Autonomous Vehicles to Evolve to a New Urban Experience

DELIVERABLE D5.6

Transport service optimization approach and results

Co-funded by the Horizon 2020 programme of the European Union

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769033

Disclaimer

Document Information

Grant Agreement Number	**769033**
Full Title	Autonomous Vehicles to Evolve to a New Urban Experience
Acronym	AVENUE
Deliverable	D5.6 Transport service optimization approach and results
Due Date	31.4.2022
Work Package	WP5
Lead Partner	CERTH
Leading Author	Antonios Lalas
Dissemination Level	Public

Document History

Version	**Date**	**Author**	**Description of change**
0.1	17.06.2021	Maria Eleni Kadoglou, Leon Vitanos, Athanasios Papadakis, Nikolaos Dimitriou, Antonios Lalas, Konstantinos Votis, Dimitrios Tzovaras, CERTH	First report about predictive maintenance and traffic prediction
0.2	03.10.2022	Theoktisti Marinopoulou, Evangelos Athanasakis, Eleni Diamantidou, Ioannis Papadimitriou, Dimitrios Tsiktsiris, Anastasios Vafeiadis, Nikolaos Dimitriou, Antonios Lalas, Konstantinos Votis, Dimitrios Tzovaras,	Final report about predictive maintenance, traffic prediction and energy consumption

		CERTH	
0.3	10.10.2022	Antonios Lalas, Konstantinos Votis, Dimitrios Tzovaras, CERTH	Final version ready for review
0.4	27.10.2022	Antonios Lalas, Konstantinos Votis, Dimitrios Tzovaras, CERTH	Final version
0.5	01.09.2023	Maher Ben Moussa	Review and finalisation

Table of Contents

List of Figures

List of Tables

Acronyms

ADS	Automated Driving Systems
AI	Artificial Intelligence
AM	Automated Mobility
API	Application Protocol Interface
AV	Automated Vehicle
BM	Bestmile
BMM	Business Modelling Manager
CAV	Connected and Automated Vehicles
CB	Consortium Body
CERN	European Organization for Nuclear Research
D7.1	Deliverable 7.1
DC	Demonstration Coordinator
DI	The department of infrastructure (Swiss Canton of Geneva)
DMP	Data Management Plan
DSES	Department of Security and Economy - Traffic Police (Swiss Canton of Geneva)
DTU test track	Technical University of Denmark test track
EAB	External Advisory Board
EC	European Commission
ECSEL	Electronic Components and Systems for European Leadership
EM	Exploitation Manager
EU	European Union
EUCAD	European Conference on Connected and Automated Driving
F2F	Face to face meeting
FEDRO	(Swiss) Federal Roads Office
FOT	(Swiss) Federal Office of Transport
GDPR	General Data Protection Regulation
GIMS	Geneva International Motor Show
GNSS	Global Navigation Satellite System
HARA	Hazard Analysis and Risk Assessment
IPR	Intellectual Property Rights
IT	Information Technology
ITU	International Telecommunications Union
LA	Leading Author
LIDAR	Light Detection And Ranging
MEM	Monitoring and Evaluation Manager
MT	MobileThinking
OCT	General Transport Directorate of the Canton of Geneva
ODD	Operational Domain Design
OEDR	Object And Event Detection And Response
OFCOM	(Swiss) Federal Office of Communications
PC	Project Coordinator
PEB	Project Executive Board
PGA	Project General Assembly
PRM	Persons with Reduced Mobility
PSA	Group PSA (PSA Peugeot Citroën)
PTO	Public Transportation Operator
PTS	Public Transportation Services
QRM	Quality and Risk Manager
QRMB	Quality and Risk Management Board
RN	Risk Number
SA	Scientific Advisor
SAE Level	Society of Automotive Engineers Level (Vehicle Autonomy Level)
SAN	(Swiss) Cantonal Vehicle Service
SDK	Software Development Kit
SLA	Sales Lentz Autocars
SMB	Site Management Board
SoA	State of the Art
SOTIF	Safety Of The Intended Functionality
SWOT	Strengths, Weaknesses, Opportunities, and Threats.
T7.1	Task 7.1
TM	Technical Manager
TPG	Transport Publics Genevois
UITP	Union Internationale des Transports Publics (International Transport Union)
V2I	Vehicle to Infrastructure communication
WP	Work Package
WPL	Work Package Leader

Executive Summary

This deliverable will introduce the contribution of CERTH in Task 5.4 “Transport service optimisation” of the AVENUE project. The AVENUE service platform would require an intelligence able to run optimisations to send vehicles to the right place at the right time in order to answer to the user’s demand. The main objective of this task is to work on the following aspects assisting mobility providers:

1. Planning & scheduling of vehicles in the network based on forecasted demands and energy constraints.
2. Real-time automated dispatching of missions to vehicles servicing the network when demands arise.
3. Intelligent routing of vehicles taking into consideration current and forecasted traffic as well as weather.
4. Pooling (also known as ride-sharing) to combine similar requests into one, maximizing the capacity of the service while still guaranteeing excellent transit times to travelers.
5. Automatic electrical energy management by evenly spreading the usage of the fleet and sending to charging stations vehicles when needed.
6. Health monitoring, defect and maintenance management using state-of-the-art machine learning techniques to predict and anticipate issues by scheduling maintenance. The overall reliability of the system is improved by ensuring the availability of the right number of vehicles.

The target is to optimise the use of autonomous vehicles, augment the service quality, and reduce the operation costs.

CERTH’s contribution refers to a theoretical study and algorithms’ development for traffic flow prediction and predictive maintenance tasks, crucial to achieving the aforementioned goals. By having a better view of road congestion and AV’s battery life state the overall AV’s transport service is optimised.
The results of this deliverable will be used by the Fleet Orchestrators to modify their algorithms and improve the routing optimization.

1 Introduction

AVENUE aims to design and carry out full-scale demonstrations of urban transport automation by deploying, for the first time worldwide, fleets of Automated minibuses in low to medium demand areas of 4 European demonstrator cities (Geneva, Lyon, Copenhagen, and Luxembourg) and 2 to 3 replicator cities. The AVENUE vision for future public transport in urban and suburban areas, is that Automated vehicles will ensure safe, rapid, economic, sustainable, and personalised transport of passengers. AVENUE introduces disruptive public transportation paradigms based on demand, door-to-door services, aiming to set up a new model of public transportation, by revisiting the offered public transportation services, and aiming to suppress prescheduled fixed bus itineraries.

Vehicle services that substantially enhance the passenger experience as well as the overall quality and value of the service will be introduced, also targeting elderly people, people with disabilities and vulnerable users. Road behaviour, security of the Automated vehicles and passengers' safety are central points of the AVENUE project.

At the end of the AVENUE project four-year period the mission is to have demonstrated that Automated vehicles will become the future solution for public transport. The AVENUE project will demonstrate the economic, environmental, and social potential of Automated vehicles for both companies and public commuters while assessing the vehicle road behaviour safety.

1.1 On-demand Mobility

Public transportation is a key element of a region's economic development and the quality of life of its citizens.

Governments around the world are defining strategies for the development of efficient public transport based on different criteria of importance to their regions, such as topography, citizens' needs, social and economic barriers, environmental concerns, and historical development. However, new technologies, modes of transport and services are appearing, which seem very promising to the support of regional strategies for the development of public transport.

On-demand transport is a public transport service that only works when a reservation has been recorded and will be a relevant solution where the demand for transport is diffuse and regular transport is inefficient.

On-demand transport differs from other public transport services in that vehicles do not follow a fixed route and do not use a predefined timetable. Unlike taxis, on-demand public transport is usually also not individual. An operator or an automated system takes care of the booking, planning and organization.

It is recognized that the use and integration of on-demand Automated vehicles has the potential to significantly improve services and provide solutions to many of the problems encountered today in the development of sustainable and efficient public transport.

1.2 Fully Automated Vehicles

A self-driving car, referred in the AVENUE project as a **Fully Automated Vehicle** (**AV**), or as Autonomous Vehicle, is a vehicle that can sense its environment and moving safely with no human input.

The terms *automated vehicles* and *autonomous vehicles* are often used together. The Regulation 2019/2144 of the European Parliament and of the Council of 27 November 2019 on type-approval

requirements for motor vehicles defines "automated vehicle" and "fully automated vehicle" based on their autonomous capacity:

An "automated vehicle" means a motor vehicle designed and constructed to move autonomously for certain periods of time without continuous driver supervision but in respect of which driver intervention is still expected or required.

"Fully automated vehicle" means a motor vehicle that has been designed and constructed to move autonomously without any driver supervision.

In AVENUE we operate ***Fully Automated minibuses for public transport***, (previously referred as Autonomous shuttles, or Autonomous buses), and we refer to them as simply *Automated minibuses* or *the AVENUE minibuses.*

In relation to the SAE levels, the AVENUE project will operate SAE Level 4 vehicles.

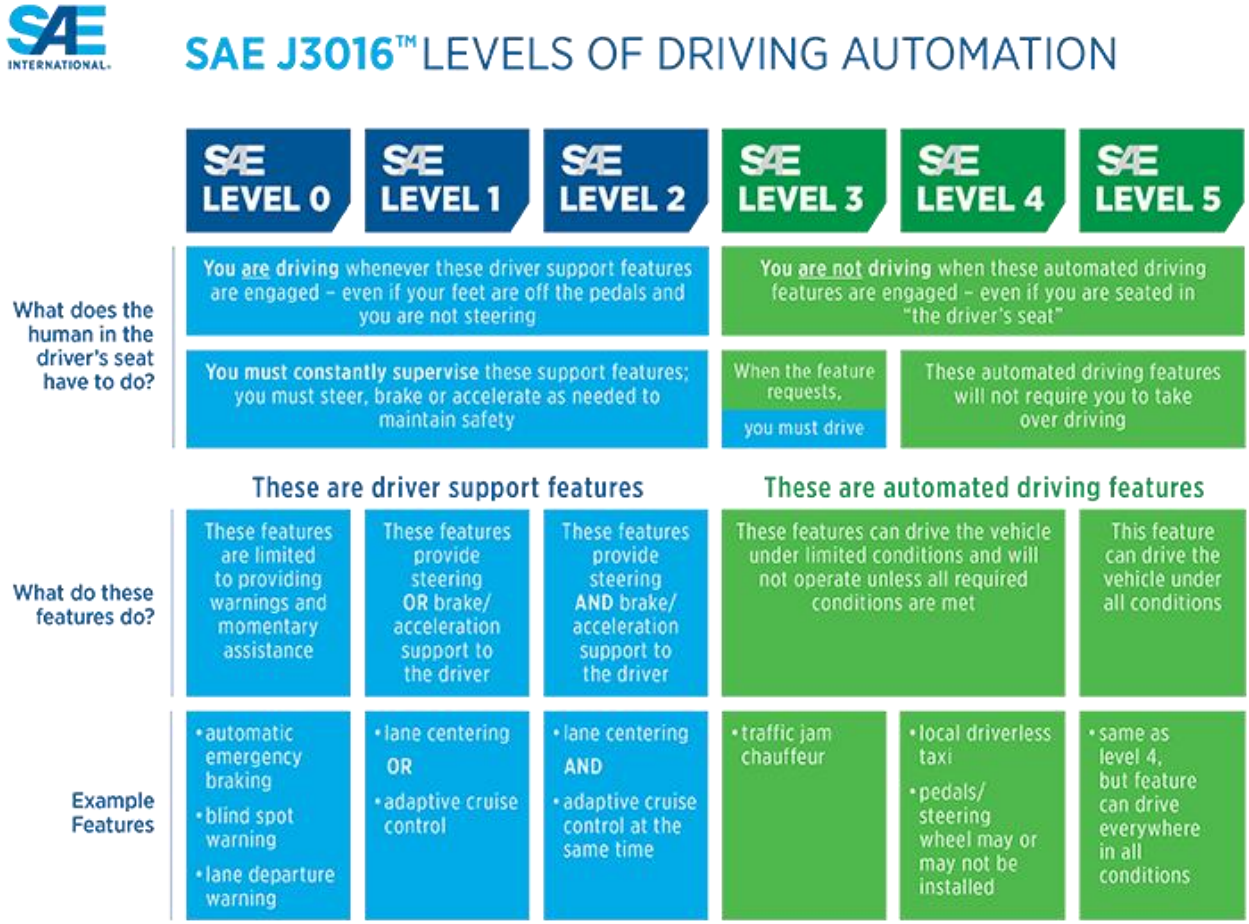

Table 1: SAE Driving Automation levels (©2020 SAE International)

1.2.1 Automated vehicle operation overview

We distinguish in AVENUE two levels of control of the AV: micro-navigation and macro-navigation. Micro navigation is fully integrated in the vehicle and implements the road behaviour of the vehicle, while macro-navigation is controlled by the operator running the vehicle and defines the destination and path of the vehicle, as defined the higher view of the overall fleet management.

For micro-navigation Automated Vehicles combine a variety of sensors to perceive their surroundings, such as 3D video, LIDAR, sonar, GNSS, odometry and other types of sensors. Control software and systems, integrated in the vehicle, fusion and interpret the sensor information to identify the current position of the vehicle, detecting obstacles in the surround environment, and choosing the most

appropriate reaction of the vehicle, ranging from stopping to bypassing the obstacle, reducing its speed, making a turn etc.

For the Macro-navigation, that is the destination to reach, the Automated Vehicle receives the information from either the in-vehicle operator (in the current configuration with a fixed path route), or from the remote-control service via a dedicated 4/5G communication channel, for a fleet-managed operation. The fleet management system considers all available vehicles in the services area, the passenger request, the operator policies, the street conditions (closed streets) and send route and stop information to the vehicle (route to follow and destination to reach).

1.2.2 Automated vehicle capabilities in AVENUE

The Automated vehicles employed in AVENUE fully and automatically manage the above defined, micro-navigation and road behaviour, in an open street environment. The vehicles are automatically capable to recognise obstacles (and identify some of them), identify moving and stationary objects, and automatically decide to bypass or wait behind them, based on the defined policies. For example, with small changes in its route the AVENUE minibus is able to bypass a parked car, while it will slow down and follow behind a slowly moving car. The AVENUE mini-buses are able to handle different complex road situations, like entering and exiting round-about in the presence of other fast running cars, stop in zebra crossings, communicate with infrastructure via V2I interfaces (ex. red light control).

The minibuses used in the AVENUE project technically can achieve speeds of more than 60Km/h. However, this speed cannot be used in the project demonstrators for several reasons, ranging from regulatory to safety. Under current regulations the maximum authorised speed is 25 or 30 Km/h (depending on the site). In the current demonstrators the speed does not exceed 23 Km/h, with an operational speed of 14 to 18 Km/h. Another, more important reason for limiting the vehicle speed is safety for passengers and pedestrians. Due to the fact that the current LIDAR has a range of 100m and the obstacle identification is done for objects no further than 40 meters, and considering that the vehicle must safely stop in case of an obstacle on the road (which will be “seen” at less than 40 meters distance) we cannot guarantee a safe braking if the speed is more than 25 Km/h. Note that technically the vehicle can make harsh break and stop with 40 meters in high speeds (40 -50 Km/h) but then the break would too harsh putting in risk the vehicle passengers. The project is working in finding an optimal point between passenger and pedestrian safety.

Due to legal requirements a **Safety Operator** must always be present in the vehicle, able to take control any moment. Additionally, at the control room, a **Supervisor** is present controlling the fleet operations. An **Intervention Team** is present in the deployment area ready to intervene in case of incident to any of the minibuses. Table 2 provides an overview of the AVENEU sites and ODDs.

	Summary of AVENUE operating sites demonstrators							
	TPG		Holo		Keolis	Sales-Lentz		
	Geneva		Copenhagen	Oslo	Lyon	Luxembourg		
Site	Meyrin	Belle-Idée	Nordhavn	Ormøya	ParcOL	Pfaffental	Contern	Esch sur Alzette
Funding	TPG	EU + TPG	EU + Holo	EU + Holo	EU + Keolis	EU + SLA	EU + SLA	EU + SLA
Start date of project	August 2017	May 2018	May 2017	August 2019	May 2017	June 2018	June 2018	February 2022
Start date of trial	July 2018	June 2020	September 2020	December 2019	November 2019	September 2018	September 2018	April 2022
Type of route	Fixed circular line	Area	Fixed circular line	Fixed circular line	Fixed circular line	Fixed circular line	Fixed circular line	Fixed circular line
Level of on-demand service*	Fixed route / Fixed stops	Flexible route / On-demand stops	Fixed route / Fixed stops	Fixed route / Fixed stops	Fixed route/Fixed stops	Fixed route / Fixed stops	Fixed route / Fixed stops	Fixed route / Fixed stops
Route length	2,1 km	38 hectares	1,3 km	1,6 km	1,3 km	1,2 km	2,3 km	1 km
Road environment	Open road	Semi-private	Open road	Open road	Open road	Public road	Public road	Main pedestrian road
Type of traffic	Mixed	Mixed	Mixed	Mixed	Mixed	Mixed	Mixed	Pedestrians, bicycles, delivery cars
Speed limit	30 km/h	30 km/h	30 km/h	30 km/h	8 to 10 km/h	30 km/h	50 km/h	20 km/h
Roundabouts	Yes	Yes	No	No	Yes	No	No	No
Traffic lights	No	No	No	No	Yes	Yes	Yes	No
Type of service	Fixed line	On demand	Fixed line	Fixed line	Fixed line	Fixed line	Fixed line	On Demand
Concession	Line (circular)	Area	Line (circular)	Line (circular)	Line (circular)	Line (circular)	Line (circular)	Line (circular)
Number of stops	4	> 35	6	6	2	4	2	3
Type of bus stop	Fixed	Fixed	Fixed	Fixed	Fixed	Fixed	Fixed	Fixed
Bus stop infrastructure	Yes	Sometimes, mostly not	Yes	Yes	Yes	Yes	Yes	Yes
Number of vehicles	1	3-4	1	2	2	2	1	1
Timetable	Fixed	On demand	Fixed	Fixed	Fixed	Fixed	Fixed	On-demand
Operation hours	Monday-Friday (5 days)	Sunday-Saturday (7 days)	Monday-Friday (5 days)	Monday-Sunday (7 days)	Monday-Saturday (6 days)	Tuesday & Thursday Saturday, Sunday & every public holiday	Monday - Friday	Monday – Saturday
Timeframe weekdays	06:30 – 08:30 / 16:00 – 18:15	07:00 – 19:00	10:00 – 18:00	7:30 – 21:30	08:30 – 19:30	12:00 – 20h00	7:00 – 9:00 16:00 – 19:00	11:00 – 18:00 11:00 – 18:00
Timeframe weekends	No service	07:00 – 19:00	No service	9:00 – 18:00	08:30 – 19:30	10:00 – 21:00	No Service	On Suterday only
Depot	400 meters distance	On site	800 meters distance	200 meters distance	On site	On site	On site	500 m distance
Driverless service	No	2021	No	No	No	No	No	No
Drive area type/ODD	B-Roads	Minor roads/parking	B-Roads/minor roads	B-Roads	B-Roads	B-Roads	B-Roads/parking	
Drive area geo/ODD	Straight lines/plane	Straight lines/ plane	Straight lines/ plane	Curves/slopes	Straight Lines/ plane	Straight lines/ plane	Straight lines/ plane	Straight lines / plane
Lane specification/ODD	Traffic lane	Traffic lane	Traffic lane	Traffic lane	Traffic lane	Traffic lane	Traffic lane	Open area
Drive area signs/ODD	Regulatory	Regulatory	Regulatory, Warning	Regulatory	Regulatory	Regulatory	Regulatory	Regulatory
Drive area surface/ODD	Standard surface, Speedbumps	Standard surface, Speedbumps	Standard surface Speedbumps, Roadworks	Frequent Ice, Snow	Standard surface, Potholes	Standard surface	Standard surface	Standard Surrface

Table 2: Summary of AVENUE operating site (+ODD components)

1.3 Preamble

Task 5.4 focuses on "Transport service optimisation." Its primary goal is to enhance autonomous vehicle (AV) service by addressing several key aspects: vehicle planning and scheduling based on forecasted demands and energy limits, real-time dispatching, intelligent routing considering traffic and weather forecasts, ride-sharing for efficient service, automated energy management, and predictive maintenance using advanced machine learning. These efforts aim to maximize AV efficiency, enhance service quality, and cut operational costs. CERTH's role includes theoretical studies and algorithm development for traffic flow prediction and predictive maintenance. The findings will guide Fleet Orchestrators in refining routing algorithms.

