}$$

Considering the size of the dataset and the number of hyperparameters, 90% of data is used as training and 10% as validation. Since the sequential traffic prediction needs to use the historical speed to predict the incoming speed, the time lag is utilized to divide the dataset. The divided dataset still has a time series feature, and this study samples the dataset in order and then shuffles it. Given the modularity and user-friendly interface, the Keras framework which is released in 2015 is used to train the deep learning models and it can run over the popular TensorFlow and Theano.

4.4 Performance Evaluation Metrics

To test the prediction accuracy of different models in a comprehensive perspective, there are five metrics mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), root mean square error (RMSE), and R^2 are applied to evaluate the performance. Equations are given below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x_i}| \tag{11}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|x_i - \hat{x_i}|}{x_i} \tag{12}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x_i})^2 \tag{13}$$

$$RMSE = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(x_i - \hat{x_i})^2} \tag{14}$$

Where x_i is the actual average traffic speed, and $\hat{x_i}$ is the predicted average traffic speed. The lower these metrics, the better the performance.

Chapter 5. Results and Discussions

5.1 Introduction

This chapter presents the numerical results of the traffic speed prediction on the freeway. The collected traffic speed data is divided into two subsets, the training set and the testing set. The training set is used to train the two deep learning models. And the testing set is used to predict the accuracy of the three proposed models. The prediction accuracy is compared between the deep learning models and the IDM. The chapter is organized as follows. Section 5.2 describes the performance of the training set for the proposed models. Section 5.3 presents prediction accuracy of different methods.

5.2 Performance of the Training Set

This section first shows the training results for both supervised and unsupervised deep learning models, then illustrates the prediction accuracy of three different models and compares the performance by the time of day. Finally, the evolution trend is displayed over time. Figure 7 shows the changes in loss function of GRU and SAEs. The loss function is used to measure the degree of consistency between the estimated value of the model and the real value. It is a non-negative real-valued function. The smaller the loss function, the better the robustness of the model. The loss rates of the training set with black line drop rapidly at the beginning before 20 epochs for both GRU and SAE. With the increase of time, the loss rate of the GRU training set tends to remain flat at a relative minimum value and is infinitely close to 0. For the GRU validation set, there is a small oscillation at the beginning. As the epoch increases, the loss rate continues to decrease, which indicates that the network is still learning. It eventually stabilizes and the validation set converges well, avoiding underfitting and overfitting problems. For the validation set of SAEs, the volatility is significantly larger than that of the supervised learning algorithm. However, it finally stabilizes and fits the training set as the epoch increases. From the performance of the loss function, both deep learning networks are well trained.

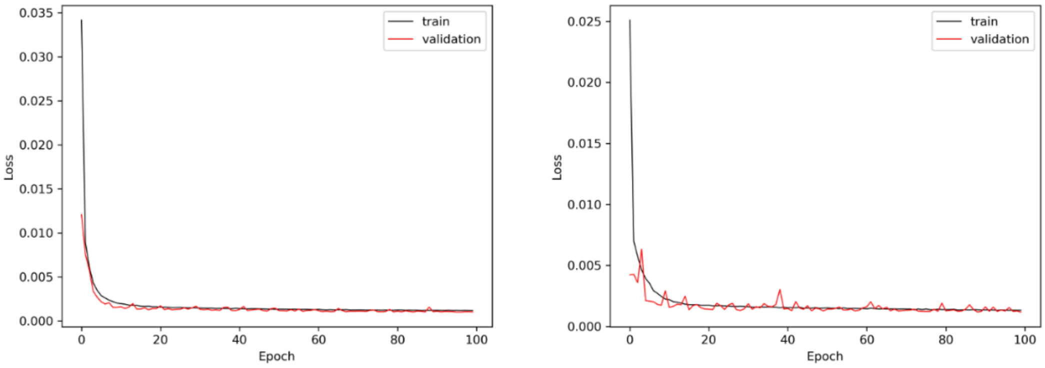

Figure 7. The Loss Rate of GRU (Left) and SAEs (Right)