2 Traffic Flow prediction

In the last few years, the utilisation of autonomous vehicles (AVs) has been significantly increased over the globe. AVs have the potential to improve the quality and productivity of the time spent in cars, increase the safety and efficiency of the transportation system and transform transportation into a utility available to anyone, anytime. Traffic flow prediction is an important component of the autonomous driving system and is used to handle traffic congestion problem. It can assist to decide their itinerary and take adaptive decisions such as turn left or right, move straight, lane change, stop or accelerate with respect to their surrounding objects. In recent years, new models and frameworks for predicting traffic flow have been rapidly developed to enhance the performance of traffic flow prediction, alongside the implementation of Artificial Intelligence (AI) methods such as machine learning and deep learning.

In the next sections, state of the art approaches in traffic flow prediction tasks are presented. The problem formulation, the dataset analysis and the methodology used for the purpose of this task is also described in detail. The performance of the developed algorithms is estimated and compared in terms of various statistical metrics.

2.1 Literature Review

Traffic flow prediction has a pivotal role to play in Intelligent Transport Systems (ITS) and as a result has attracted much attention from the research community over the last few decades. Due to the increasing amount of vehicles and the development of the autonomous vehicles operations, apart from the problem of short-term traffic prediction that researchers have been struggling with, the need for long–term traffic forecasting has made its appearance and grows rapidly. In light of this, parameters such as accuracy, efficiency and robustness are essential problems for many ITS applications and as a result several modelling efforts have been made in the literature in order to solve them, since 1970s. Based on the forecasting horizon, traffic forecasting can be categorised as short-term forecasting and long-term forecasting. The first category refers to a forecast horizon about less or equal than an hour while the latter to more than one hour.

Concerning the short-time traffic flow prediction, a wide variety of techniques has been applied, depending upon the type of data that are available and the potential end use of the forecast. According to [1], the approaches used in short-term traffic forecast can be broadly classified into four categories: Naïve, parametric, non-parametric, and hybrid. The first category refers to models that provide simple estimate of traffic in the future, e.g., historic averages. The parametric approaches refer to the models that capture all its information about the data within its parameters. In other words, the prediction of a future data value from the current state of the model depends only on its parameters. On the other hand, a non-parametric model can capture more subtle aspects of the data. It allows more information to pass from the current set of data that is attached to the model at the current state, so to be able to predict any future data. The parameters are usually said to be infinite in dimensions and so can express the characteristics in the data much better than parametric models. As a result, this allows the model to have more degrees of freedom, be more flexible and in cases of multivariate settings, simpler [2]. Lastly, other short-term traffic models have implemented a hybrid of the above-mentioned approaches. In [1] numerous short traffic prediction models and states were applied and it was proved that there is no “best technique”. Thus, research work in recent years has focused on combining different state of the art techniques (parametric and non-parametric) [3] [4]. Research has shown that the prediction accuracy

of nonparametric methods and hybrid methods is superior to parametric methods. Admittedly, the traffic flow prediction problem is challenging mainly due to the complex spatial and temporal dependencies [5]. While the traffic time series demonstrate strong temporal dynamics (accidents, rush hours, weekdays and weekends traffic differences), at the same time sensors on the road contains spatial correlations. To sum up, another division to the traffic prediction techniques that has been made is based on whether they model the spatial correlation among different traffic time series or not [6].

Traffic Forecasting without Modeling Spatial Dependency

In more detail, the parametric models that have been proposed for traffic flow prediction are based on time series analysis. A time series is described as the set of observed data x, each one being recorded at a specific time t [7]. The goal is to determine a trend from the observed traffic flow data in order to predict future values. Most of them are based on the classic Box and Jenkins Auto-Regressive Integrated Moving Average (ARIMA) model and have seen a satisfactorily successful application to traffic prediction. ARIMA consists of three parts: The autoregression (AR) part which correspond to the dependent relationship between an observation and some number of historical observations, the Moving Average (MA) part which is used to model the dependency between an observation and residual errors from a moving average model applied to historical observations and Intergraded (I) part, which is used in order to make the time series stationary, using differencing of raw observations. Mohammed, first used ARIMA model to predict short-term freeway traffic flow in 1979. At the same year, Ahmed used ARIMA model to predict short-term traffic [8]. It has been shown that the proposed model ARIMA (0, 1, 3) produce more accurate results contrary to moving average and double-exponential smoothing methods in terms of MAE and MSE. Other time-series models include techniques such as nonlinear regression, averaging algorithm, seasonal ARIMA (SARIMA). Especially, in [9], the application of seasonal time series models to traffic flow forecasting is addressed for the first time, and SARIMA and Winters exponential smoothing models were developed and demonstrate the necessity of usage of seasonal time series. Another popular time series forecasting model is exponential smoothing [10] [11], which in recent years has been used mostly in combination with other techniques so as for the results to be even more accurate.

Another parametric technique, which still remains very popular among time-series models, is Kalman filtering (KF). Okutani and Stephanedes introduced KF theory into this field and the derived results indicated improved performances [3]. In addition, the KF has been studied by many authors considering a first order traffic model, as in [12], and in [7], in order to solve the problem of significant data requirements in time-series models, especially in cases that the sufficient flow data is unavailable. They are generally applied both to the stationary and the non-stationary stochastic environment and its major advantage is that it allows the selected state variable to be updated continuously. In other words, KF updates the prediction of state variables based on the observation in the previous step. As a result, it is only needed the storage of the previous estimated information, which makes the algorithm more computationally efficient than utilising all the previous estimated data in each step of the prediction process [13]. Moreover, it can export useful information from data observations that could be noisy or inaccurate. Hence, they estimate a process by estimating the process state at a given time and then obtaining feedback in the form of noisy measurements. Based on the KF theory, there have also been some modifications [14] and hybrid models [15]. Unfortunately, despite the frequently good traffic prediction accuracy KF method yields, traffic conditions are mostly unstable and this can lead to generate over-prediction or under-prediction results.

On contrary, Historical Average (HA) models address the traffic flow as a seasonal process and use the weighted average of previous seasons as the prediction. For instance, let the season be 1 week, then the prediction for this Monday is the averaged traffic speeds from the last six Mondays. As the historical average method does not depend on short-term data, its performance is invariant to the small increases in the forecasting horizon. Thus, its ability to respond to unanticipated events and incidents is low. Despite that, it is easy for implementation and it is a fast working model.

As already discussed, these models usually rely on the stationarity assumption and require high quality of data set so as for them to be accurate and fail to capture non-linear temporal dynamics of traffic flow [16]. Generally, parametric techniques are useful when the pattern of the observed data has a regular variation, which in this case is either sparse or even impossible as the traffic data is usually stochastic and unstable. Subsequently, due to the fact that the traffic flow is uncertain, nonlinear and complex, it is difficult to predict the traffic flow effectively and accurately by the prediction method based on traditional mathematics and physics models [10]. Nevertheless, some of these models still remain popular. Compared to this, non-parametric algorithms consist of flexible number of parameters while the data are not assumed to follow any particular distribution. Therefore, it turns out that these models appear to be more suitable to illustrate traffic information. As for the accuracy, due to the learning ability and strong generalisation, nonparametric techniques are able to archive better performance. In summary, the mainly advantages of non-parametric algorithms include intuitive formulation, totally data-driven and thus free of assumptions on data distribution, high flexibility and easy extendibility [2].

All these reasons, in combination with the advent and rapid growth of Artificial Intelligence (AI), have led in the last few years to the statistical traffic prediction methods displacement to the AI approach. Specifically, a variety of Machine Learning (ML) algorithms have been used since 1990 to model traffic patterns, such as Neural Networks (NN), K-nearest neighbours' regression (k-NN) [11], Support Vector Regression (SVR) [17], Long Short-term Memory networks (LSTM) [18], and Gated Recurrent Units (GRU) [19].

As a typical nonparametric method, the k-NN model has received considerable attention. Many scholars have successfully applied the traditional k-NN model to short term traffic prediction and along with SVR, turns out to be the most common methods used by researchers for traffic flow forecasting. The use of k-NN method in time series forecasting was suggested for the first time in 1987, by Yakowit [20] and in 1991 Davis and Nihan [21] used the k-NN approach in traffic forecasting, which performed comparably to, but not better than, the linear time-series approach. Since then, a lot of research has been conducted regarding k-NN approaches in traffic flow forecasting and in combination with the use of larger data bases have led to the amelioration of k-NN's accuracy. In order to forecast short term traffic flows, in [22], the k-NN nonparametric regression has been applied and indicated that forecasting intervals calculated by k has an obvious improvement in comparison with NNs performance, in unconventional road condition forecasting. According to [23], nonparametric regression aims at finding past events that had input values identical to the current state of the system, namely at the moment that prediction is performed. The k-NN method is a non-parametric regression method that searches for the k optimal nearest neighbour and predicts traffic flow at the next time. In other words, it predicts the traffic flow y(p) for a given x(p), while using series of observation of input and output pairs ([x(t), y(t)], t=1, 2, 3..., n), that have been collected from historical data. In order for this to succeed, k-NN method sorts the past input measurements in the training sample from the additional input measurement according to their distance from the given x(p). The main advantage of the k-NN algorithm is the ability of adding data

from multiple locations into the examined area. Notably, nonparametric algorithms are theoretically grounded. As an asymptotically optimal forecaster, when applied to a state space with m members, k-NN approach will asymptotically be at least comparable to any m_{th} order parametric model [22]. Motivated by this attractive property, there is a steady stream of refining and extending k-NN in the literature. Basically, k-NN algorithms are single-step [21] [22] [23] and that leads to two crucial problems: firstly, for multiple-step forecasting it is noticed that the algorithm generates overlapping nearest neighbours and secondly, its performance is sometimes sensitive due to noisy neighbours [24]. Another disadvantage of the k-NN algorithm is the inability to perform spatial and temporal dependencies at the same time.

SVR is another remarkable nonparametric method that has been widely used for traffic flow forecasting. SVR is an adaptation of the SVM algorithm used for regression problems. The purpose of SVR is to map given data to a high dimensional feature space followed by performing linear regression with the same space. Firstly, each item in the dataset is plotted as a point in n-dimensional feature space. Then, classification is performed by locating hyperplane that divides the given input into classes. In literature, SVR has been successfully used to predict traffic parameters such as hourly flow and travel time [8] [6]. In particular, in [25] Neto et al. applied a supervised online SVR approach to investigate the accuracy of traffic flow prediction under both usual as well as unusual traffic conditions. Moreover, in [6], Wu et al. presented the SVR for travel-time prediction and compared it with other baseline travel-time prediction methods using real highway traffic data. Another spectacular work was proposed in [17], in which a novel prediction model was presented, called online learning weighted SVR. In comparison with several well-known prediction models including artificial neural network models, locally weighted regression, conventional SVR, and online learning SVR, it has shown superior performance to that of existing models.

Moreover, Bayesian networks have been proposed as models that could provide information from other road link in order to help the traffic flow forecasting at the examined link. Bayesian forecasting is a learning process that sequentially reviews the state of the travel time a priori knowledge based on new available data. In a few words, it is a directed graphic model for representing conditional dependences between a set of random variables [26]. As a non-parametric model, it is able to handle non-linear and non-stationary processes. Nevertheless, due to the difficulty of describing the influence of traffic flows at all the other links to the traffic flow at the examined one, since there would be too many variables to be determined in order to access this relationship, in [26], it is assumed the independence of the links with the examined. As a result, the calculations are simplified and an estimation of the joint probability distribution among all nodes is now feasible with accurate results, as the network is smaller. [27] used a scalar-based data model such as time series, and instead of using classical inference, the Bayesian Method was applied to estimate the parameters of a SARIMA model. The Markov Chain Monte Carlo Method was used to solve the posterior problem in high dimension. Within this method, it was calculated the posterior probability distributions of the Bayesian model, where ‘‘posterior’’ means after taking into account the relevant evidence related to the particular case under examination. Their study showed that the Bayesian inference of SARIMA model provides a more rational technique toward short-term traffic flow prediction compared to the commonly applied classical inference. Thus, forecasts from the Bayesian approach can better model the traffic behavior in reality with rapid fluctuations and extreme peaks.

Among the nonparametric techniques used for traffic flow prediction, the NN approaches have been commonly used for the problem [28] [29] and is one of the most popular approaches as this technique has resulted in hundreds of publications. As a matter of fact, mathematical theorems have proved that a three-layer feed-forward NN, with sigmoidal units in the hidden layer, can approximate a given real-valued, continuous multi-variate function to any desired degree of accuracy [30] [31]. It was noted that for traffic prediction purposes, artificial neural networks (ANNs) can be understood as nonlinear regression models, although they are typically used in this context for clustering, classification, and feature extraction. ANNs provide functionalities such as self-learning, self-organization and pattern recognition. They can also perform non-linear approximation between input and output spaces and their parallel structure makes them capable of implementing on parallel computing. The idea of predicting traffic flow using ANNs was initially introduced by Hua and Faghri in 1994 [32]. Following that, Smith and Demetsky designed a NN model which was compared with traditional traffic prediction methods and indicated that during the peak traffic periods the NN structures succeed better performance than other traditional ML architectures [36]. At this point, it is noteworthy to mention the research of Ledoux [29] that proposes a cooperation based neural network traffic flow model, which aims at being integrated into a real time adaptive urban traffic control system. Firstly, a single ANN was used to model traffic patterns on a signalised link. Then, the information was exchanged between connected local NN to model traffic flow at a junction. Unfortunately, using NNs model individually may not acquire good generalisation capability for traffic flow prediction. For this reason, incorporating other intelligent methods has widely been investigated for better prediction results. Specifically, in [33] a hybrid NN model is presented, which uses a fuzzy rule-based system (FRBS) which combines prediction output from an online KF and NN. The results appear to be really optimistic as hybrid prediction responds better for increasing non-linear, uncertain and highly fluctuating nature of urban traffic flow. [34] is one of the many significant works concerning the traffic flow prediction problem, in which, in contrast to some previous works, a dynamic NN architecture was used. In addition, due to the object-oriented approach, it was possible to model complex networks with a mixture of learning rules and processing element interactions. On the other hand, Jiang et al. [35] developed a dynamic wavelet NN model for traffic flow forecasting for capturing the dynamics of the traffic flow and for pattern recognition with enhanced feature detection capability.

Although nonparametric models' regression forecasting seems to give more accurate results in comparison with parametric techniques, this is mostly observable for the cases where interactions between travellers and infrastructure are relatively constant. That is because SVR and k-NN models are founded on chaotic system theory. Chaotic systems are defined by state transitions that are deterministic and non-linear. As for ANNs, despite their advantages, such as their capability to work with multi-dimensional data, implementation flexibility, generalisation ability, and strong forecasting power [36], they have inherent deficiencies as well. For example, determining the architecture of network is a difficult issue. Also, weight adjustment using gradient descent-based error propagation algorithm often converges slowly. In that way, another non-parametric approach, deep learning (DL), has been found to be useful for traffic flow prediction having multidimensional characteristics. DL is a form of machine learning that can be viewed as a nested hierarchical model which includes traditional neural networks and compared to other ML techniques, can provide enhanced performance for predicting traffic flow. In other words, by exploiting the dependencies in the high-dimensional set of variables, the capture of sharp discontinuities in traffic flow that emerges in large-scale networks becomes possible.

At this point, it is worth mentioning the applicability of recurrent neural networks (RNNs), a type of neural network with self-connection, which is able to perform nonlinear auto-regression, and its variants for traffic flow prediction. Due to the dynamic nature of transportation systems, RNNs had been proposed a dozen years ago to forecast traffic flow conditions [37]. In [38], Elman gives prominence to the abilities of RNNs' to learn complex spatiotemporal patterns. Furthermore, it is clearly explained how the RNN manages to represent these spatiotemporal patterns in a very efficient distributed manner through its weights. The essence of the current analysis is that the neural network learns to interpret current inputs in the context of its previous internal states. The State Space Neural Network (SSNN) is considered as a variant of Elman NN, and has been applied to predict urban travel time [37] [39] [40]. Different from the Elman NN, the Time-Delay Neural Network (TDNN) feeds back the previous input values into the current input values, and thus can be considered as a nonlinear multivariate AR model [41]. In a previous work [42], it was identified that TDNN could achieve a higher travel time prediction accuracy compared with the SSNN. Nevertheless, traditional RNN fails for traffic prediction not only because this process requires both temporal-spatial interactions in the network but also due to the problem of vanishing gradient and exploding gradient. Moreover, traditional RNNs rely on the predetermined time lags to learn the temporal sequence processing, but it is difficult to find the optimal time window size in an automatic way [43]. To handle these issues, variants of RNN, such as LSTM and GRU, are widely used in predicting short-term traffic flow in the network. Depending on RNNs, several hybrid models were proposed.

LSTMs are designed to handle long-term dependencies. This feature is advantageous for traffic flow prediction, because of lack of previous knowledge on correspondence between the length of input past data and prediction results. LSTMs have the capability to acquire features with a long-time span for time series data. Ma et al highlighted the utility of LSTM NNs in traffic flow prediction methods as recently as in 2015 [43]. In this work, they indicate that LSTM NN can overcome the issue of back-propagated error decay through memory blocks, and thus exhibits the superior capability for time series prediction with long temporal dependency. Additionally, the comparison with different topologies of dynamic neural networks as well as other prevailing parametric and nonparametric algorithms implies that LSTM NN can achieve the best prediction performance in terms of both accuracy and stability. Subsequently, at the same year, LSTM RNN was proposed in order to overcome the issue of static and predefined input data that the already existed models required [44]. Their model utilizes the three multiplicative units in the memory block to determine the optimal time lags dynamically and achieve better performance regarding the accuracy in comparison with other models such as random walk, support vector machine, single layer feed forward neural network (FFNN) and stacked autoencoder. Despite the extensive variety of LSTM-based models, GRU models were widely applied in traffic prediction problem, as well. It was Fu et al. [19] who used GRU models for the first time, in the area of traffic flow prediction and showed that they achieve better results than LSTM RNN. Another typical example is the work of Li et al. [45], [50], in which they build a model based on LSTM and their experimental results that indicate the performances of models with GRU and LSTM are similar, and both of them better than the basic RNN. Alternatively, Huang, Bohan et al. [46] used the bidirectional RNN (BRNN) traffic prediction model to improve traffic forecasts and to have a better effect in comparison to the LSTM and GRU models. Their model achieved smaller MAE and RMSE and higher accuracy than LSTM or GRU model.

Traffic Forecasting with Modeling Spatial Dependency

Unfortunately, most of these ML-based methods are unable to capture in deep the correlation among different traffic conditions or other relevant traffic information. A representative characterisation of those spatial-temporal features is the key to successful traffic forecasting. Over the past few years, one of the most efficient deep neural networks to model the spatial dependencies is the convolutional neural network (CNN) [47], as it uses filters to find relationships between neighbouring inputs, which can make it easier for the network to converge on the correct solution. For long-term characterisation, LSTMs seem to be the most suitable algorithm to be used, as they are able to learn both short-term and long-term memory by enforcing constant error flow through the designed cell state [48]. In particular, to capture the spatial dependency of the traffic, recent studies [49] [50] [51], propose to model the transportation network as an image and use a CNN to extract spatial features while the historical data is viewed as an image. To take full advantage of spatial features, some researchers use CNN to capture adjacent relations among the traffic network, along with employing recurrent neural network (RNN) on time axis. In [51], a CNN-based method is proposed that learns traffic as images and predicts large-scale, network-wide traffic speed with a high accuracy. Spatiotemporal traffic dynamics are converted to images describing the time and space relations of traffic flow through a two-dimensional time-space matrix. A CNN is applied to the image following two consecutive steps: abstract traffic feature extraction and network-wide traffic speed prediction. The performance of the proposed method is finally compared with other naïve algorithms namely ordinary least squares, k-NN, ANN, random forest, stacked autoencoder, RNN, and LSTM networks. The results indicated an average of 42.91% accuracy improvement within an acceptable execution time.

[52] investigated a spatiotemporal Bayesian Network predictor. This approach incorporated all the spatial and temporal information available in a transportation network to carry out traffic flow forecasting. In a transportation network, there are usually many road segments related to or providing information about the traffic flow of the road segment under investigation. However, using all the related segments as input variables (nodes) would involve much irrelevance and redundancy, as well as being prohibitive computationally. To solve this problem, authors of [52] adopted the Pearson Correlation Coefficient to rank the input variables (traffic flows) for prediction, and the best-first strategy was employed to select only a subset as nodes of a Bayesian network.

In overall, the most popular solution is a combination of CNN and LSTM. In [49] Wu et al. based on the assets of LSTMs and CNNs networks, and with consideration of the spatial-temporal characteristics of traffic flow, proposed a novel short-term traffic flow prediction method based on the combination of CNN and LSTM (CLTFP). They developed a short-term traffic flow prediction method based on the combination of CNNs and LSTMs on an arterial road. In more detail, a one-dimension CNN was used to illustrate spatial features of the traffic flow and two LSTMs for its short-term variability and periodicities. Given those meaningful features, the feature-level fusion is performed to achieve short-term traffic flow forecasting. The proposed CLTFP was compared with other popular forecasting methods and the experimental results showed that the CLTFP has considerable advantages in traffic flow forecasting. Although this method could extract spatiotemporal correlations on a single arterial road, it failed to consider ramps, interchanges, and intersections, which are significant components of any transportation network [12]. Thus, it ignores the effect of congestion in terms of spatial-propagation. For instance, a traffic incident that occurs on one link may influence the traffic conditions in far-side regions. In order to solve these drawbacks, [53] proposes a novel NN structure that combines deep 2D CNNs and deep LSTMs to obtain the spatiotemporal correlations among all links in a traffic network. Specifically, they manage the traffic network as a visual process, where every frame represents a traffic state and several

future frameworks can efficiently use image-processing algorithms. The numerical experiments demonstrate that the proposed model outperformed algorithms such as LSTMs, DCNNs, SAEs and SVM method, in terms of accuracy and stability.

Summarising, these early works had one common drawback, all of them had ignored the topology relations among the sensors, regardless they tried to model the spatial correlation, as the spatial structure is in the Euclidean space (e.g., 2D images). For instance, two roads in different directions of a highway, though close in Euclidean distance, can have significantly different traffic pattern because of the network topology. Defferrard et al. [54] studied graph convolution, but only for undirected graphs. In order to solve these problems, Li et al. [5], model the traffic flow as a diffusion process on a directed graph and introduce Diffusion Convolutional Recurrent Neural Network (DCRNN), a deep learning framework for traffic forecasting that incorporates both spatial and temporal dependency in the traffic flow. In particular, DCRNN captures the spatial dependency using bidirectional random walks on the graph and the temporal dependency using the encoder-decoder architecture with scheduled sampling. Later, in [55], the authors speed up this model by replacing RNN with CNN to model the temporal dependency.

Conclusions

Overall, it is quite simpleminded to claim that one method is clearly superior over other methods in any situation. One reason for this is that the proposed models that have been discussed above are developed with a small amount of separate specific traffic data, and the accuracy of traffic flow prediction methods is dependent on the traffic flow features embedded in the collected spatiotemporal traffic data [56]. In addition, traffic flow is influenced by many factors like weather, the day of the week, random events, road construction, lighting conditions, etc. Consequently, integration of external environmental factors is also crucial to decrease the error of prediction. Traffic flows are non-linear, mostly non-stationary processes influenced by many factors, as described earlier, while it also has significant spatio-temporal properties [57]. In general, literature shows promising results when using NNs, which have good prediction power and robustness. Although the deep architecture of NNs can learn more powerful models than shallow networks, existing NN-based methods for traffic flow prediction usually only have one hidden layer. It is hard to train a deep-layered hierarchical NN with a gradient-based training algorithm.

By and large, the most suitable prediction model strongly depends on the basic points that the work focuses, namely, in a microscopic view, the examination of the traffic flow at a specific point in space, or, in a macroscopic view, the correlations determination of the road segments. Considering the existed literature, it seems that the most promising models benefit from the spatio-temporal property of traffic flows, such as time–space matrix models or region-based models. Nevertheless, there is a need for a model that also works when the particles of the flow do not move in the same direction as vehicles. Another conclusion that could extracted from the literature is that the most noticeable models are the non-parametric ones, because they are able to handle non-linear, stationary or non-stationary, dynamic processes, and they can also exploit the spatio-temporal relationship of traffic flows. Nowadays, the most frequently uncounted models are variable neural networks such as LSTMs, CNN, or a combination of both.

2.2 Overview/Solution Description

The major goal of this work is implementing a theoretical approach on how an effective and accurate method that will predict real-time traffic flow could be implemented. This could be crucial for vehicle's optimised route choice. In more detail, as the vehicle is moving, for the route selection the next few minutes' traffic flow, or even the next hours, could be considered. In this way, an overview of the road condition is outlined and the possibility of getting the vehicle stuck in traffic jam will be decreased. Eventually, for this purpose, several machine and deep-learning models will be investigated, so at the end, the one with the most accurate, effective and fast performance will be chosen.

Generally, traffic flow prediction is an important component of traffic modeling, operation, and management.
Traffic flow prediction has a pivotal role to play in intelligent transport systems due to:

- The continuous increasing amount of vehicles
- The development of the autonomous vehicles' operations

By and large, traffic flow prediction is important for:

- Traffic Management
- Risk Assessment
- Public Safety

In our problem, accurate real-time traffic flow prediction can:

- provide information and guidance for the autonomous vehicles, to optimise the travel decisions in order to avoid traffic jam and reduce cost
- provide users the fastest route to their destination
- provide information to passengers about the traffic state on roads for the next few minutes, or even one hour later

Parameters must be examined:

- Traffic Flow at specific segments of the road (Vehicles / minutes of sample)
- Traffic Speed at specific segments of the road (Speed / minutes of sample)
- Weather Conditions
- The day of the week (weekdays vs. weekends)
- Lighting Conditions
- Accidents
- Road construction

Solution of the examined problem

As discussed, several possible models are proposed in literature. In Figure 1, the most fundamental techniques have been concentrated for an enhanced understanding.

No method is clearly superior over other methods in any situation
That is because:

- The traffic flow is influenced by factors such as the weather, the day of the week, random events, road construction, lighting conditions
- It also has significant spatio-temporal properties
- The amount of data consists also a crucial parameter for the model choice

Literature shows promising results when using NNs, which have good prediction power and robustness

- The most suitable prediction model strongly depends on the basic points that the work focuses:
- In a microscopic view, the examination of the traffic flow at a specific point in space
- In a macroscopic view, the correlations determination of the road segment

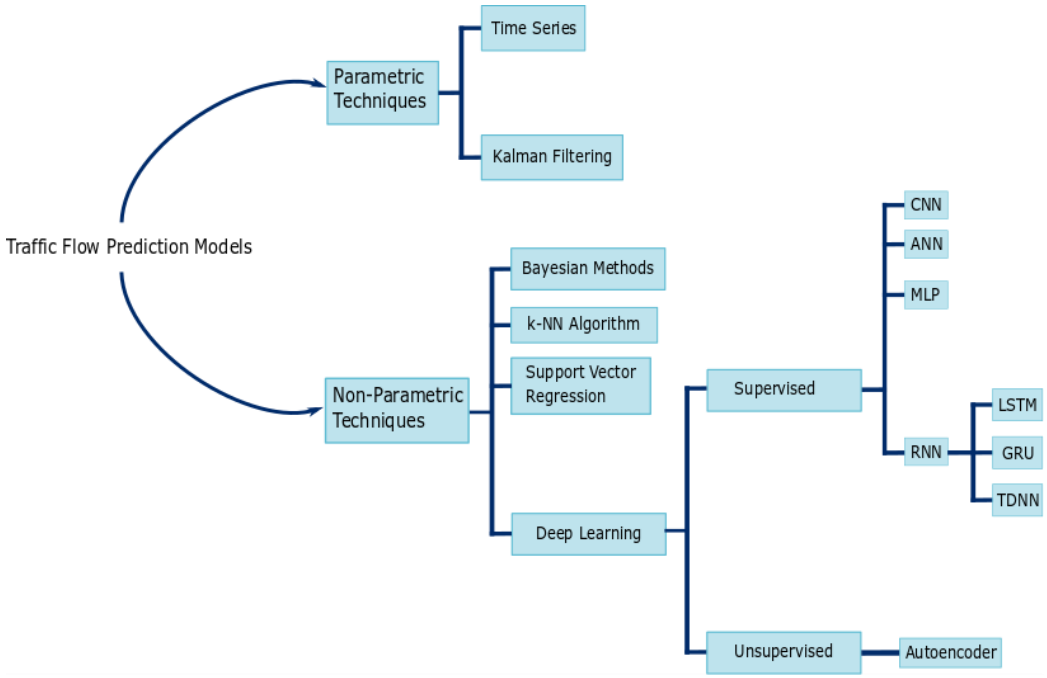

Figure 1: Traffic Flow Methods found in Literature.

More noticeable models are the non-parametric ones especially when there is a large amount of training data (they can handle non-linear, stationary or non-stationary, dynamic processes)

- For exploiting the spatio-temporal relationship of traffic flows, the most frequently uncounted models are variable neural networks such as LSTMs, CNN, or a combination of both
- Consequently, is it important to use several models, compare the prediction results with the ones produced by baseline models and conclude, via evaluating the results, to an optimum solution

2.3 Current Status and Progress

Initial investigation regarding the most suitable feature extraction methods and algorithms used.

- As there was no provided traffic flow data for the four cities the project is developed, namely Lyon, Copenhagen, Geneva and Luxembourg, we initially utilised two open datasets and extracted another one from an open platform from California State, Caltrans Performance

Measurement System (PeMS) in order to propose a methodology approach for the traffic flow prediction task

- The examined sensors were located at specific points of the road, counting the number of vehicles / 5min duration passing and the average speed

Three approaches were considered:

- Manual feature extraction from PeMS platform, suitable preprocessing and utilisation of baseline time series techniques as ARIMA and VAR models for training and prediction.
- Manual feature extraction from PeMS platform, training and prediction using Deep Learning algorithms, namely LSTM, GRU, CNN.
- A combination of a GCN and GRU model, in order to capture traffic forecasting's spatial and temporal dependences. The elementary idea is to use the historical n time series data as input and the graph convolution network in order to illustrate topological structure of urban road network to obtain the spatial feature. At a second time, the obtained time series with spatial features are input into the GRU model and the dynamic change is achieved by information transmission between the units, while capturing the temporal features. At the end the prediction is performed as the two models are suitable connected in a layer.

Finally, we compared the manual extracted data from PeMS platform with two already existed datasets, SZ-taxi dataset and Los-loop dataset, so as to have a more accurate evaluation of our models.

After the implementation of traffic-flow prediction models on public data, an overview of various algorithms' performance on such kind of tasks was retrieved. In order to adjust the aforementioned implementation on the current task, traffic flow data from the pilot sites that the AV operated were necessary. Nevertheless, the data collected from the AVs' routes only included information on the traffic speed of the particular AV at a specific timestamp. The average traffic speed of the road extracted from all the passing vehicles was not possible to be collected with the existing equipment. Magnetic sensors, infrared sensors, photoelectric sensors, Doppler and radar sensors, inductive loops and video camera systems that are installed on, in and above the roadway constitute such systems that can provide the required information [41]. As a result, there was no a realistic insight of the traffic state of the road from the available data.

An alternative approach was applied in order to implement a solution that meets the T5.4 requirements. In particular:

- Real-world data was collected from the pilot sites that the AV operated via TomTom API
- Weather historical data from the corresponding regions were also collected from the Visual Crossing API
- Different machine learning and deep learning approaches were applied and the one with the best performance was suggested as the optimum solution to provide an overview of the road that the AV is operating

2.4 Dataset Overview

In this Section the public datasets as well as the datasets retrieved from the relevant APIs are described in detail.

2.4.1 Public Dataset

The first dataset we used was extracted from Caltrans Performance Measurement System (PeMS). The freeway Performance Measurement System (PeMS) collects real time traffic data from sensors and generates performance measures of vehicle miles traveled, hours traveled, and travel time. This project is sponsored by the California Department of Transportation (Caltrans) and provides tools and reports for traffic planners, operators, and engineers.

The traffic data is collected in real-time from over 39,000 individual detectors. These sensors span the freeway system across all major metropolitan areas of the State of California. PeMS is also an Archived Data User Service (ADUS) that provides over ten years of data for historical analysis. It integrates a wide variety of information from Caltrans and other local agency systems including:

- Traffic Detectors
- Incidents
- Lane Closures
- Toll Tags
- Census Traffic Counts
- Vehicle Classification
- Weight-In-Motion
- Roadway Inventory

We estimate that similar measures could take place for the cities we are interested in (Lyon, Copenhagen, Geneva and Luxembourg).

The already extracted datasets we used in order to compare the performance of our models and confirmed that our models work properly and accurately, are SZ-taxi and Los-loop datasets. These two sets are related to traffic speed, in contrast with PeMS dataset, thus we decide to use data related to traffic flow. However, those two different features have the same structure, so they can both be used as traffic information without loss of generality.

2.4.1.1 Loading Data

The PeMS dataset is available at http://pems.dot.ca.gov/ . The traffic data is collected in real-time from over 39,000 individual detectors which were deployed across the major metropolitan areas of California state highway system. They were aggregated into 5-minute interval from 30-second data samples.

1. In our problem, we used a medium scale dataset as we randomly selected 241 sensors among District 6 of California. We selected two months for examination (01/04/19 – 31/05/19) and keep only the traffic flow information of weekdays. At the end, the used set was about 3.123.360 traffic flow data. Additionally, from the geographic coordinates of the sensors, the 241*241 adjacency matrix for the GCN model was calculated, by computing the driving distance among them. Each row represents one sensor and the values in the matrix represent the connectivity between the roads.

2. SZ-taxi. This dataset was the taxi trajectory of Shenzhen from Jan. 1 to Jan. 31, 2015. It was selected 156 major roads of Luohu District as the study area. The experimental data mainly

includes two parts. One is a 156*156 adjacency matrix, which describes the spatial relationship between roads. Each row represents one road and the values in the matrix represent the connectivity between the roads. Another one is a feature matrix, which describes the speed changes over time on each road. Each row represents one road; each column is the traffic speed on the roads in different time periods. This matrix aggregates the traffic speed on each road every 15 minutes, for every of 156 roads. This dataset could be found at: https://github.com/lehaifeng/T-GCN/tree/master/data .

3. Los-loop. This dataset was collected in the highway of Los Angeles County in real time by loop detectors. We selected 207 sensors and its traffic speed from Mar.1 to Mar.7, 2012. This matrix aggregates the traffic speed every 5 minutes, for every of 156 roads. Similarly, the data concludes an adjacency matrix and a feature matrix. The adjacency matrix is calculated by the distance between sensors in the traffic networks. Since the Los-loop dataset contained some missing data, the linear interpolation method was used to fill missing values. This dataset could be found at: https://github.com/lehaifeng/T-GCN/tree/master/data .

2.4.1.2 Public Dataset Preview

In the PeMS dataset we picked 241 sensors located in Fresno, a town in California State. In Figure 2 the traffic flow data of the first 8 sensors, which were selected are presented.
In Figure 3, we illustrate all the 241 sensors which were used and as it is shown, there might be a topological correlation between some sensors which could affect the prediction. The illustration was applied with the help of OSMnx package. This let us download spatial geometries and model, project, visualize, and analyze real-world street networks from OpenStreetMap's APIs. Also, through this package we can download and calculate the driving distance between specific locations on roads so as to calculate the adjacency matrix.
Additionally, in Figure 4 and Figure 5 the SZ-taxi and Los-loop dataset were plotted, accordingly. Those two datasets contain speed information, but the problem and the proposed solution remains the same. They were used in order to compare the results and conclude to the optimal solution.

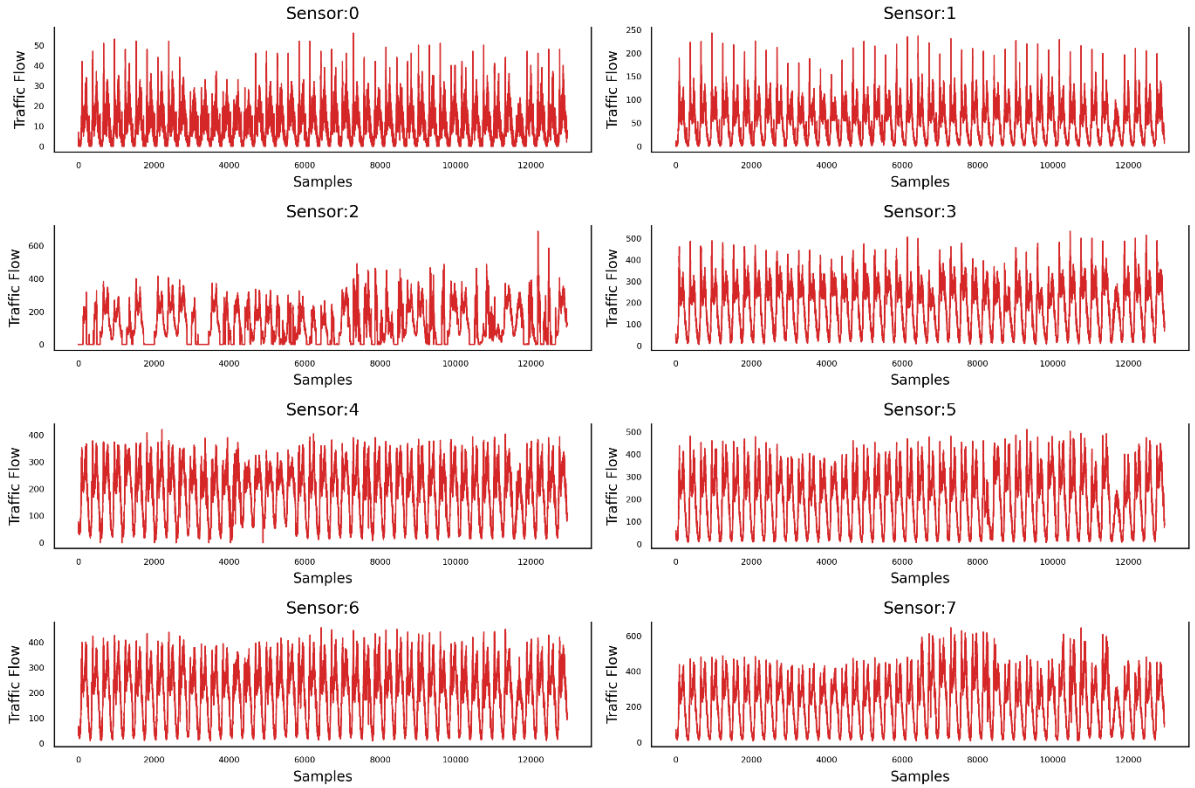

Figure 2: Traffic flow display of first 8 sensors in PeMS dataset vs number of samples