5.3 Prediction Accuracy of Different Methods

Table 12 illustrates the performance of each model based on different statistical metrics. It can be seen that for the MAE, MSE, and RMSE that describe the absolute error, the unsupervised deep learning represented by SAEs is modestly higher than the supervised deep learning represented by GRU, and the performances of both are better than the traditional IDM model. For MAPE describing a relative error, GRU also performs modestly better (3.410%) than SAEs (3.478%), and both outperform the IDM model (5.240%). For the degree of fitness, the R^2 of them are similar (floating around 0.986), demonstrating a relatively good fitting result. Overall, both supervised learning and unsupervised learning methods are superior to the traditional simulation-based car-following model in the prediction of traffic speed. While the difference between the two different deep learning is small, GRU is slightly better than SAEs in time series prediction. This plays an important role in the application of prediction technology in ITS.

Table 12. Performance Comparison of Different Models

Model	MAE	MAPE	MSE	RMSE	R2
GRU	1.345	3.341%	4.535	2.130	0.986
SAEs	1.340	3.347%	4.334	2.082	0.987
IDM	2.486	5.240%	8.896	2.983	0.986

Figure 8 demonstrates the prediction of average speed for different models by the time of day. The actual value is selected as a baseline with a solid red line. To account for the different traffic states, it is divided into three intervals according to the size of the traffic flow (with dash blue line), low traffic loads, transition state, and heavy traffic loads.

For low traffic loads, it can be classified into two time periods, Before congestion (0:00-7:00) and After congestion (19:00-0:00). It can be seen that before congestion, both

GRU and SAEs match well with real value. Although IDM model changes more softly, the response at high speed is not timely enough. After congestion, the IDM model cannot revert to the previous accuracy, and there is a small gap with the original value, but both GRU and SAEs can maintain high accuracy. This shows that the deep learning network can mitigate cumulative error propagation over time. Given that the IDM model is collision-free when the distance between the front and rear vehicles decreases sharply, the IDM model will produce strong braking on the target vehicle, which is unrealistic in reality.