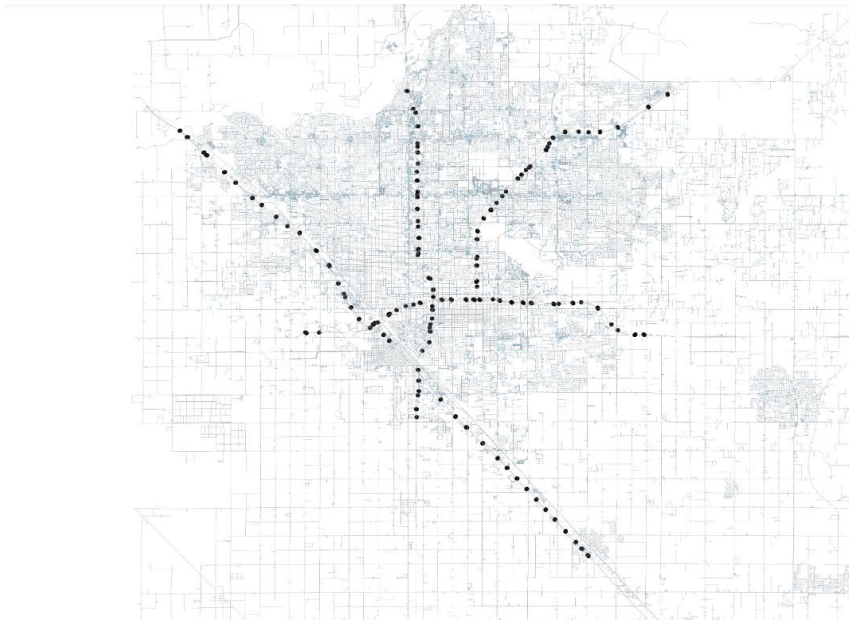

Figure 3: Display of the used sensors in Fresno of California State

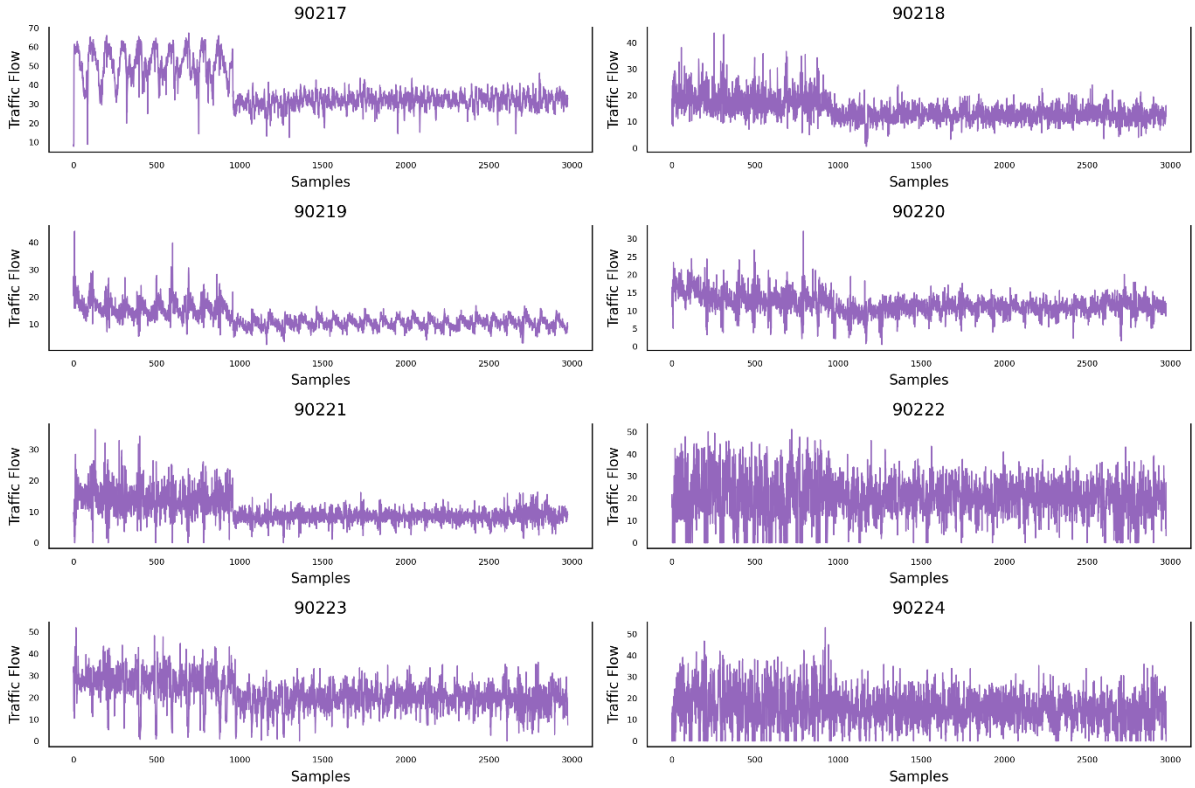

Figure 4: Plot of the first 8 sensors' speed measurements in SZ-Taxi dataset vs number of samples

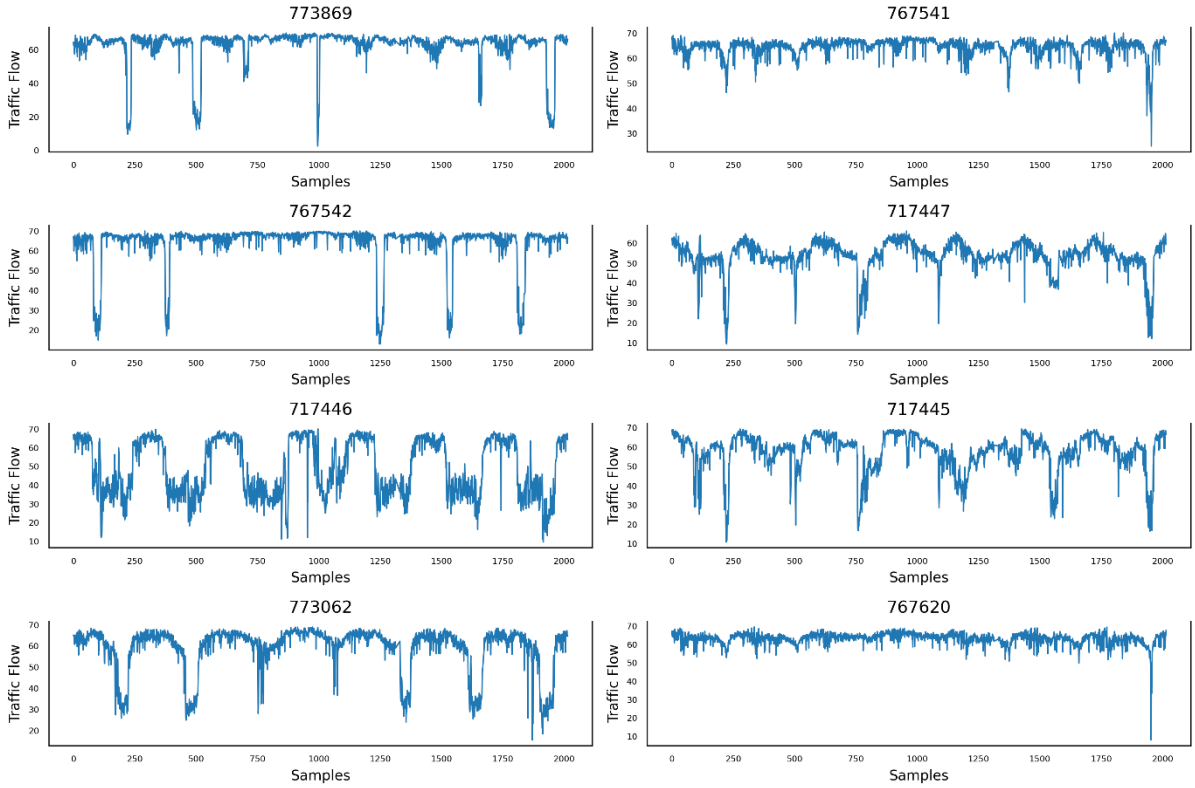

Figure 5: Plot of the first 8 sensors' speed measurements in Los-Loop dataset vs number of samples

2.4.2 TomTom API

The Traffic Flow service is a suite of web services that TomTom provides and designed for developers to create web and mobile applications around real-time traffic. In particular, this service provides information about the speeds and travel times of the road fragment closest to the given coordinates. The data can be retrieved through the relevant RESTful API which contains the following characteristics:

- Is updated every minute with the very latest traffic speed information
- Is based on the zooming level of different road categories that are displayed
- Provides traffic speed information for display on the map view with an option to use the **absolute** or **relative** speed information.
- Returns detailed information about traffic speed that will be analysed below. Details include:
  - **Current speed**
  - **Freeflow speed**
  - **A quality indicator**

In order to retrieve the traffic flow data, a user registration was required and a script was implemented to request data via https method. In particular, the url format should be as follows:

https://{baseURL}/traffic/services/{versionNumber}/flowSegmentData/{style}/{zoom}/{format}?key={Your_API_Key}&point={point}&unit={unit}&thickness={thickness}&openLr={boolean}&jsonp={jsonp}

The required parameters are described in detail in Table 3.

Table 3 : Request parameters for the TomTom Traffic Flow RESTful API

Required parameters	Description
baseURL *string*	The base URL for calling TomTom services. Values: api.tomtom.com: The default global API endpoint. kr-api.tomtom.com: The region-specific endpoint for South Korea.
versionNumber *string*	The version of the service to call. Value: The current value is 4.
style *string*	The style used with Raster Flow Tiles and Vector Flow Tiles. This has an effect on the coordinates in the response. Values: absolute relative relative0 relative0-dark relative-delay reduced-sensitivity
zoom *integer*	The zoom level. This has an effect on the following items: Traffic flow coordinates: There may be a slight deviation between the provided coordinates on different zoom levels. Visibility of the particular road: Roads of lower importance are only visible on zoom levels with a higher value.

	When Flow Segment data is used together with the Traffic Flow service, the zoom should be the same in both calls. Values: [0,22]
format *string*	The content type of the response structure. If the content type is jsonp, a callback method can be specified at the end of the service call. Values: xml json jsonp
key *string*	The authorization key for access to the API. Value: Your valid API Key.
point *float*	The coordinates of the point close to the road segment. They must be comma-separated and calculated using EPSG:4326 projection (also known as WGS84). Value: latitude, longitude

In particular, coordinates were given for the pilot sites of Slagelse, Luxembourg and Geneva which are illustrated in Figure 6, Figure 7 and Figure 8 respectively. The following parameters were inserted as input to the *https* request changing the location points according to the region, as shown in Table 4.

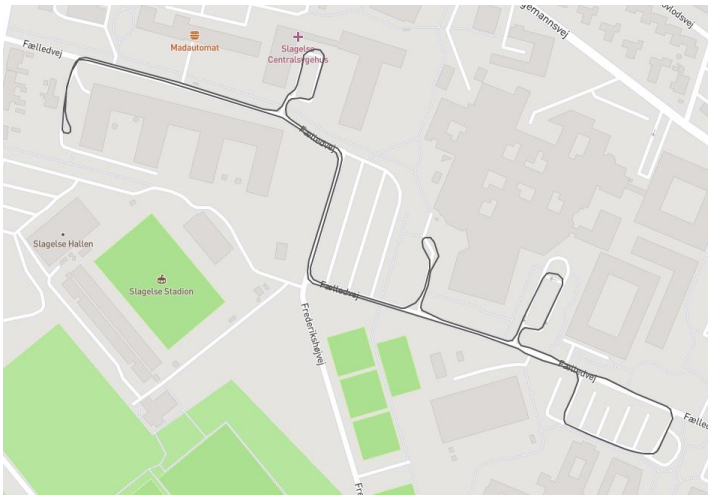

Figure 6: AV route at Slagelse pilot site.

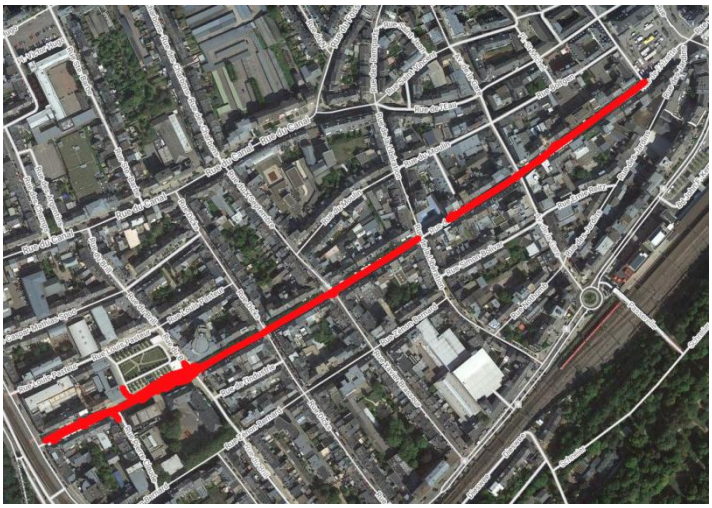

Figure 7: AV route at Luxembourg pilot site.

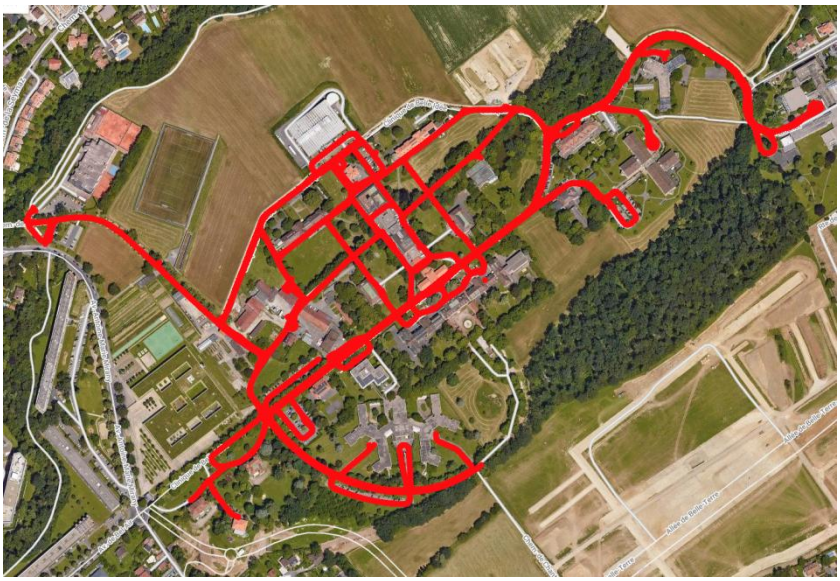

Figure 8: AV route at Geneva pilot site.

Table 4: Input values to the TomTom RESTful API for collecting traffic flow data from each pilot site

Required parameters	Input for Slagelse	Input for Luxembourg	Input for Geneva
baseURL	api.tomtom.com	api.tomtom.com	api.tomtom.com
versionNumber	4	4	4
style	absolute	absolute	absolute
zoom	20	20	20
format	json	json	json
key	API KEY	API KEY	API KEY
point	55.40035296, 11.36712993	49.49407992995702, 5.9813682472420915	46.206084, 6.207991

The Flow Segment Data API endpoint for a valid single request returns a response in XML or JSON format. The response fields with the relevant description are analysed in Table 5.

Table 5: Response fields from the Flow Segment Data API

Field	Description
<flowSegmentData> *object*	Main response element. The version attribute indicates the software version that generated the response.
<frc> *string*	**F**unctional **R**oad **C**lass. This indicates the road type: **FRC0:** Motorway, freeway or other major road **FRC1:** Major road, less important than a motorway **FRC2:** Other major road **FRC3:** Secondary road **FRC4:** Local connecting road **FRC5:** Local road of high importance **FRC6:** Local road
<currentSpeed> *integer*	The current average speed at the selected point, in the unit requested. This is calculated from the currentTravelTime and the length of the selected segment.
<freeFlowSpeed> *integer*	The free flow speed expected under ideal conditions, expressed in the unit requested. This is related to the freeFlowTravelTime
<currentTravelTime> *integer*	Current travel time in seconds based on fused real-time measurements between the defined locations in the specified direction
<freeFlowTravelTime> *integer*	The travel time in seconds which would be expected under ideal free flow conditions.
<confidence> *float*	The confidence is a measure of the quality of the provided travel time and speed. A value ranges between 0 and 1 where 1 means full confidence, meaning that the response contains the highest quality data. Lower values indicate the degree that the response may vary from the actual conditions on the road.
<coordinates> *object*	This includes the coordinates describing the shape of the segment. Coordinates are shifted from the road depending on the zoom level to support high quality visualization in every scale.
<openlr> *string*	The OpenLR code for segment
<roadClosure> *boolean*	This indicates if the road is closed to traffic or not.

For example, given the coordinates that represent the vehicle shown in red in Figure 9, the API's response on the current timestamp is as follows: The frc indicates that the vehicle's coordinates corresponds to a local road, the average speed at the current time equals to 20 km/h and the free flow speed is at 35 km/h. The confidence value is 1 and the returning coordinates are highlighted by the blue points in the picture.

Figure 9 : Illustration of the requested coordinates via TomTom RESTful API for the Slagelse street.

2.4.2.1 Traffic flow data preprocess

The final dataset collected for each pilot site is illustrated in Table 6. Results for the Slagelse site which was the one with the most operating time are presented. A similar procedure was followed for the sites at Luxembourg and Geneva.

Table 6: Duration of traffic flow data from each pilot site.

Pilot site	Duration of traffic flow data
Slagelse	5 months
Luxembourg	1 month
Geneva	1 month

In order to perceive a better insight of the road's traffic flow, the Traffic Performance Index (TPI) for each sample was calculated. TPI constitutes an indicator of traffic flow condition as it is a measure of congestion. The possible values range between 0 and 1 inclusive where 1 is a traffic jam state and 0 is a free flow state. A traffic jam state means that no vehicles can move, while a free flow state means that all vehicles travel at maximum speed with no influence from other vehicles. Its formula is as follows:

$$TPI = (V_{max} - V_i)/V_{max} \quad (1)$$

Where *Vmax* is the maximum speed of traffic data and *Vi* is the average link travel speed at a i-th time period.

Figure 10 illustrates a characteristic example of the TPI indicator versus speed on time axis. As it is shown, the TPI values represent a more realistic overview of the traffic state of the road.

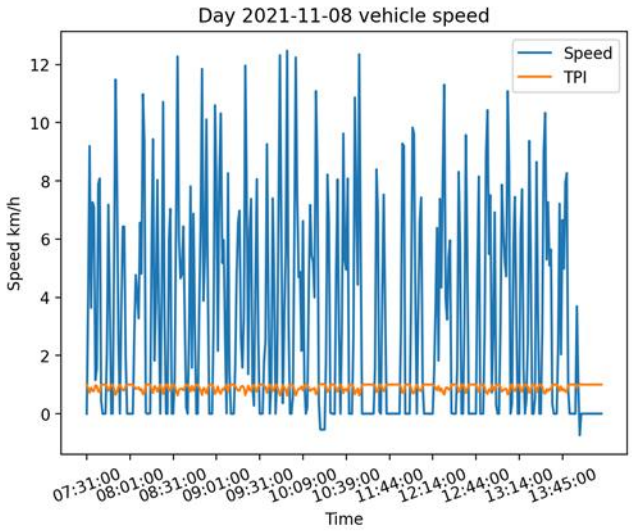

Figure 10: TPI representation versus traffic speed on time axis.

Data analysis procedures were applied on the collected data. Since the road where the AV operated was a rural road, there was no significant traffic through the day. However, as it is shown in Figure 11, it was observed that some days between 9am and 5pm the traffic appeared to be increasing. Therefore, the traffic flow prediction model for the Slagelse pilot site focused on the working hours.

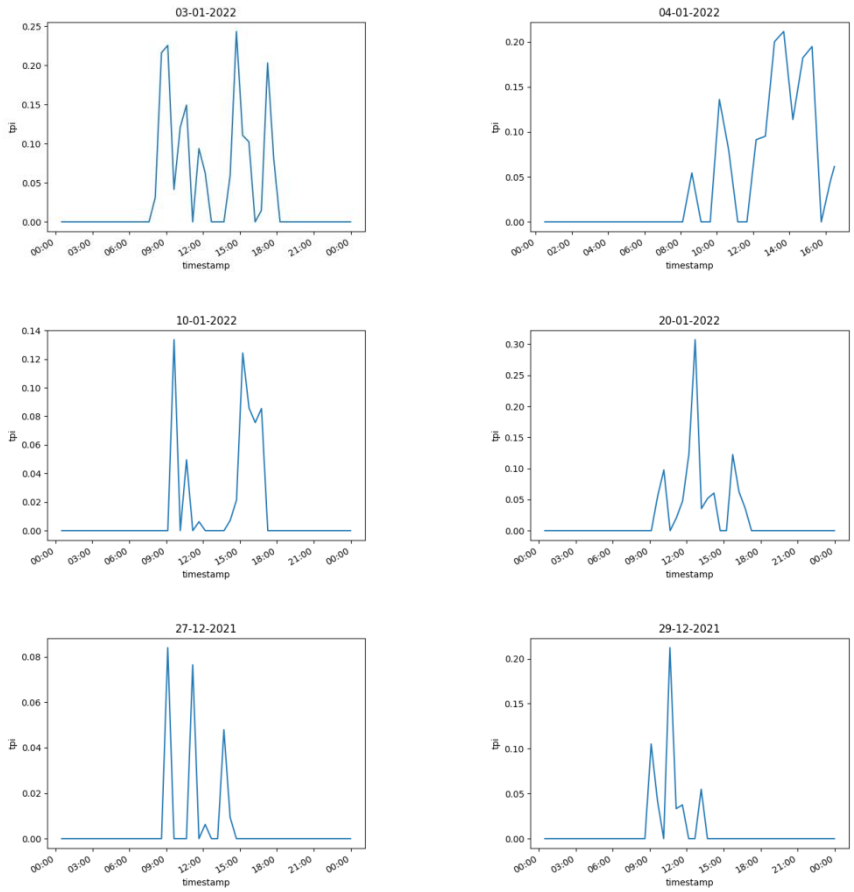

Figure 11 : Traffic flow representation of several days at Slagelse pilot site.

The traffic flow data was aggregated to half hour intervals and further processed to properly feed the prediction models. A necessary preprocessing step in forecasting tasks is the definition of the past timesteps and the forecast horizon parameters. This specifies how many observations the model will take into account and how many timesteps ahead it will make a speed prediction for. For the specific task, the past timesteps were set equal to one and a half hour and the forecast horizon to half an hour as it is illustrated in Figure 12.

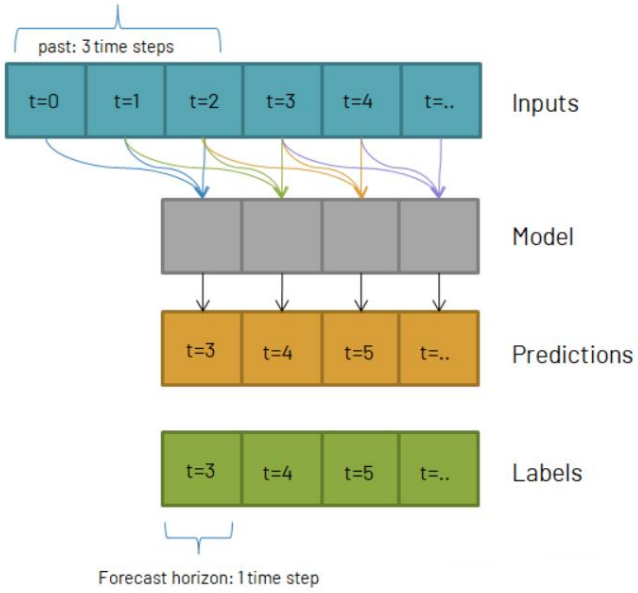

Figure 12: Definition of the past timestamps and forecast horizon parameters.

2.4.3 Visual Crossing API

Visual Crossing Weather is an open source service that provides access to weather forecast data, historical weather observation data and historical summary data. The data is accessible in three ways – directly in the browser, as a data download or as a RESTful API link. For the needs of the particular task historical weather data from Slagelse were also collected. The API provides sub-hourly, hourly and daily weather and climate data elements. In particular the response weather data elements are analysed in Table 7.

Table 7 : Response data elements of the Visual Crossing Weather API

Data element	Description
cloudcover	how much of the sky is covered in cloud ranging from 0-100%
conditions	textual representation of the weather conditions.
description	longer text descriptions suitable for displaying in weather displays. The descriptions combine the main features of the weather for the day such as precipitation or amount of cloud cover. Daily descriptions are provided for historical and forecast days. When the timeline request includes the model forecast period, a seven day outlook description is provided at the root response level.
datetime	ISO formatted date, time or datetime value indicating the date and time of the weather data in the local time zone of the requested location

datetimeEpoch	number of seconds since 1st January 1970 in UTC time
tzoffset	the time zone offset in hours. This will only occur in the data object if it is different from the global time zone offset.
dew	dew point temperature
feelslike	what the temperature feels like accounting for heat index or wind chill. Daily values are average values (mean) for the day.
feelslikemax (day only)	maximum feels like temperature at the location.
feelslikemin (day only)	minimum feels like temperature at the location.
hours	array of hourly weather data objects. This is a child of each of the daily weather object when hours are selected.
humidity	relative humidity in %
icon	a fixed, machine readable summary that can be used to display an icon
moonphase	represents the fractional portion through the current moon lunation cycle ranging from 0 (the new moon) to 0.5 (the full moon) and back to 1 (the next new moon)
normal	array of normal weather data values – Each weather data normal is an array of three values representing, in order, the minimum value over the statistical period, the mean value, and the maximum value over the statistical period.
offsetseconds (hourly only)	time zone offset for this weather data object in seconds – This value may change for a location based on daylight saving time observation.
precip	the amount of liquid precipitation that fell or is predicted to fall in the period. This includes the liquid-equivalent amount of any frozen precipitation such as a snow or ice.
precipcover (days only)	the proportion of hours where there was non-zero precipitation
precipprob (forecast only)	the likelihood of measurable precipitation ranging from 0% to 100%
preciptype	an array indicating the type(s) of precipitation expected or that occurred. Possible values include rain, snow, freezingrain and ice.
pressure	the sea level atmospheric or barometric pressure in millibars (or hectopascals)
snow	the amount of snow that fell or is predicted to fall
snowdepth	the depth of snow on the ground
source	the type of weather data used for this weather object. – Values include historical observation (“obs”), forecast (“fcst”), historical forecast (“histfcst”) or statistical forecast (“stats”). If multiple types are used in the same day, “comb” is used. Today a combination of historical observations and forecast data.
stations (historical only)	the weather stations used when collecting an historical observation record
sunrise (day only)	The formatted time of the sunrise (For example “2022-05-23T05:50:40”)
sunriseEpoch	sunrise time specified as number of seconds since 1st January 1970 in UTC time
sunset	The formatted time of the sunset (For example “2022-05-23T20:22:29”)
sunsetEpoch	sunset time specified as number of seconds since 1st January 1970 in UTC time
moonrise (day only,	The formatted time of the moonrise (For example “2022-05-23T02:38:10”)

optional)	
moonriseEpoch (day only, optional)	moonrise time specified as number of seconds since 1st January 1970 in UTC time
moonset (day only, optional)	The formatted time of the moonset (For example “2022-05-23T13:40:07”)
moonsetEpoch (day only, optional)	The formatted time of the moonset (For example “2022-05-23T13:40:07”)
temp	temperature at the location. Daily values are average values (mean) for the day.
tempmax (day only)	maximum temperature at the location.
tempmin (day only)	minimum temperature at the location.
uvindex	a value between 0 and 10 indicating the level of ultra violet (UV) exposure for that hour or day. 10 represents high level of exposure, and 0 represents no exposure. The UV index is calculated based on amount of short wave solar radiation which in turn is a level the cloudiness, type of cloud, time of day, time of year and location altitude. Daily values represent the maximum value of the hourly values.
visibility	distance at which distant objects are visible
winddir	direction from which the wind is blowing
windgust	instantaneous wind speed at a location – May be empty if it is not significantly higher than the wind speed.
windspeed	the sustained wind speed measured as the average windspeed that occurs during the preceding one to two minutes.
windspeedmax (day only, optional)	maximum wind speed over the day.
windspeedmean (day only , optional)	average (mean) wind speed over the day.
windspeedmin (day only , optional)	minimum wind speed over the day.
solarradiation	(W/m^2) the solar radiation power at the instantaneous moment of the observation (or forecast prediction)
solarenergy	MJ /m^2) indicates the total energy from the sun that builds up over an hour or day.
severerisk – (forecast only)	a value between 0 and 100 representing the likelihood of severe weather such as thunderstorms, hail or tornados. 0 is very low chance of severe weather. 30-60 represents there is a chance of severe weather, 60-100 indicates there is a high chance of severe weather.
degreedays (day only)	Optional elements indicating the number of degree days for this date. See the degree days API for more information on degree days. To turn degree days and degree day accumulation on, use the elements parameter.

A visual representation of the weather information that the Visual Crossing Weather API provides, is illustrated in Figure 13.

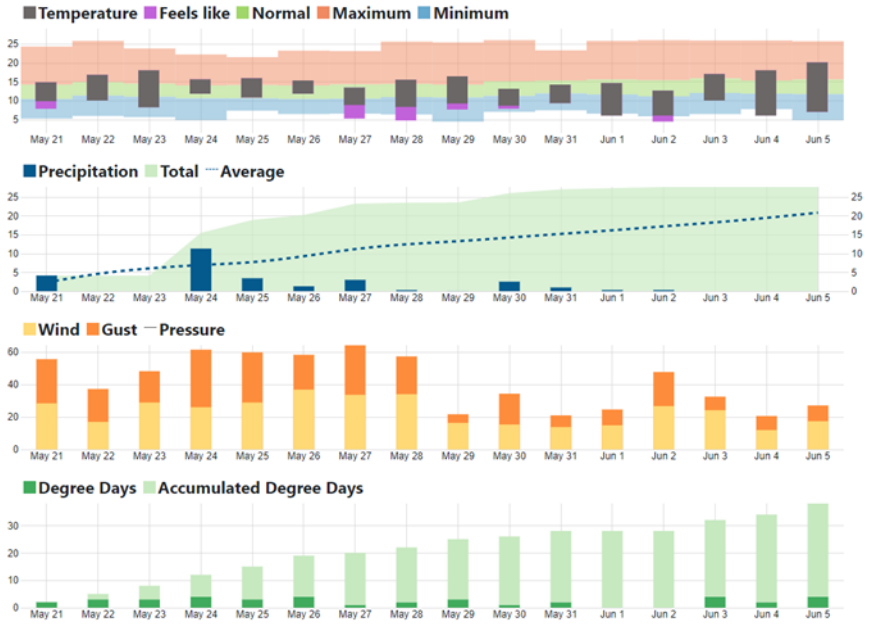

Figure 13 : Visual Representation of the weather data of Visual Crossing API.

2.5 Methodology

In this Section, the methodology followed on the public dataset as well as the real-world dataset that was collected from the APIs, is described in detail. The machine and deep learning algorithms that were applied, the parametrization and the results extracted in each case, are also illustrated and further discussed.

2.5.1 Methodology on the public dataset

In order to find the optimum solution in the traffic prediction problem, we applied several models and compared the results. At a second phase we modify them in order to improve their performance. In Figure 14, we summarize the procedure that was followed.

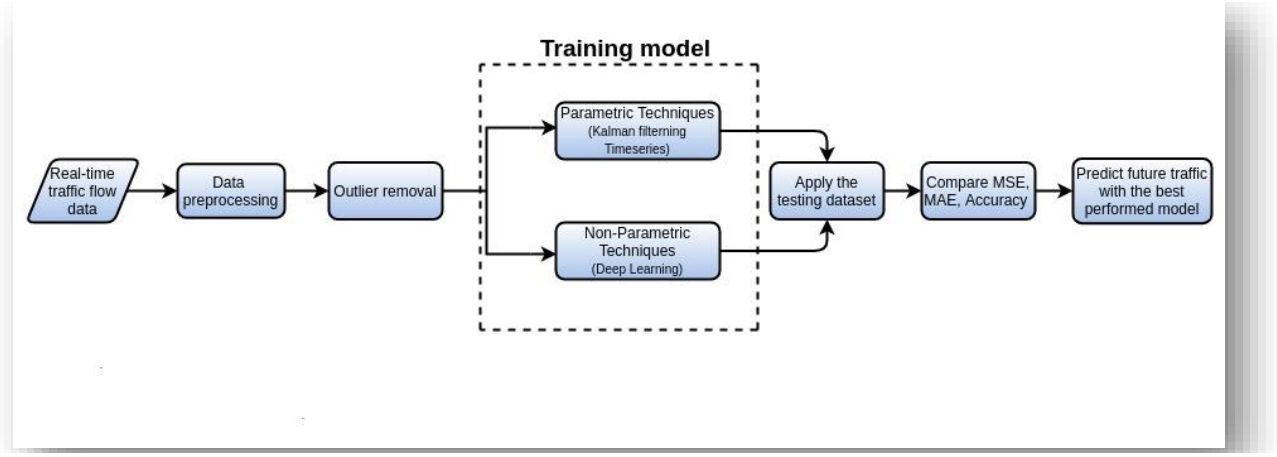

Figure 14 : Methodology followed on the public datasets.

As the real-time data set from PeMS platform was collected, the data preprocessing followed. In the experiments, the input data was normalized to the interval [0,1]. In addition, for every dataset, 80% of

the data was used as the training set and the remaining 20% was used as the testing set. We predicted the traffic speed of the next 5 minutes, 15 minutes, 30minutes, 45 minutes and 60 minutes. Since each time series may have a different traffic flow pattern and no pattern could fit all the traffic flow series, we create a unique model for each of the traffic flow series collected by a single sensor. This method applied to all the implemented techniques.

The training data was used to select the model features and set the values of the coefficients, and the primary testing data was used to evaluate the model performance. Multiple metrics were used to evaluate our predictive performance: Mean Squared Error (MSE), with units of cycles, and Mean Absolute Error (MAE), Accuracy, R^2, Var. In more detail, RMSE and MAE are used to measure the prediction error: the smaller the value is, the better the prediction effect is. Accuracy is used to detect the prediction precision: the lager the value is, the better the prediction effect is. $R^2$and Var calculate the correlation coefficient, which measures the ability of the prediction result to represent the actual data: the larger the value is, the better the prediction effect is.

Finally, we compare the performance of different baseline, deep learning and complex custom models such as ARIMA, VAR, LSTM, GRU, CNN and T-GCN, regarding the three datasets.

1. Mean Squared Error (RMSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_t - \hat{Y}_t\right)^2 \quad (2)$$

2. Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} Y_t - \hat{Y}_t \vee \quad (3)$$

3. Accuracy:

$$Accuracy = 1 - \frac{\left\|Y - \hat{Y}\right\|_F}{\|Y\|_F} \quad (4)$$

4. Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}\left(Y_t - \hat{Y}_t\right)^2}{\sum_{i=1}\left(Y_t - \acute{Y}_t\right)^2} \quad (5)$$

5. Explained Variance Score (Var):

$$var = 1 - \frac{Var\{Y - \hat{Y}\}}{Var\{Y\}} \quad (6)$$

2.5.1.1 ARIMA model

ARIMA(p,d,q) forecasting equation: ARIMA models are, in theory, the most general class of models for forecasting a time series which can be made to be “stationary” by differencing (if necessary), perhaps in conjunction with nonlinear transformations such as logging or deflating (if necessary).

The acronym ARIMA stands for Auto-Regressive Integrated Moving Average. Lags of the stationarized series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said

to be an "integrated" version of a stationary series. Random-walk and random-trend models, autoregressive models, and exponential smoothing models are all special cases of ARIMA models.

A non-seasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where:

- **p** is the number of autoregressive terms,
- **d** is the number of non-seasonal differences needed for stationarity, and
- **q** is the number of lagged forecast errors in the prediction equation.

AR(p) means **p** lagged error terms are going to be used in the ARIMA model. ARIMA relies on Autoregression, which is a process of regressing a variable on past values of itself. Autocorrelations gradually decay and estimate the degree to which white noise characterizes a series of data.

Integrated (d): If a trend exists then time series is considered non stationary and shows seasonality. Integrated is a property that reduces seasonality from a time series. ARIMA models have a degree of differencing which eliminates seasonality

Moving Average MA (q): Error terms of previous time points are used to predict current and future point's observation. Moving average (MA) removes non-determinism or random movements from a time series. The property **Q** represents Moving Average in ARIMA. It is expressed as MA(x) where x represents previous observations that are used to calculate current observation.
Moving average models have a fixed window and weights are relative to the time. This implies that the MA models are more responsive to current event and are more volatile.

The forecasting equation is constructed as follows. First, let y denote the d^{th} difference of Y, which means:

If d = 0: $y_t = Y_t$

If d = 1: $y_t = Y_t - Y_{t-1}$

If d = 2: $y_t = (Y|t - Y_{t-1}) - (Y|t - 1 - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

ARIMA(1,1,2) without constant = damped-trend linear exponential smoothing:

$$\hat{Y_t} = Y_{t-1} + \varphi_1(Y|t - 1 - Y_{t-2}) - \theta_1 e_{t-1} - \theta_1 e_{t-1} \quad (7)$$

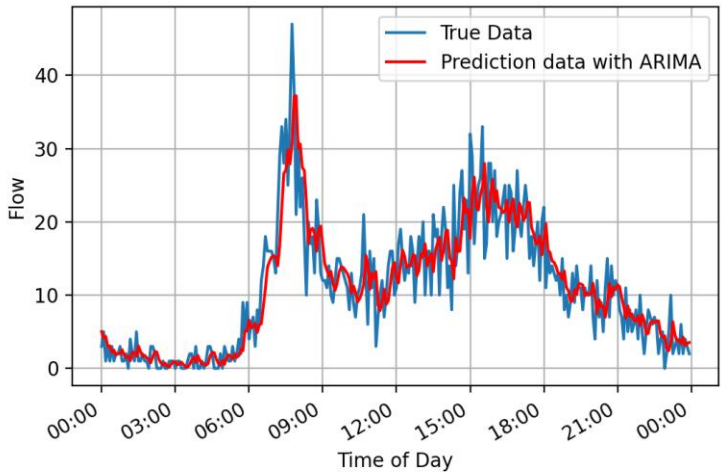

Figure 15 : Prediction results using ARIMA model on PeMS dataset

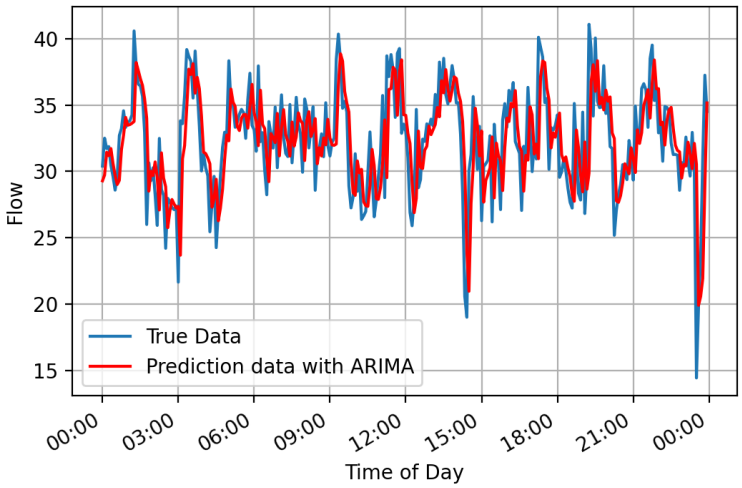

Figure 16: Prediction results using ARIMA model on SZ-Taxi dataset

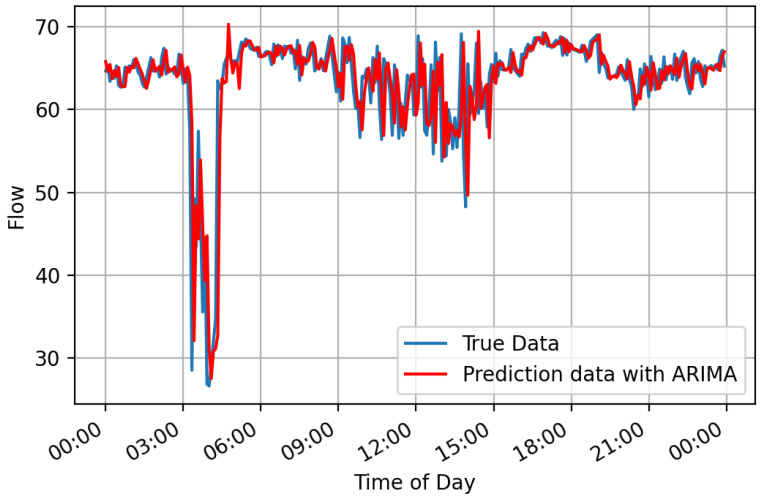

Figure 17: Prediction results using ARIMA model on Los-Loop dataset

Table 8 : Performance metrics using ARIMA model on public datasets

	MAE	MSE	Var	R^2	Accuracy
1^{rst} **Sensor/ PeMS-Data**	3.062	18.176	0.759	0.759	0.69261
1^{rst} **Sensor/SZ-Taxi**	2.454	21.78	0.8855	0.8855	0.9244
1^{rst} **Sensor/Los-Loop**	2.494	10.872	0.238	0.238	0.898

2.5.1.2 Vector Autoregressions (VAR) Model

The vector autoregression model (VAR), is arguably the simplest and most often used multivariate time series model for forecasting. It is a stochastic process model used to capture the linear interdependencies among multiple time series. VAR models generalize the univariate autoregressive model (AR model) by allowing for more than one evolving variable. All variables in a VAR enter the model in the same way: each variable has an equation explaining its evolution based on its own lagged values, the lagged values of the other model variables, and an error term. VAR modeling does not require as much knowledge about the forces influencing a variable as do structural models with simultaneous equations: The only prior knowledge required is a list of variables which can be hypothesized to affect each other intertemporally. Consider a first-order VAR, call it VAR(1):

$$R_t = \varphi_o + \Phi R_{t-1} + \varepsilon_t, Var(\varepsilon_t) = \Sigma \quad (8)$$

where R_t is a by 1 vector of variables.
The bivariate case is:

$$R_{1,t} = \varphi_{0,1} + \Phi_{11} R_{1,t-1} + \Phi_{12} R_{2,t-1} + \varepsilon_{1,t} \quad (9)$$
$$R_{2,t} = \varphi_{0,1} + \Phi_{21} R_{1,t-1} + \Phi_{22} R_{2,t-1} + \varepsilon_{2,t} \quad (10)$$

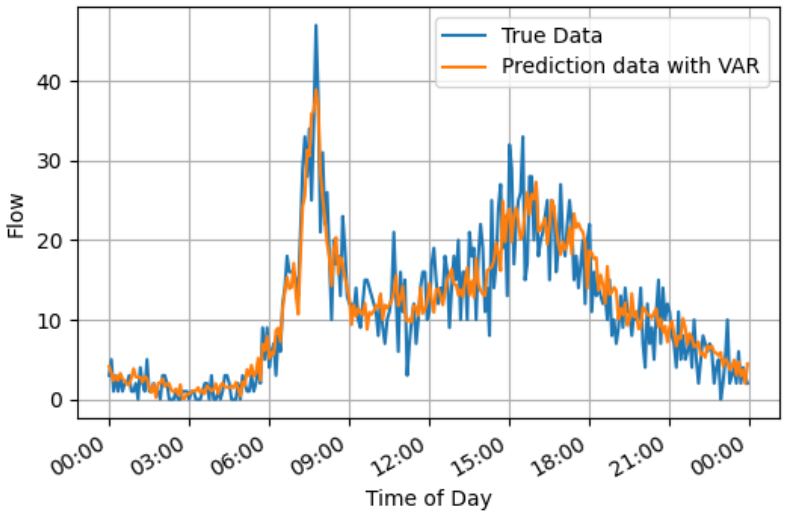

Figure 18 : Prediction results using VAR model on PeMS dataset

Table 9 : Performance metrics comparison using the ARIMA and VAR statistical models on the public datasets

Model	1^{rst} Sensor/ PeMS-Data		1^{rst} Sensor/SZ-Taxi		1^{rst} Sensor/Los-Loop	
	MAE	MSE	MAE	MSE	MAE	MSE
ARIMA	3.062	18.176	2.494	10.872	2.454	21.78
VAR	3.81	14.52	2.68	12.06	4.983	51.79

2.5.1.3 LSTM Model

LSTM has been widely used in many fields and achieved great success, such as in music generation, image caption, speech recognition and machine translation. LSTM improves the hidden-layer cell on the basis of RNN. The improvement of cell can make up for the gradient disappearance problem of RNN. LSTM adds some memory units, including forget gate, input gate and output gate. The memory units can further control the data and decide which should be retained and which should be deleted.