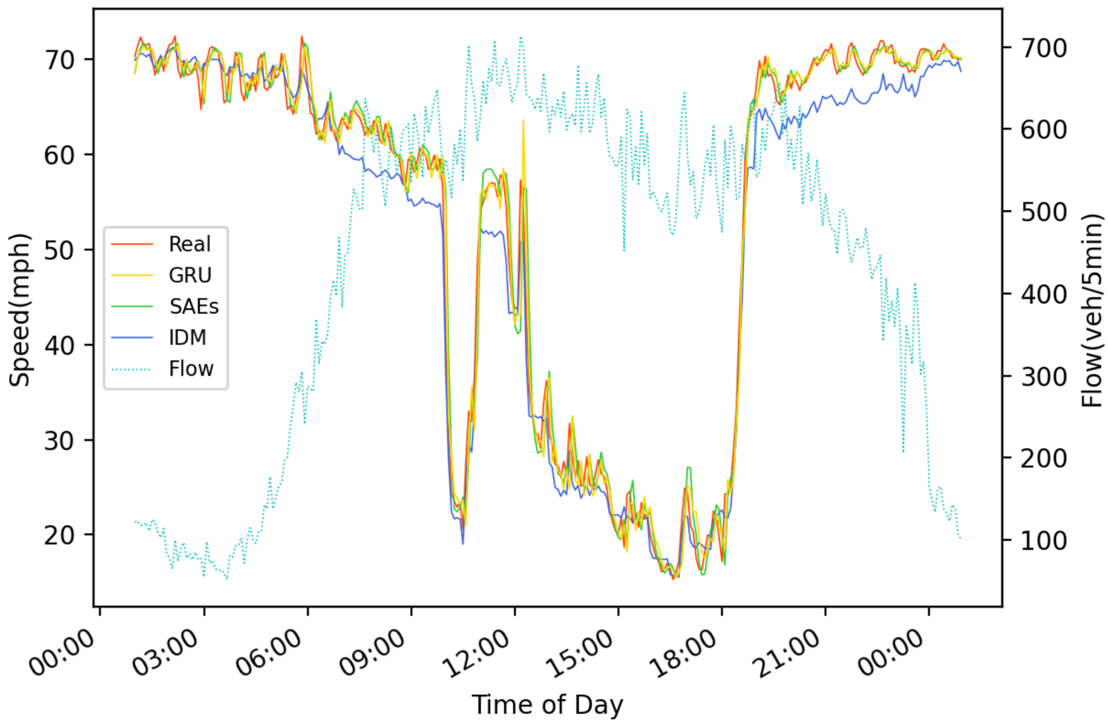

Figure 8. Prediction of the Average Speed of Different Models by the Time of Day

This is also the problem with the simulation-based car-following model. Transition state is classified into Buildup of congestion (7:00-10:00, 12:00-15:00) and Dissipation of congestion (11:00-12:00, 18:00-19:00). For the buildup of congestion, IDM's performance is inferior to deep learning networks. In addition, IDM still cannot rebound to the previous accuracy in dissipation of congestion. According to the length of the congestion time, heavy traffic loads are classified into Short-term full congestion (10:00-11:00), Long-term full congestion (15:00-18:00). In short-term full congestion, all models have different degrees of bias, and the most obvious one goes to the IDM. For long-term full congestion (15:00-18:00), the situation is similar to the before congestion state under the low traffic loads. The three models perform almost the same, but IDM is smoother and with less fluctuation.

This study also investigates the speed distribution for different models by the time of day with a heatmap, which is displayed in Figure 4. There are two points worth noting. Firstly, for a short period from 10:00 to 10:05, there is a certain prediction delay for both GRU and SAEs, and this phenomenon can continue until the congestion dissipates at 18:00. But this situation does not exist in the IDM model, which suggests that for short-term slowdowns, IDM can detect the buildup of congestion earlier than deep learning networks. Another finding is that after congestion at 18:30, all models have a prediction lag of about five minutes. But from the dark blue area afterward, the accuracy of deep learning networks recovers faster than IDM. The above analysis reveals that deep learning networks and simulation-based car-following models have their latent performance features for different time dimensions.

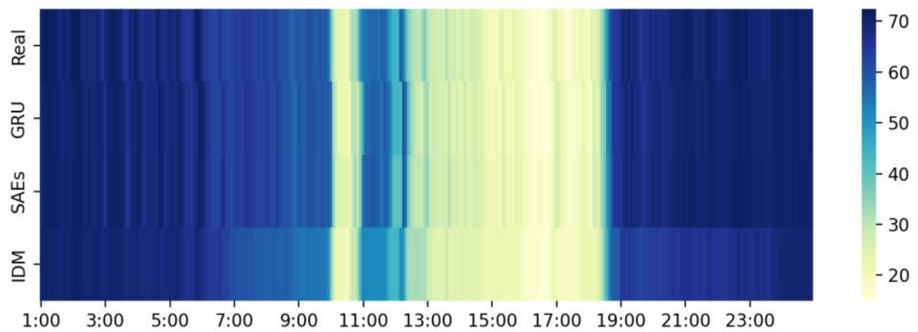

Figure 9. Speed Distribution for Different Models by the Time of Day

Chapter 6. Summary

6.1 Conclusions

The development of Intelligent Transportation Systems has given impetus to intelligent vehicles, which have the potential to address the traffic congestion problem. Meanwhile, it also brings real-time traffic prediction issues. Given the complex and dynamic spatiotemporal dependency embedded in traffic data, traditional prediction models have many drawbacks.

In order to improve the accuracy of traffic speed prediction, this study focuses on emerging deep neural networks using real-world traffic data. Additionally, a simulation-based model is built for intelligent vehicles in SUMO. A series of statistical evaluation metrics, MAE, MAPE, MSE, RMSE, and R^2 are employed to assess the prediction accuracy of the supervised learning method, unsupervised learning method, and simulation-based model. The PeMS dataset is used to train and evaluate the constructed DNNs, and the results suggest that both GRU and SAEs outperform the traditional IDM model in the prediction of traffic speed on the freeway. In addition, there is no difference between the deep learning networks, and GRU outperforms SAEs slightly in time series prediction. It also demonstrates that car-following simulation-based models and deep learning networks both contain latent performance attributes for various time dimensions under low, transition state, and heavy traffic loads. This has a significant impact on how prediction technology is applied in ITS. The outcomes can assist researchers and traffic engineers to improve dynamic traffic control, such as highway operation, bottleneck detection, and Level of Service assessment. The predicted traffic speed can also be used for further research on variable speed limit control, platooning management, and route guidance, etc.