In traffic flow prediction, LSTM models have been used widely, as they achieve to skip the gradient descent problem. LSTM model was applied using Tensorflow 2.0 and Keras 2.3.1 Functional API as backend. Two different sequences of layers have been examined, which are illustrated in Figure 19 and Figure 20.
The following parameters were used for training the neural network models.

Table 10 : Parameters used for training LSTM

Parameter	Value
Optimizer	Adam
Learning Rate	0.0001
Batch Size	64
Loss Function	MSE
Gradient Norm Scaling	1
Early Stopping Patience	30

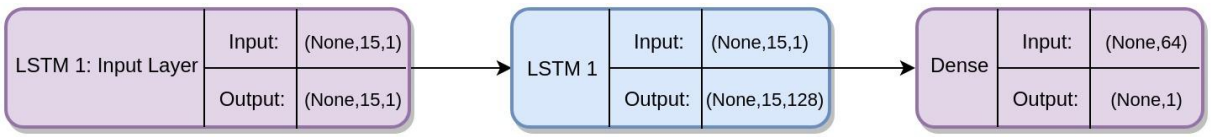

Figure 19 : Architecture of LSTM 1

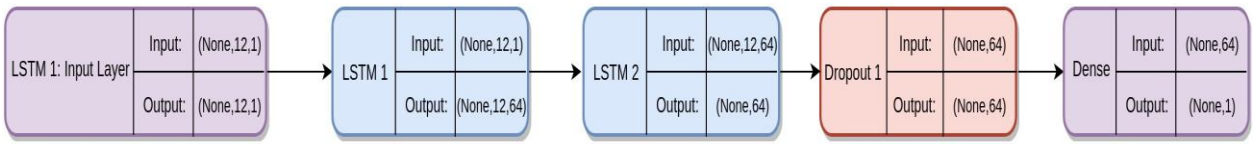

Figure 20: Architecture of LSTM 2

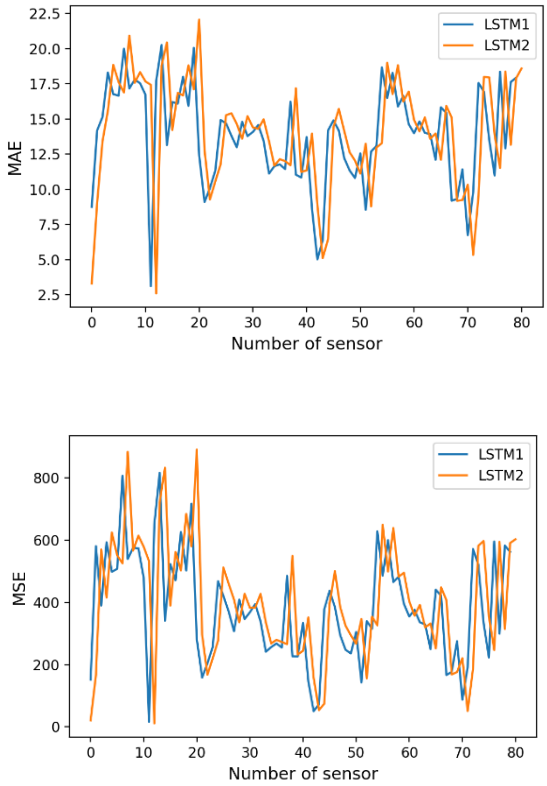

Figure 21 : Comparison of MAE and MSE for the different sequences of layers

2.5.1.4 GRU model

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun Cho et al. GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results. The GRU is like a long short-term memory (LSTM) with a forget gate but has fewer parameters than LSTM, as it lacks an output gate. GRU's performance on certain tasks of polyphonic music modeling, speech signal modeling and natural language processing was found to be similar to that of LSTM. GRUs have been shown to exhibit even better performance on certain smaller and less frequent datasets.

Table 11 : Performance metrics comparison using LSTM and GRU model on the public datasets

Model	**1^{rst} Sensor/ PeMS-Data**		**1^{rst} Sensor/SZ-Taxi**		**1^{rst} Sensor/Los-Loop**	
	MAE	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
LSTM	3.29	21	4.62	37.49	2.98	30.58
GRU	3.4	22.31	4.63	37.34	3.15	31.78

2.5.1.5 Convolutional Neural Networks

As it is already discussed, CNNs can be applied to time series forecasting. In order to do a prediction for the next 5 minutes traffic flow we used an one 1D CNN. A one-dimensional CNN is a CNN model that has a convolutional hidden layer that operates over a 1D sequence. This is followed by perhaps a second convolutional layer in some cases, such as very long input sequences, and then a pooling layer whose job is to distill the output of the convolutional layer to the most salient elements. The convolutional and pooling layers are followed by a dense fully connected layer that interprets the features extracted by the

convolutional part of the model. A flatten layer is used between the convolutional layers and the dense layer to reduce the feature maps to a single one-dimensional vector.

Table 12 : Performance metrics comparison using CNN, LSTM, GRU and VAR models on public datasets

Model	PEMS-DATA (SENSOR 0)	PEMS-DATA (SENSOR 25)	Los -speed(773869)	Sz -speed(90217)	Sz - speed(90224)
	MAE-MSE	MAE-MSE	MAE-MSE	MAE-MSE	MAE-MSE
CNN	3.08-18.34			2.42-10.23	5.53-49.5
LSTM	3.29-21	15.32-506.37	2.98-30.58	4.62-37.49	8.78- 117.05
GRU	3.4-22.31	15.53-517.22	3.15-31.78	4.63-37.34	
VAR	3.81-14.52	10.92-42.09		2.68-12.06	5.96-58.32

2.5.1.6 Graph Convolutional Neural Networks

As it is already discussed traffic flow data has spatial dependencies. A representative characterization of the spatial-temporal features is the key to successful traffic forecasting. A T-GCN model proposed in literature [58] was used in order to further improve the results of prediction. This model, consists of two parts: the graph convolutional network and the gated recurrent unit. As is shown in Figure 22 the historical n time series data performs as the input of the model and the graph convolution network is used to capture spatial structure of urban road network to obtain the spatial feature. Second, the obtained time series with spatial features are input into the gated recurrent unit model and the dynamic change is obtained by information transmission between the units, to capture temporal feature. Finally, we get results through the fully connected layer.

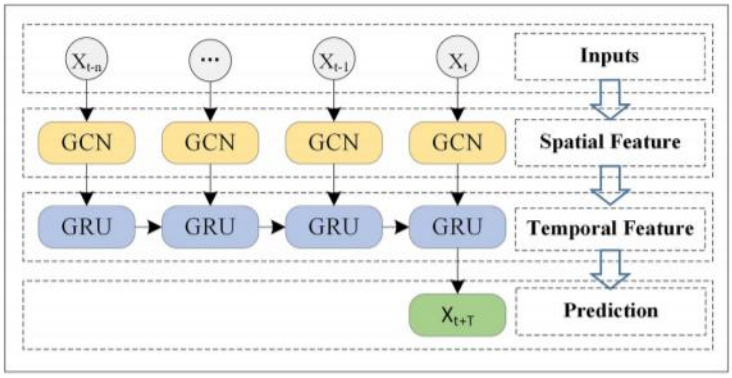

Figure 22 : Overview of taking the historical traffic information as input and obtain the finally prediction result through the Graph Convolution

To capture the spatial and temporal dependences from traffic data at the same time the overall T-GCN model was used, which is illustrated in Figure 23. The left side is the process of spatio-temporal traffic prediction, the right side shows the specific structure of a T-GCN cell, h_{t-1}denotes the output at time t-1, GC is graph convolution process, and u_t, $r_t$are update gate and reset gate at time t, while h_t denotes the output at time t. $f(A, X_t)$ represents the graph convolution process and is defined as:

$$f(X, A) = \sigma\left(\hat{A}Relu\left(\hat{A}XW_0\right)W_1\right)$$

where X represents the feature matrix, A represents the adjacency matrix $\hat{A} = \hat{D}^{\frac{-1}{2}}\hat{A}\hat{D}^{\frac{-1}{2}}$ denotes preprocessing step, $\hat{A} = A + I_N$ is a matrix with self-connection structure, $\hat{D}$ is the degree matrix, $\hat{D} = \sum_j \hat{A}_{ij}$. W_0 and W_1 represent the weight matrix in the first and second layer, and $\sigma(\cdot)$, $Relu(\quad)$represent the activation function.

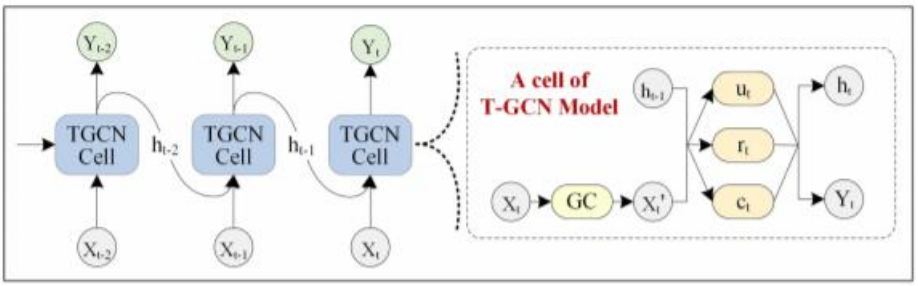

Figure 23: Overall process of spatio-temporal prediction. The right part represents the specific architecture of a T-GCN unit, and GC represents graph convolution

Adjacency Matrix of SZ-Taxi and Los-Loop dataset

The adjacency matric is available in https://github.com/lehaifeng/T-GCN/tree/master/data.

Definitions for the matrices used, regarding Graph Theory

Road Network G: an unweighted graph $G = (V, E)$ was used to describe the topological structure of the road network, and we treat each road as a node, where V is a set of road nodes , $V = \{v1, v2, \cdots, vN\}$, N is the number of the nodes, and E is a set of edges. The adjacency matrix A is used to represent the connection between roads, $A \in R^{NXN}$. The adjacency matrix contains only elements of 0 and 1. The element is 0 if there is no link between roads and 1 denotes there is a link.

Feature Matrix X^{NXP}: Traffic information on the road network as the attribute feature of the node in the network, is expressed as $X \in R^{NXP}$, where P represents the number of node attribute features (the length of the historical time series) and $X_t \in R^{NXi}$ is used to represent the speed on each road at time i. As mentioned, the node attribute features can be any traffic information such as traffic speed, traffic flow, and traffic density.

Thus, the problem of spatio-temporal traffic forecasting can be considered as learning the mapping function f on the premise of road network topology G and feature matrix X and then calculating the traffic information in the next T moments

Proposed Adjacency Matrix:

$$w_{ij} = \begin{cases} exp(-\frac{d_{ij}^2}{\sigma^2}), & i \neq j \; and \; exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon \\ 0, & otherwise \end{cases} \qquad (11)$$

where d_{ij}^2 is the driving distance, $\sigma^2 = 10$ considering a gaussian normalization process and ϵ an empirical boundary.

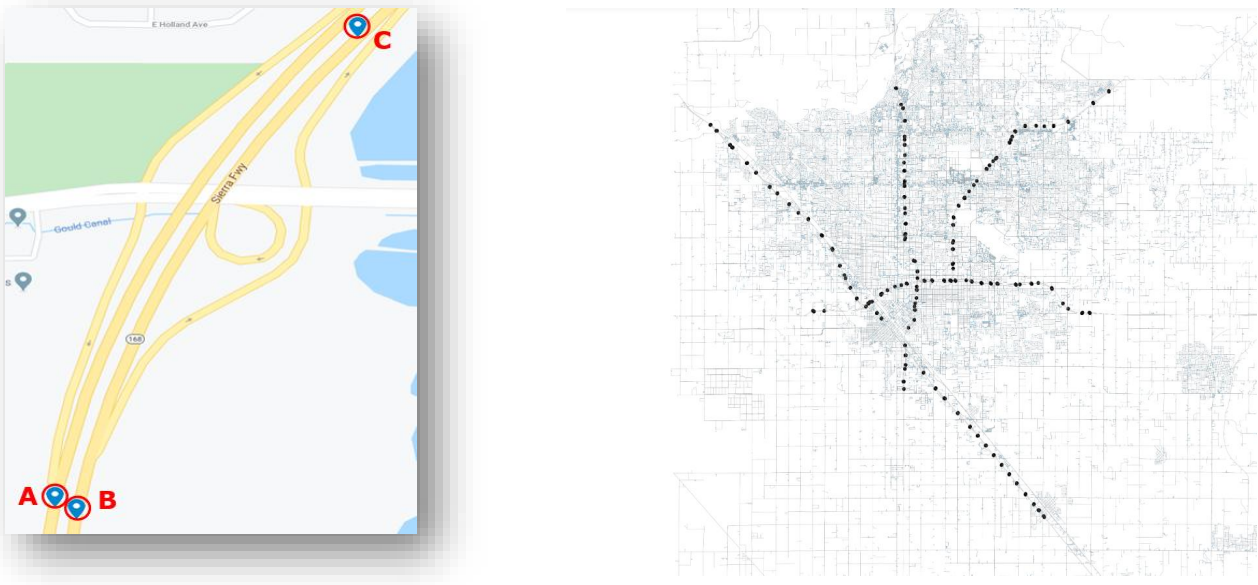

Figure 24: Zoom in network's correlations

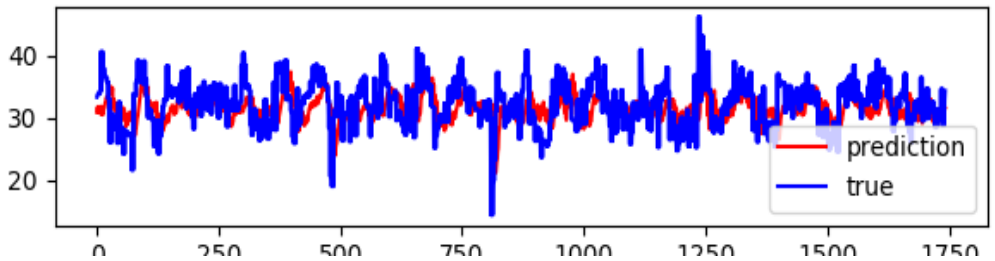

Figure 25: Prediction results of T-GCN algorithm on SZ-taxi public dataset.

Table 13: Performance metrics of T-GCN algorithm on SZ-taxi public dataset.

T-GCN	SZ-Taxi				
	MSE	MAE	ACC	R2	VAR
Epoch: 80	28.32	4.002	0.629	0.7404	0.74052
Epoch: 1000	18.181	2.9387	0.702	0.833	0.8339
Epoch: 1500	16.97	2.85	0.706	0.8385	0.837

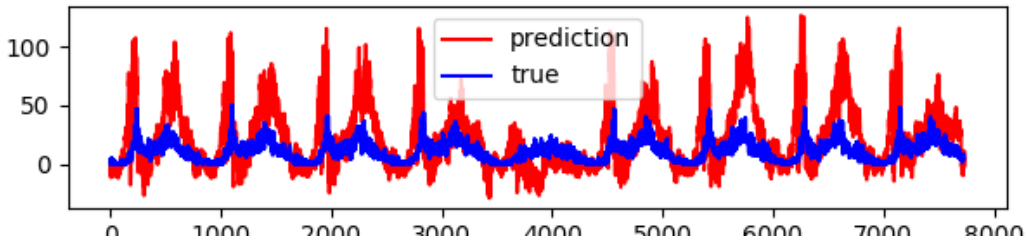

Figure 26: Prediction results of T-GCN algorithm on PEMS public dataset.

Table 14: Performance metrics of T-GCN algorithm on PEMS public dataset.

T-GCN	PEMS				
	MSE	MAE	ACC	R2	VAR
Epoch: 1500	25.32	11.85	0.683	0.743	0.745

2.5.2 Results comparison

Table 15: Performance metrics comparison using CNN, LSTM, GRU and VAR models on public datasets.

Model	PEMS-DATA (SENSOR 0)	PEMS-DATA (SENSOR 25)	Los -Loop (773869)	Sz-Taxi (90217)	Sz-Taxi (90224)
	MAE-MSE	MAE-MSE	MAE-MSE	MAE-MSE	MAE-MSE
CNN	3.08-18.34			2.42-10.23	5.53-49.5
LSTM	3.29-21	15.32-506.37	2.98-30.58	4.62-37.49	8.78- 117.05
GRU	3.4-22.31	15.53-517.22	3.15-31.78	4.63-37.34	
VAR	3.81-14.52	10.92-42.09	4.938-51.79	2.68-12.06	5.96-58.32

As it is witnessed in Figure 21 the prediction is not so good in some specific sensors. That is due to the different characteristics of the timeseries in every sensor. By and large, the timeseries that is highly stochastic is not so easy both in training and in prediction. For this reason, the most "problematic" sensors were isolated for further examination. More specifically, we kept a sensor that gave a good performance, such as sensor 0 at PeMS data, for training the LSTM and GRU model and then test the data of sensor 25 for evaluation.

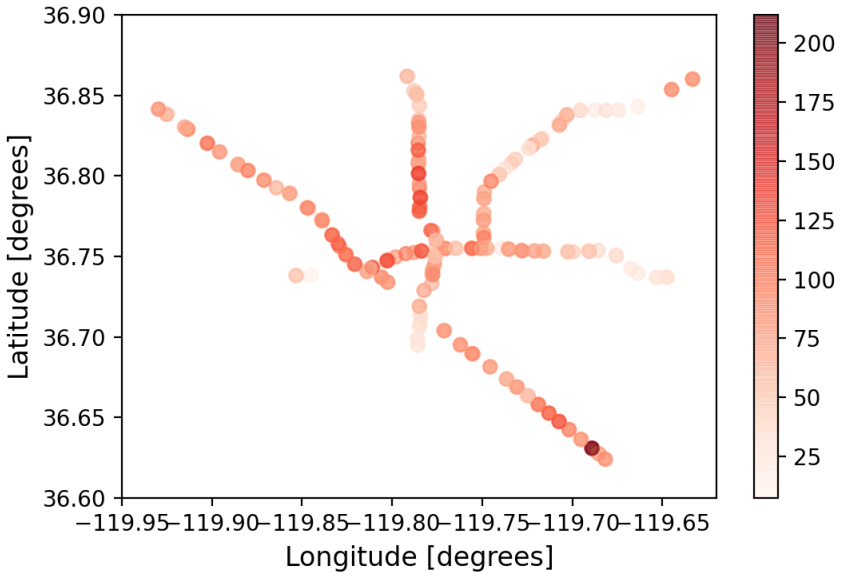

Figure 27: Map of Fresno's traffic flow for the next five minutes.

2.5.3 Methodology on the real-world data

In order to find the optimum solution for the traffic flow prediction task using the real-world data both statistical and machine learning algorithms were applied. In particular, the ARIMA model, XGBoost gradient-boosted decision tree (GBDT) and LSTM models were tested and their performance was compared in terms of the performance metrics mentioned in Section 2.5.1.

Traffic flow data combined with weather information such as the temperature, were used as input features to the models. The final models are taking as input the estimated traffic speed of the road and the temperature in the last hour and a half and they're predicting the traffic speed in the next half hour. In each case, the dataset was divided into training and validation set with a ratio of 80% and 20% respectively.

2.5.3.1 SARIMAX model

Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors, or SARIMAX, is an extension of the ARIMA class of models. ARIMA includes an autoregressive integrated moving average, while SARIMAX includes seasonal effects and eXogenous factors with the autoregressive and moving average component in the model. In the SARIMAX models parameter, two kinds of orders need to be provided. The first one is similar to the ARIMAX model (p, d, q) as it is mentioned in Section 2.5.1.1, and the other is to specify the effect of the seasonality. This order is called seasonal order and requires four elements to be configured:

1. Seasonal autoregressive order
2. Seasonal difference order
3. Seasonal moving average order
4. The number of time steps for a single seasonal period

Mathematically the model is expressed as:

$$\varphi_p(L)\bar{\varphi}_p(L^s)\Delta^d\Delta_s^D y_t = A(t) + \theta_q(L)\bar{\theta}_Q(L^s)\in_t \quad (12)$$

Where:

- $\varphi_p(L)$ is the non-seasonal autoregressive lag polynomial
- $\bar{\varphi}_p(L^s)$ is the seasonal autoregressive lag polynomial
- $\Delta^d\Delta_s^D y_t$ is the time series, differenced d times, and seasonally difference D times
- $A(t)$ is the trend polynomial (including the intercept)
- $\theta_q(L)$ is the non-reasonal moving average lag polynomial
- $\bar{\theta}_Q(L^s)$ is the seasonal moving average lag polynomial

Using the weather temperature as an exogenous factor and defining the p,q,d parameters to 3,2,1 respectively the results emerged from the SARIMAX model are illustrated in Figure 28. As it is shown, in most cases the SARIMAX model correctly predicts the traffic speed the next half hour, recognising when the traffic speed is increased and missing only the exact values of the peaks in the diagram. Table 16 illustrates the performance metrics using SARIMAX model on Slagelse traffic data.

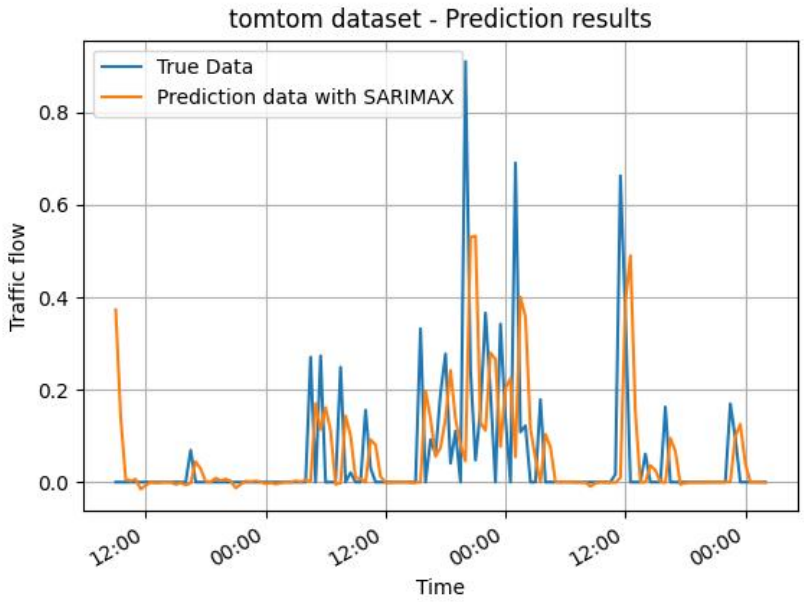

Figure 28: Prediction results using SARIMAX model on Slagelse data

Table 16 : Performance metrics using SARIMAX model on Slagelse data

Dataset	MAE	MSE	Var	R2
Slagelse data	0.0784	0.0261	0.0371	0.0370

2.5.3.2 XGBoost model

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, GBDT machine learning library. It provides parallel tree boosting and is a dominant solution in time series forecasting problems as it is considered as an advancement of traditional modelling techniques (e.g., ARIMA, VAR).
There is a plethora of tuning parameters for tree-based learners in XGBoost, the most common of those are:

- **learning_rate:** step size shrinkage used to prevent overfitting. Range is [0,1]
- **max_depth:** determines how deeply each tree is allowed to grow during any boosting round.
- **subsample:** percentage of samples used per tree. Low value can lead to under fitting.
- **colsample_bytree:** percentage of features used per tree. High value can lead to overfitting.
- **n_estimators**: number of trees to build.
- **objective:** determines the loss function to be used like reg:linear for regression problems, reg:logistic for classification problems with only decision, binary:logistic for classification problems with probability.

XGBoost also supports regularization parameters to penalize models as they become more complex and reduce them to simple (parsimonious) models.

- **gamma:** controls whether a given node will split based on the expected reduction in loss after the split. A higher value leads to fewer splits. Supported only for tree-based learners.
- **alpha:** L1 regularization on leaf weights. A large value leads to more regularization.

- **lambda:** L2 regularization on leaf weights and is smoother than L1 regularization.

Using GridSearchCV as a brute force method to seek for the best parameters, those chosen to be used as input to the model are shown in Table 17. The results emerged from this parametrization are also illustrated in Figure 29. Table 18 shows in detail the performance metrics of this implementation.
According to the above, XGBoost appeared to be less effective in predicting the traffic speed on the specific pilot site.

Table 17 : XGBoost hyper parameters

Parameters	Value
learning_rate	0.3
max_depth	7
subsamples	1
colsample_bytree	1
n_estimators	12
objective	squarederror
gamma	0
alpha	1
lambda	1

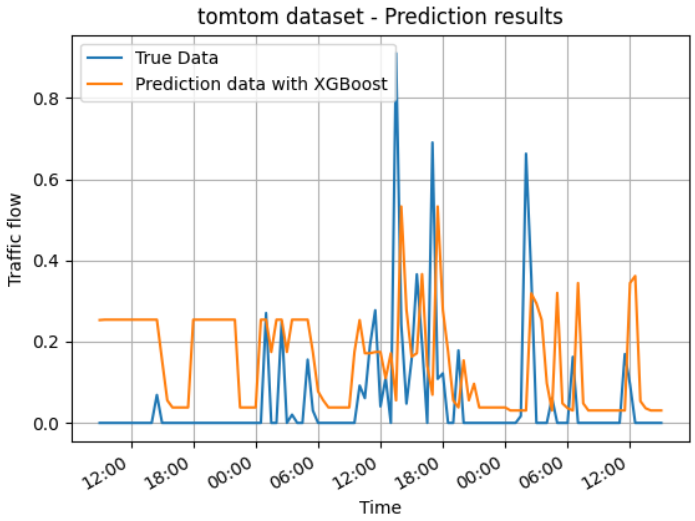

Figure 29 : Prediction results using XGBoost model on Slagelse data

Table 18 : Performance metrics using XGBoost on Slagelse data

Dataset	MAE	MSE	Var	R2
Slagelse data	0.1500	0.0421	-0.6117	-1.0656

2.5.3.3 LSTM model

As already described, LSTM models have been widely used in traffic flow prediction problems since they have reported better performance than some common that used traditional prediction models.

Therefore, a vanilla LSTM architecture with 50 units (Figure 30) has been tested on Slagelse data in order to evaluate its performance for the specific traffic flow prediction problem.

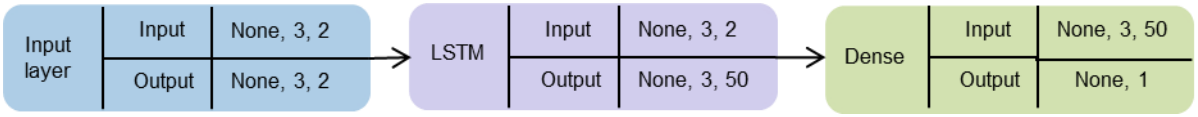

Figure 30 : Architecture of vanilla LSTM model.

The dataset was divided into training and validation set with a ratio of 80% and 20% respectively. For this regression problem, the stochastic gradient descent (SGD) and the MeanSquaredError loss function for optimization have been used. Figure 31 and Table 19 illustrate the prediction results emerged from the LSTM model. As it is shown, LSTM appeared to be less efficient on the current problem, probably due to the small dataset size.

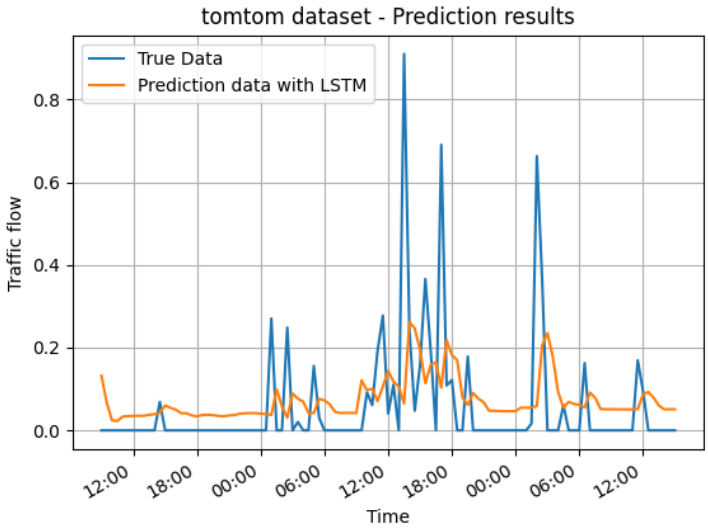

Figure 31: Prediction results using LSTM model on Slagelse data.

Table 19 : Performance results using LSTM model on Slagelse data.

Dataset	MAE	MSE	Var	R2
Slagelse data	0.0857	0.0201	0.0276	0.0128

2.5.4 Results comparison

After experimenting with different models and various parametrizations, SARIMAX appeared to achieve the highest performance on Slagelse data as it is shown in Table 20. Therefore, the statistical model, SARIMAX is suggested as the optimum solution for predicting traffic speed on this specific pilot site and providing an insight of the traffic road state each half hour. In general, statistical models are preferable when the dataset size is small and there is no high complexity of data.

Table 20 : Comparison of performance metrics using SARIMAX, XGBoost and LSTM model.

Model	MAE	MSE	Var	R2
SARIMAX	0.0784	0.0261	0.0371	0.0370
XGBoost	0.1500	0.0421	-0.6117	-1.0656
LSTM	0.0857	0.0201	0.0276	0.0128

2.6 Future Work

Since traffic data for each pilot site was not directly available via the existing equipment, additional procedures were needed to acquire necessary data for the T5.4 implementation. Moreover, the fact that the pilot site in Slagelse is a rural road and in most cases there was no significant traffic, rendered a large part of the data unusable for the model training. Therefore, additional data would enhance model's performance having a better insight of the area for the entire year. The working, and non-working traffic prediction models could be further improved and a model would be developed and activated for each season, month, week, etc. based on the input features.

Finally, in order to create traffic prediction models for all the pilot sites, a large amount of traffic data should be collected for each of them. Thus far, the amount of data is not sufficient to train a deep learning model for the remaining pilot locations whose coordinates have been provided. After the collection of complementary data, the process described in the current deliverable would also be applied to the new pilot sites data. The proposed model architectures would be tested and compared having as input the new data and the one with the best performance would be suggested as the optimum solution for predicting the traffic speed on the specific road.

3 Energy Consumption in Electric Vehicles

Electric vehicles (EVs) and Autonomous vehicles (AVs) are emerging solutions and have become viable technology in recent years. AVs are considered a major disruptive technology in public transportation. Since that, decades of research have been spent on batteries and power electronic development. The technology to enable vehicle automation is developing rapidly, however, energy consumption remains an open problem.

AVs are equipped with certain automated features, such as adaptive and optimised energy prediction. At AVs, improvements in energy consumption efficiency can be achieved through energy management strategies, such as battery level prediction. Proper estimation and prediction at the battery level can be approached using machine learning concepts.

In the next sessions, state of the art and data analysis on AV battery data is presented. Moreover, a prediction model for energy consumption is described. The data of the current document are associated with the HOLO vehicles.

3.1 Literature Review

Recent studies have shown the necessity of electric AVs in public transportation. AVs generally use electricity to operate and in most cases are equipped with embedded high-technological sensors. This new technology can transport passengers without the need for drivers and fossil-fuel-driven engines that can pollute the atmosphere. Many European citizens are affected by air pollution and they are exposed to high levels of pollution that can be harmful to public health[1]. Electric or even hybrid AVs have widespread use, however, energy management in these vehicles is of major importance. It is necessary to investigate novel methods to properly manage energy consumption. The current studies are focused basically on establishing optimal approaches for energy consumption in hybrid vehicles. There is still a gap in the research of holistic energy efficiency on fully electric AVs. For this reason, more and more studies are researching robust ways to tackle the energy consumption challenge in AVs. In [59] several board categories that can influence the energy consumption in AVs are discussed. This paper remarks significant factors among vehicle characteristics, transportation network and consumer choice are affecting the energy consumption in AVs. Moreover, in [60] he authors propose an interesting method that employs fuzzy-logic advances to reduce the energy consumption of the vehicle. This work relies on modelling several factors among environmental, vehicle operation and specification. Based on these, a schematic power flow model was studied.

[1] https://www.eea.europa.eu/media/infographics/air-pollution-exposure-in-cities

3.2 Battery Data from HOLO AVs

HOLO provided energy data from five weeks in order to understand and analyse patterns and trends in the battery levels from AVs. Figure 32 presents an indicative example of the structure of input energy data. The timestamp of the recording as well as the location in the form of latitude and longitude, are delivered. Moreover, information for the battery is given involving the battery status which describes if the AV is charging or not and the current battery level, which can take values from 0 to 100. Finally, route details are offered to define the speed of the AV, the covered distance and the navigation mode, which refers basically to AYTO and MANUAL in cases where the operator handles the vehicles.

timestamp	timestamp_int	battery_status	battery_level	speed_value	distance	latitude	longitude	navigation_mode
2021-10-18 06:20:40.411 UTC	1.634538E+12	NO_CHARGE	100	0	0	55.4006950000032	11.367751	AUTO
2021-10-18 06:20:41.413 UTC	1.634538E+12	NO_CHARGE	100	0	0	55.4006950000032	11.367751	AUTO
2021-10-18 06:20:42.413 UTC	1.634538E+12	NO_CHARGE	100	0	0	55.4006950000032	11.367751	AUTO
2021-10-18 06:20:43.413 UTC	1.634538E+12	NO_CHARGE	100	0	0	55.4006950000032	11.367751	AUTO
2021-10-18 06:20:44.414 UTC	1.634538E+12	NO_CHARGE	100	0	0	55.4006950000032	11.367751	AUTO
2021-10-18 06:20:45.415 UTC	1.634538E+12	NO_CHARGE	100	0	0	55.4006950000032	11.367751	AUTO

Figure 32: Sample of energy data provided by HOLO

Using the knowledge mentioned above, the battery prediction problem was formulated. However, at this point, it is important to note that to model AV energy consumption properly, more information is required. Several external factors among the number of passengers, probable slopes on the route, and air-conditioning enabling can affect the energy consumption in an AV.

Despite that, in the AVENUE project, we try to develop an initial model for battery prediction that can be easily adapted to future solutions with additional data.

In Figure 33, battery level and status are presented concerning a sample day of the given data. The blue line in the plot describes the battery level that decreases over time since the AV is moving and following a specific route. In the same notions, the orange line represents the battery status. Based on Figure 33 a battery status equal to zero is assigned to NO_CHARGE status and a battery status equal to 100 refers to CHARGE status. In particular, in the example figure, it seems that the vehicle was fully charged at the begging of the day. Then at the end of the route, the vehicle battery status was turned from NO_CHARGE to CHARGE.

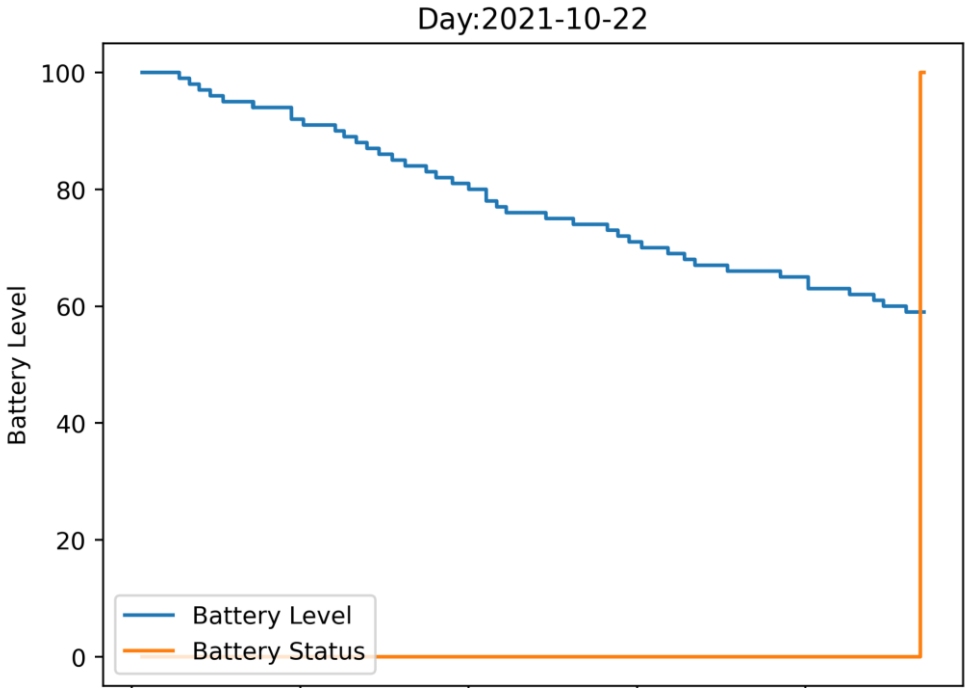

Figure 33: Daily battery data concerning the level and state of the battery

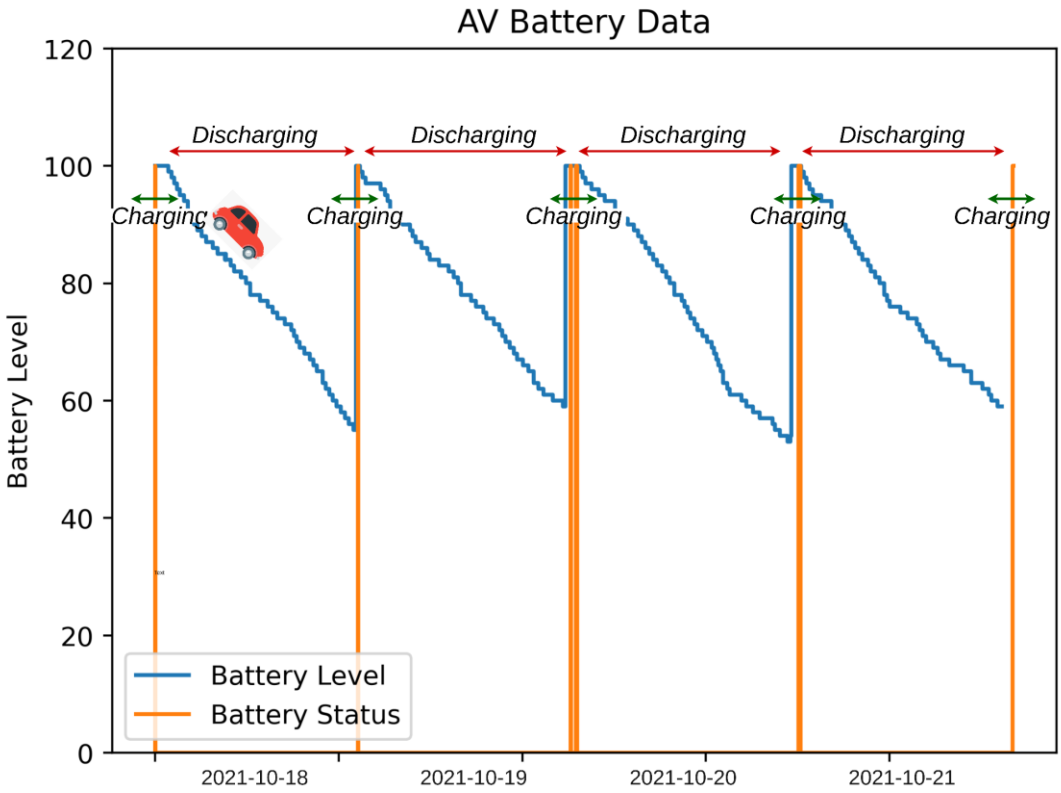

Figure 34: Overview of AV Battery data, illustrating the Battery Status (Charging/ Discharging) and the Battery Level trends

3.2.1 Overview and Limitations

The overview of the HOLO battery data gives that the AV follow a very specific pattern in battery consumption which is easily visible and recognisable. This fact creates many limitations in the energy consumption problem formulation. These patterns concern mainly battery discharging trends. Moreover, there is a specific pattern in the battery state in which the AV is charging at the end of each day. However, the most important limitation on the daily time window of the data. Battery data is provided for a certain period of the day which mainly corresponds to almost office working hours.
All these limitations can affect the prediction performance of the energy consumption model. A machine learning model can overfit in explicit data patterns and not be able to generalise in more cases.