6.2 Future Work

This study mainly uses traffic speed as the input for prediction. Future research work can introduce hand-engineering factors, such as weather, events, and other traffic parameters. Moreover, more spatiotemporal dependency can be captured by more advanced deep learning networks. In addition, attention mechanism can be combined to model the long sequence data (Zheng et al., 2020). For the simulation environment, it can focus on improving the car-following model (Salles et al., 2020). The lane changing model can also be considered to better simulate intelligent driving behaviors. Lastly, the transferability issue that all adaptive frameworks face could be addressed, especially in metropolitan areas.

References

1. Abbas, Z., Al-Shishtawy, A., Girdzijauskas, S., & Vlassov, V. (2018, July). Short-term traffic prediction using long short-term memory neural networks. In 2018 IEEE International Congress on Big Data (BigData Congress) (pp. 57-65). IEEE.
2. Abdulhai, B., Porwal, H., & Recker, W. (1999). Short term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks.
3. Abdulhai, B., Porwal, H., & Recker, W. (1999). Short term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks.
4. Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19.
5. Castro-Neto, M., Jeong, Y. S., Jeong, M. K., & Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert systems with applications, 36(3), 6164-6173.
6. Castro-Neto, M., Jeong, Y. S., Jeong, M. K., & Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert systems with applications, 36(3), 6164-6173.
7. Chen, C., Li, K., Teo, S. G., Zou, X., Wang, K., Wang, J., & Zeng, Z. (2019, July). Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 485-492).
8. Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., & Feng, X. (2020, April). Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 04, pp. 3529-3536).
9. Chung, J., & Sohn, K. (2017). Image-based learning to measure traffic density using a deep convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, 19(5), 1670-1675.
10. Chung, J., & Sohn, K. (2017). Image-based learning to measure traffic density using a deep convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, 19(5), 1670-1675.
11. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
12. Coifman, B. (2015). Empirical flow-density and speed-spacing relationships: Evidence of vehicle length dependency. Transportation Research Part B: Methodological, 78, 54-65.
13. Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Transactions on Intelligent Transportation Systems, 21(11), 4883-4894.
14. Davis, G. A., & Nihan, N. L. (1991). Nonparametric regression and short-term freeway traffic forecasting. Journal of Transportation Engineering, 117(2), 178-188.

15. Davis, G. A., & Nihan, N. L. (1991). Nonparametric regression and short-term freeway traffic forecasting. Journal of Transportation Engineering, 117(2), 178-188.
16. Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29.
17. Do, W., Rouhani, O. M., & Miranda-Moreno, L. (2019). Simulation-based connected and automated vehicle models on highway sections: a literature review. Journal of Advanced Transportation, 2019.
18. Du, S., Li, T., Gong, X., Yang, Y., & Horng, S. J. (2017, November). Traffic flow forecasting based on hybrid deep learning framework. In 2017 12th international conference on intelligent systems and knowledge engineering (ISKE) (pp. 1-6). IEEE.
19. Duan, Y., Lv, Y., & Wang, F. Y. (2016, July). Performance evaluation of the deep learning approach for traffic flow prediction at different times. In 2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI) (pp. 223-227). IEEE.
20. Duan, Z., Yang, Y., Zhang, K., Ni, Y., & Bajgain, S. (2018). Improved deep hybrid networks for urban traffic flow prediction using trajectory data. Ieee Access, 6, 31820-31827.
21. Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems, 28.
22. EDES, Y. J. S., Michalopoulos, P. G., & Plum, R. A. (1980). Improved estimation of traffic flow for real-time control. Transportation Research Record, 95, 28-39.
23. Feng, W., Chen, H., & Zhang, Z. (2017, November). Short-term traffic flow prediction based on wavelet function and extreme learning machine. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 531-535). IEEE.
24. Feng, W., Chen, H., & Zhang, Z. (2017, November). Short-term traffic flow prediction based on wavelet function and extreme learning machine. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 531-535). IEEE.
25. Fouladgar, M., Parchami, M., Elmasri, R., & Ghaderi, A. (2017, May). Scalable deep traffic flow neural networks for urban traffic congestion prediction. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2251-2258). IEEE.
26. Fu, R., Zhang, Z., & Li, L. (2016, November). 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.
27. Fu, R., Zhang, Z., & Li, L. (2016, November). 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.

28. Gallego, J. L. (2009). The exact likelihood function of a vector autoregressive moving average process. Statistics & probability letters, 79(6), 711-714.
29. Gao, J., Leng, Z., Qin, Y., Ma, Z., & Liu, X. (2013, May). Short-term traffic flow forecasting model based on wavelet neural network. In 2013 25th Chinese control and decision conference (CCDC) (pp. 5081-5084). IEEE.
30. Gora, P., Katrakazas, C., Drabicki, A., Islam, F., & Ostaszewski, P. (2020). Microscopic traffic simulation models for connected and automated vehicles (CAVs)–state-of-the-art. Procedia Computer Science, 170, 474-481.
31. Helbing, D., Treiber, M., Kesting, A., & Schönhof, M. (2009). Theoretical vs. empirical classification and prediction of congested traffic states. The European Physical Journal B, 69(4), 583-598.
32. Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, 14(8), 2.
33. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
34. Hou, Y., Edara, P., & Sun, C. (2014). Traffic flow forecasting for urban work zones. IEEE transactions on intelligent transportation systems, 16(4), 1761-1770.
35. Hua, J., & Faghri, A. (1994). Apphcations of artificial neural networks to intelligent vehicle-highway systems. Transportation Research Record, 1453, 83.
36. Hua, J., & Faghri, A. (1994). Apphcations of artificial neural networks to intelligent vehicle-highway systems. Transportation Research Record, 1453, 83.
37. Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Transactions on Intelligent Transportation Systems, 15(5), 2191-2201.
38. Ishak, S., & Al-Deek, H. (2002). Performance evaluation of short-term time-series traffic prediction model. Journal of transportation engineering, 128(6), 490-498.
39. Ishak, S., & Al-Deek, H. (2002). Performance evaluation of short-term time-series traffic prediction model. Journal of transportation engineering, 128(6), 490-498.
40. Jabari, S. E., Zheng, J., & Liu, H. X. (2014). A probabilistic stationary speed–density relation based on Newell's simplified car-following model. Transportation Research Part B: Methodological, 68, 205-223.
41. Jia, Z., Chen, C., Coifman, B., & Varaiya, P. (2001, August). The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585) (pp. 536-541). IEEE.
42. Kamarianakis, Y., & Prastacos, P. (2003). Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. Transportation Research Record, 1857(1), 74-84.
43. Kamarianakis, Y., & Prastacos, P. (2003). Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. Transportation Research Record, 1857(1), 74-84.