3.3 Problem Formulation

In this section, we describe the formulation of the energy consumption model based on the HOLO AV battery data.

3.3.1 Cyclic time

To proper formulate the energy consumption model, we initialised cyclic time features for our problem. When dealing with time series data in the context of machine learning algorithms, it is important to use encoded properties of time. In the case of the energy consumption model, the time of day is an important factor in correctly predicting the battery level. For example, 23:59 to 00:00 concerns a separate day but the model would not consider the time gap as 1 minute apart.
A possible solution to this problem relies on feature engineering which means using cyclic time instead of conventional. It is more efficient to understand cyclic patterns such as hours, minutes, and seconds of the day.
As a result, the cyclicity of time could be very helpful in our time series forecasting. Cyclic time uses periodic functions such as "sine" and "cosine" to encode time data. Figure 35 represents the encoding of the conventional time to the cyclic time. The cyclic time was used as an extra feature in the energy consumption forecasting model.

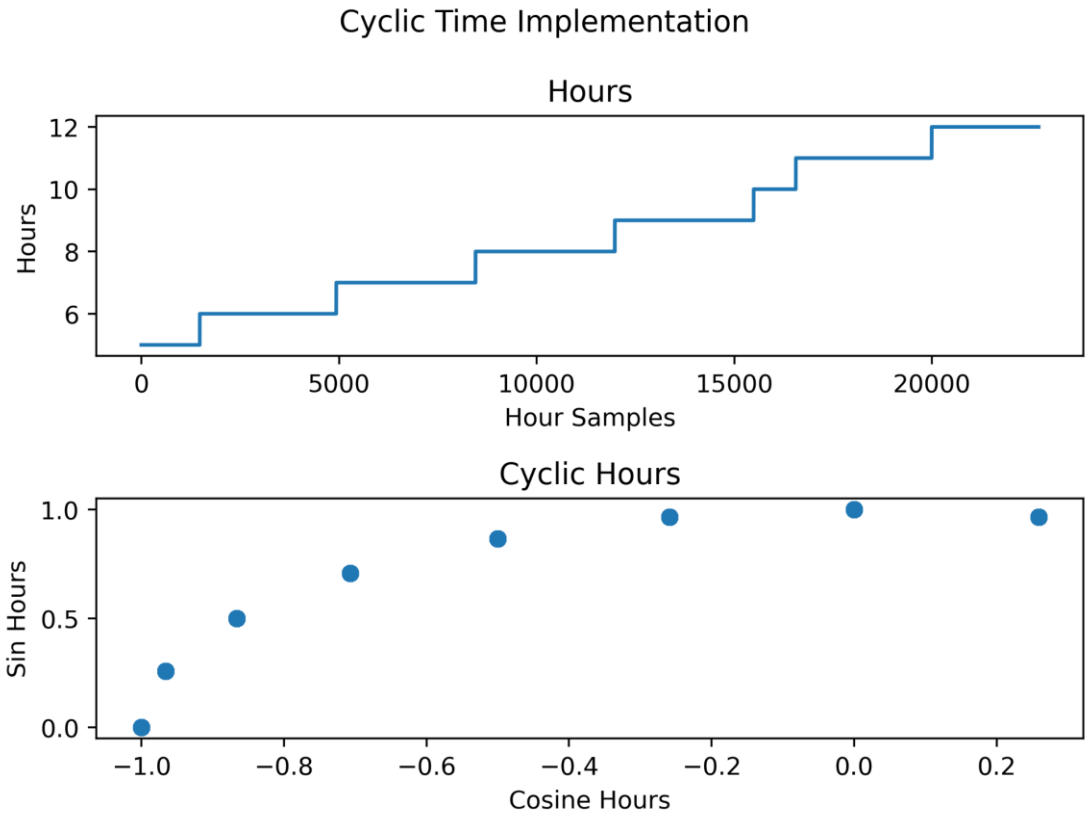

Figure 35: Conventional time (hours) to cyclic time (hours) conversion

3.3.2 XGBoost

For the energy consumption forecasting task gradient boosting approach was employed. **XGBoost** is an optimized distributed gradient boosting framework which was designed to be efficient, flexible and portable especially. The stochastic gradient boosting algorithm, also called gradient boosting machines or tree boosting, is a powerful machine learning technique that performs well on many machine learning challenges, involving time series forecasting.

Our case refers to time series data modelling, which can be phrased as supervised learning. We give a sequence of numbers for a time series dataset, so we can restructure the data to understand patterns. Figure 36 demonstrates the architecture of the XGBoost algorithm that was employed to design the energy consumption, forecasting model.

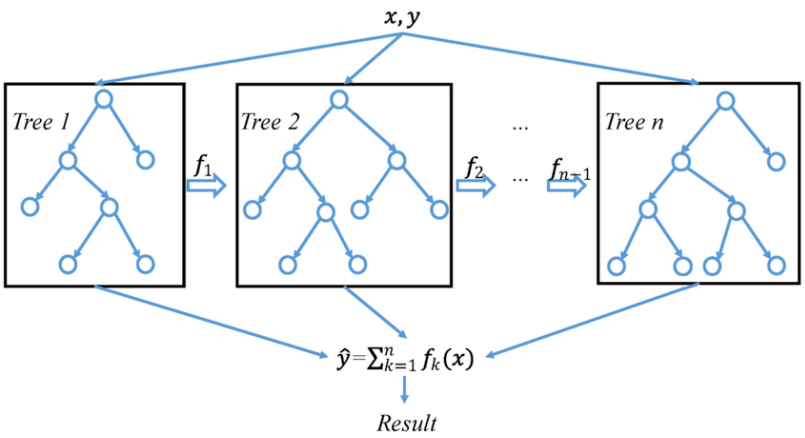

Figure 36: XGBoost algorithm representation

Time series forecasting is very sensitive to sequences of data. For this reason, instead of using previous knowledge of battery data to predict the next steps, we employed cyclic time features. Finally, the input of the XGBoost algorithm, as demonstrated in Figure 37, involves the battery status, the AV speed, the AV distance and the cyclic time. The output of the forecasting model is the battery level of the AV.

Figure 37: AV Battery prediction model formulation

3.4 Energy Consumption Prediction

In this section the results and the evaluation metrics of the energy consumption forecasting are presented. For the training of the forecasting model, we used 70% of the data and the rest of 30% for evaluating the performance of the model. Figure 38 presents the forecasting results of the XGBoost forecasting model. The blue line represents the true battery values and the orange one the predicted values. It is observable that the XGBoost model almost achieves to follows the exact pattern of the true battery levels.

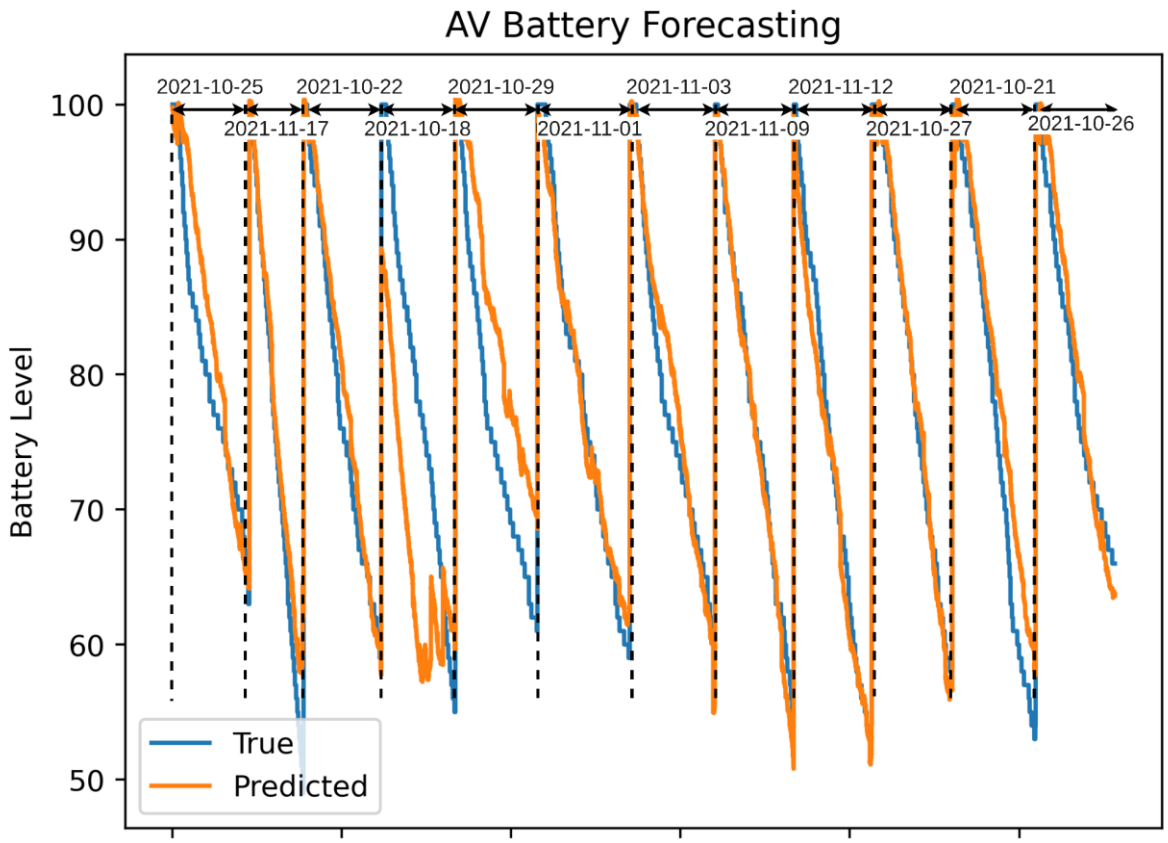

Figure 38: AV Battery forecasting for indicative testing days

For the evaluation of the prediction model we used well-known forecasting performance metrics, as described in Table 21. These metrics are among Mean absolute error (MAE), Mean squared error (MSE), Root means squared error (RMSE) and R-squared.

The average of the absolute difference between the dataset's actual and anticipated values is represented by the Mean Absolute Error. It calculates the dataset's residuals' average. The mean of the squared difference between the data set's original and forecasted values is known as the mean squared error. It calculates the residuals' variance. The square root of the Mean Squared Error is called Root Mean Squared Error. It calculates the residuals' standard deviation.

Table 21: Performance of the XGBoost forecasting model

Forecasting Performance Metrics	
MAE	3.4787
MSE	25.6513
RMSE	5.0647
R-squared	0.8447

Based on the overview of the results, optimal predictive maintenance in determining the direction of future battery consumption trend.

4 Bibliography

[1] C. P. V. Hinsbergen, J. W. V. Lint and F. M. Sanders, "Short term traffic prediction models," in *PROCEEDINGS OF THE 14TH WORLD CONGRESS ON INTELLIGENT TRANSPORT SYSTEMS (ITS), HELD BEIJING, OCTOBER 2007*, 2007.

[2] S. Clark, "Traffic prediction using multivariate nonparametric regression," *Journal of transportation engineering,* vol. 129, p. 161–168, 2003.

[3] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research Part B: Methodological,* vol. 18, p. 1–11, 1984.

[4] H. Yin, S. Wong, J. Xu and C. K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transportation Research Part C: Emerging Technologies,* vol. 10, p. 85–98, 2002.

[5] Y. Li, R. Yu, C. Shahabi and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926,* 2017.

[6] C.-H. Wu, J.-M. Ho and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems,* vol. 5, p. 276–281, 2004.

[7] F. Zuurbier, H. van Lint and V. Knoop, "Traffic network state estimation using extended Kalman filtering and DSMART," *IFAC Proceedings Volumes,* vol. 39, p. 37–42, 2006.

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

[9] M. Williams, Y. Malhi, A. D. Nobre, E. B. Rastetter, J. Grace and M. G. P. Pereira, "Seasonal variation in net carbon exchange and evapotranspiration in a Brazilian rain forest: a modelling analysis," *Plant, Cell & Environment,* vol. 21, p. 953–968, 1998.

[10] L. Catania and T. Proietti, "Forecasting volatility with time-varying leverage and volatility of volatility effects," *International Journal of Forecasting,* vol. 36, p. 1301–1317, 2020.

[11] P. Billaudel, A. Devillez and G. V. Lecolier, "Performance evaluation of fuzzy classification methods designed for real time application," *International journal of approximate reasoning,* vol. 20, p. 1–20, 1999.

[12] Z. He and L. Zheng, "Visualizing traffic dynamics based on floating car data," *Journal of Transportation Engineering, Part A: Systems,* vol. 143, p. 04017005, 2017.

[13] S. S. Haykin and S. S. Haykin, Kalman filtering and neural networks, vol. 284, Wiley Online Library, 2001.

[14] J. Guo, W. Huang and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies,* vol. 43, p. 50–64, 2014.

[15] S. Jin, D.-h. Wang, C. Xu and D.-f. Ma, "Short-term traffic safety forecasting using Gaussian mixture model and Kalman filter," *Journal of Zhejiang University SCIENCE A,* vol. 14, p. 231–243, 2013.

[16] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Vehicular Communications,* vol. 20, p. 100184, 2019.

[17] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation*

Systems, vol. 14, p. 1700–1707, 2013.

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

[19] R. Fu, Z. Zhang and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016.

[20] S. Yakowitz, "Nearest-neighbour methods for time series analysis," *Journal of time series analysis,* vol. 8, p. 235–247, 1987.

[21] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *Journal of Transportation Engineering,* vol. 117, p. 178–188, 1991.

[22] B. L. Smith, B. M. Williams and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies,* vol. 10, p. 303–321, 2002.

[23] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of transportation engineering,* vol. 123, p. 261–266, 1997.

[24] Z. Zheng and D. Su, "Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transportation Research Part C: Emerging Technologies,* vol. 43, p. 143–157, 2014.

[25] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications,* vol. 36, p. 6164–6173, 2009.

[26] S. Sun, C. Zhang and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems,* vol. 7, p. 124–132, 2006.

[27] B. Ghosh, B. Basu, M. O'Mahony and others, "Bayesian time-series model for short-term traffic flow forecasting," *Journal of transportation engineering,* vol. 133, p. 180–189, 2007.

[28] M. S. Dougherty and M. R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *International journal of forecasting,* vol. 13, p. 21–31, 1997.

[29] C. Ledoux, "An urban traffic flow model integrating neural networks," *Transportation Research Part C: Emerging Technologies,* vol. 5, p. 287–300, 1997.

[30] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks,* vol. 2, p. 359–366, 1989.

[31] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural networks,* vol. 2, p. 183–192, 1989.

[32] J. Hua and A. Faghri, "Apphcations of artificial neural networks to intelligent vehicle-highway systems," *Transportation Research Record,* vol. 1453, p. 83, 1994.

[33] A. Stathopoulos, L. Dimitriou and T. Tsekeris, "Fuzzy Modeling Approach for Combined Forecasting of Urban Traffic Flow," *Computer-Aided Civil and Infrastructure Engineering,* vol. 23, p. 521–535, 2008.

[34] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research,* vol. 131, p. 253–261, 2001.

[35] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of transportation engineering,* vol. 131, p. 771–779, 2005.

[36] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies,* vol. 19, p. 387–399, 2011.

[37] J. W. C. Van Lint, S. P. Hoogendoorn and H. J. van Zuylen, "Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks," *Transportation Research Record,* vol. 1811, p. 30–39, 2002.

[38] J. L. Elman, "Finding structure in time," *Cognitive science,* vol. 14, p. 179–211, 1990.

[39] J. W. C. Van Lint, S. P. Hoogendoorn and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C: Emerging Technologies,* vol. 13, p. 347–369, 2005.

[40] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies,* vol. 11, p. 121–135, 2003.

[41] J. W. C. Van Lint and C. P. I. J. Van Hinsbergen, "Short-term traffic and travel time prediction models," *Artificial Intelligence Applications to Critical Transportation Issues,* vol. 22, p. 22–41, 2012.

[42] L. Shen, "Freeway travel time estimation and prediction using dynamic neural networks," 2008.

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

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

[45] J. Li, L. Gao, W. Song, L. Wei and Y. Shi, "Short term traffic flow prediction based on LSTM," in *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, 2018.

[46] H. Bohan and B. Yun, "Traffic flow prediction based on BRNN," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2019.

[47] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, 2012.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, p. 1735–1780, 1997.

[49] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," *arXiv preprint arXiv:1612.01022,* 2016.

[50] J. Zhang, Y. Zheng and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[51] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors,* vol. 17, p. 818, 2017.

[52] S. Sun, C. Zhang and Y. Zhang, "Traffic flow forecasting using a spatio-temporal bayesian network predictor," in *International conference on artificial neural networks*, 2005.

[53] H. Yu, Z. Wu, S. Wang, Y. Wang and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors,* vol. 17, p. 1501, 2017.

[54] M. Defferrard, X. Bresson and P. Vandergheynst, "Convolutional neural networks on graphs with

fast localized spectral filtering," *Advances in neural information processing systems,* vol. 29, 2016.

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

[56] Y. Lv, Y. Duan, W. Kang, Z. Li and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, p. 865–873, 2014.

[57] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing,* vol. 50, p. 148–163, 2018.

[58] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, p. 3848–3858, 2019.

[59] W. R. Morrow, J. B. Greenblatt, A. Sturges, S. Saxena, A. Gopal, D. Millstein, N. Shah and E. A. Gilmore, "Key factors influencing autonomous vehicles' energy and environmental outcome," in *Road vehicle automation*, Springer, 2014, p. 127–135.

[60] D. Phan, A. Bab-Hadiashar, C. Y. Lai, B. Crawford, R. Hoseinnezhad, R. N. Jazar and H. Khayyam, "Intelligent energy management system for conventional autonomous vehicles," *Energy,* vol. 191, p. 116476, 2020.