44. Kesting, A., Treiber, M., Schönhof, M., & Helbing, D. (2007). Extending adaptive cruise control to adaptive driving strategies. Transportation Research Record, 2000(1), 16-24.
45. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
46. Koesdwiady, A., Soua, R., & Karray, F. (2016). Improving traffic flow prediction with weather information in connected cars: A deep learning approach. IEEE Transactions on Vehicular Technology, 65(12), 9508-9517.
47. Krajewski, R., Bock, J., Kloeker, L., & Eckstein, L. (2018, November). The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 2118-2125). IEEE.
48. Larry, H. K. (1995). Event-based short-term traffic flow prediction model. Transportation Research Record, 1510, 45-52.
49. Ledoux, C. (1997). An urban traffic flow model integrating neural networks. Transportation Research Part C: Emerging Technologies, 5(5), 287-300.
50. Leshem, G., & Ritov, Y. A. (2007). Traffic flow prediction using adaboost algorithm with random forests as a weak learner. International Journal of Mathematical and Computational Sciences, 1(1), 1-6.
51. Li, D. M., & Liu, B. I. N. (2014, July). Modeling and prediction of highway traffic flow based on wavelet neural network. In 2014 International Conference on Machine Learning and Cybernetics (Vol. 2, pp. 675-679). IEEE.
52. Li, J., Gao, L., Song, W., Wei, L., & Shi, Y. (2018, November). Short term traffic flow prediction based on LSTM. In 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP) (pp. 251-255). IEEE.
53. Li, L., Yan, J., Yang, X., & Jin, Y. (2021). Learning interpretable deep state space model for probabilistic time series forecasting. arXiv preprint arXiv:2102.00397.
54. Liao, S., Chen, J., Hou, J., Xiong, Q., & Wen, J. (2018, July). Deep convolutional neural networks with random subspace learning for short-term traffic flow prediction with incomplete data. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-6). IEEE.
55. Liou, C. Y., Cheng, W. C., Liou, J. W., & Liou, D. R. (2014). Autoencoder for words. Neurocomputing, 139, 84-96.
56. Liu, H., Kan, X. D., Shladover, S. E., Lu, X. Y., & Ferlis, R. E. (2018). Modeling impacts of cooperative adaptive cruise control on mixed traffic flow in multi-lane freeway facilities. Transportation Research Part C: Emerging Technologies, 95, 261-279.
57. Liu, J., Wu, N., Qiao, Y., & Li, Z. (2021). A scientometric review of research on traffic forecasting in transportation. IET Intelligent Transport Systems, 15(1), 1-16.
58. Liu, P., & Fan, W. (2021). Extreme Gradient Boosting (XGBoost) Model for Vehicle Trajectory Prediction in Connected and Autonomous Vehicle Environment. Promet-Traffic&Transportation, 33(5), 767-774.

59. Liu, Y., Zheng, H., Feng, X., & Chen, Z. (2017, October). Short-term traffic flow prediction with Conv-LSTM. In 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 1-6). IEEE.
60. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873.
61. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873.
62. Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies, 54, 187-197.
63. Miglani, A., & Kumar, N. (2019). Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. Vehicular Communications, 20, 100184.
64. Okutani, I., & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research Part B: Methodological, 18(1), 1-11.
65. Pamuła, T. (2018). Impact of data loss for prediction of traffic flow on an urban road using neural networks. IEEE Transactions on Intelligent Transportation Systems, 20(3), 1000-1009.
66. Pamuła, T. (2018). Impact of data loss for prediction of traffic flow on an urban road using neural networks. IEEE Transactions on Intelligent Transportation Systems, 20(3), 1000-1009.
67. Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies, 79, 1-17.
68. Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies, 79, 1-17.
69. Qiao, S., Sun, R., Fan, G., & Liu, J. (2017, November). Short-term traffic flow forecast based on parallel long short-term memory neural network. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 253-257). IEEE.
70. Rios-Torres, J., & Malikopoulos, A. A. (2017, October). Impact of connected and automated vehicles on traffic flow. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (pp. 1-6). IEEE.
71. Salles, D., Kaufmann, S., & Reuss, H. C. (2020, October). Extending the intelligent driver model in SUMO and verifying the drive off trajectories with aerial measurements. In SUMO User Conference.
72. Sen, R., Yu, H. F., & Dhillon, I. S. (2019). Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. Advances in neural information processing systems, 32.

73. Shi, X., Qi, H., Shen, Y., Wu, G., & Yin, B. (2020). A spatial–temporal attention approach for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 22(8), 4909-4918.
74. Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2017). Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE journal of biomedical and health informatics, 22(5), 1589-1604.
75. Siqueira, A. F., Peixoto, C. J., Wu, C., & Qian, W. L. (2016). Effect of stochastic transition in the fundamental diagram of traffic flow. Transportation Research Part B: Methodological, 87, 1-13.
76. Smith, B. L., & Demetsky, M. J. (1994, October). Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches. In Proceedings of IEEE international conference on systems, man and cybernetics (Vol. 2, pp. 1706-1709). IEEE.
77. Smith, B. L., & Demetsky, M. J. (1997). Traffic flow forecasting: comparison of modeling approaches. Journal of transportation engineering, 123(4), 261-266.
78. Song, C., Lin, Y., Guo, S., & Wan, H. (2020, April). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 914-921).
79. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.
80. Stathopoulos, A., Dimitriou, L., & Tsekeris, T. (2008). Fuzzy modeling approach for combined forecasting of urban traffic flow. Computer‐Aided Civil and Infrastructure Engineering, 23(7), 521-535.
81. Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. IEEE Transactions on intelligent transportation systems, 7(1), 124-132.
82. Sun, S., Zhang, C., & Zhang, Y. (2005, September). Traffic flow forecasting using a spatio-temporal bayesian network predictor. In International conference on artificial neural networks (pp. 273-278). Springer, Berlin, Heidelberg.
83. Talebpour, A., & Mahmassani, H. S. (2016). Influence of connected and autonomous vehicles on traffic flow stability and throughput. Transportation Research Part C: Emerging Technologies, 71, 143-163.
84. Tian, Y., & Pan, L. (2015, December). 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.
85. Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. Physical review E, 62(2), 1805.
86. Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. Physical review E, 62(2), 1805.
87. U.S. Federal Highway Administration. Next Generation Simulation Program (NGSIM). http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm 2006.

88. Van Arem, B., De Vos, A. P., & Vanderschuren, M. J. (1997). The microscopic traffic simulation model MIXIC 1.3.
89. Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transportation Research Part C: Emerging Technologies, 4(5), 307-318.
90. Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transportation Research Part C: Emerging Technologies, 4(5), 307-318.
91. VanderWerf, J., Shladover, S., Kourjanskaia, N., Miller, M., & Krishnan, H. (2001). Modeling effects of driver control assistance systems on traffic. Transportation Research Record, 1748(1), 167-174.
92. Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. Transport reviews, 24(5), 533-557.
93. Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. Transport reviews, 24(5), 533-557.
94. Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transportation Research Record, 1776(1), 194-200.
95. Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transportation Research Record, 1776(1), 194-200.
96. Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of transportation engineering, 129(6), 664-672.
97. Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of transportation engineering, 129(6), 664-672.
98. Wu, L., & Coifman, B. (2014). Improved vehicle classification from dual-loop detectors in congested traffic. Transportation Research Part C: Emerging Technologies, 46, 222-234.
99. Wu, Y., & Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022.
100. Wu, Y., & Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022.
101. Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121.
102. Yang, H. F., Dillon, T. S., & Chen, Y. P. P. (2016). Optimized structure of the traffic flow forecasting model with a deep learning approach. IEEE transactions on neural networks and learning systems, 28(10), 2371-2381.
103. Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875.
104. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017, June). Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017

SIAM international Conference on Data Mining (pp. 777-785). Society for Industrial and Applied Mathematics.
105.Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017, June). Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM international Conference on Data Mining (pp. 777-785). Society for Industrial and Applied Mathematics.
106.Zhang, D., & Kabuka, M. R. (2018). Combining weather condition data to predict traffic flow: a GRU-based deep learning approach. IET Intelligent Transport Systems, 12(7), 578-585.
107.Zhang, D., & Kabuka, M. R. (2018). Combining weather condition data to predict traffic flow: a GRU-based deep learning approach. IET Intelligent Transport Systems, 12(7), 578-585.
108.Zhang, Y., & Huang, G. (2018). Traffic flow prediction model based on deep belief network and genetic algorithm. IET Intelligent Transport Systems, 12(6), 533-541.
109.Zhao, B., & Zhang, X. (2018, May). A parallel-res GRU architecture and its application to road network traffic flow forecasting. In Proceedings of 2018 International Conference on Big Data Technologies (pp. 79-83).
110.Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3848-3858.
111.Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3848-3858.
112.Zheng, C., Fan, X., Wang, C., & Qi, J. (2020, April). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1234-1241).
113.Zheng, C., Fan, X., Wang, C., & Qi, J. (2020, April). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1234-1241).
114.Zheng, Z., Pan, L., & Pholsena, K. (2018, June). Mode decomposition based hybrid model for traffic flow prediction. In 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC) (pp. 521-526). IEEE.
115.Zhou, M., Qu, X., & Jin, S. (2016). On the impact of cooperative autonomous vehicles in improving freeway merging: a modified intelligent driver model-based approach. IEEE Transactions on Intelligent Transportation Systems, 18(6), 1422-1428.
116.Zou, Z., Gao, P., & Yao, C. (2018, June). City-level traffic flow prediction via LSTM networks. In Proceedings of the 2nd International Conference on Advances in Image Processing (pp. 149-153).