NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

Prediction Quality of Service in 5G Networks

Angeliki P. Kalamari

Supervisors: **Athanasia Alonistioti,** Associate Professor

ATHENS

MARCH 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη Ποιότητας Υπηρεσιών σε 5G Δίκτυα

Αγγελική Π. Καλαμάρη

Επιβλέποντες: **Αθανασία Αλωνιστιώτη,** Αναπληρώτρια Καθηγήτρια

ΑΘΗΝΑ

ΜΑΡΤΙΟΣ 2022

BSc THESIS

Prediction Quality of Service in 5G Networks

Angeliki P. Kalamari
S.N.: 1115201400255

SUPERVISOR: **Athanasia Alonistioti,** Associate Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη Ποιότητας Υπηρεσιών σε 5G Δίκτυα

Αγγελική Π. Καλαμάρη
Α.Μ.: 1115201400255

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Αθανασία Αλωνιστιώτη,** Αναπληρώτρια Καθηγήτρια

ABSTRACT

On the eve of 5G-enabled Connected and Automated Mobility, challenging Vehicle-to-Everything services have emerged towards safer and automated driving. The requirements that stem from those services pose very strict challenges to the network primarily with regard to the end-to-end delay and service reliability. At the same time, the in-network Artificial Intelligence that is emerging, reveals a plethora of novel capabilities of the network to act in a proactive manner towards satisfying the aforementioned challenging requirements. This work presents PreQoS, a predictive Quality of Service mechanism that focuses on Vehicle-to-Everything services. PreQoS is able to timely predict specific Quality of Service metrics, such as uplink and downlink data rate and end to-end delay, in order to offer the required time window to the network to allocate more efficiently its resources. On top of that, the proactive management of those resources enables the respective Vehicle-to-Everything services and applications to perform any potential Quality of Service-related required adaptations in advance. The evaluation of the proposed mechanism based on a realistic, simulated, Connected and Automated Mobility environment proves the viability and validity of such an approach.

SUBJECT AREA: 5G Networks

KEYWORDS: 5G, CAM, Quality of Service prediction, V2X, NS3

ΠΕΡΙΛΗΨΗ

Την παραμονή της συνδεδεμένης και αυτοματοποιημένης κινητικότητας (CAM) με δυνατότητα 5G, εμφανίστηκαν οι απαιτητικές υπηρεσίες όχημα-σε-οτιδήποτε (V2X) για αυτοματοποιημένη και ασφαλέστερη οδήγηση. Οι απαιτήσεις που απορρέουν από αυτές τις υπηρεσίες δημιουργούν πολύ αυστηρές προκλήσεις για το δίκτυο κυρίως όσον αφορά τον βασικό δείκτη απόδοσης (KPI) καθυστέρησης από άκρο σε άκρο (end-to-end delay). Ταυτόχρονα, η τεχνητή νοημοσύνη (AI) που εμφανίζεται εντός του δικτύου, αποκαλύπτει μια πληθώρα νέων δυνατοτήτων του δικτύου, να ενεργεί με προληπτικό τρόπο ως προς την ικανοποίηση των προαναφερθεισών απαιτήσεων. Αυτή η πτυχιακή εργασία παρουσιάζει έναν μηχανισμό πρόβλεψης ποιότητας υπηρεσιών (PreQoS), που υποστηρίζεται από τεχνητή νοημοσύνη, εστιάζει στις υπηρεσίες όχημα-σε-οτιδήποτε και είναι σε θέση να προβλέψει έγκαιρα συγκεκριμένες μετρήσεις ποιότητας υπηρεσίας. Παράδειγμα αυτών των υπηρεσιών είναι ο ρυθμός δεδομένων (data rate) και η καθυστέρηση στις ανερχόμενες (uplink) και κατερχόμενες ζεύξεις (downlink) από άκρο σε άκρο, προκειμένου να προσφέρει το απαιτούμενο χρονικό παράθυρο στο δίκτυο για να κατανείμει αποτελεσματικότερα τους πόρους του, καθώς και στις αντίστοιχες υπηρεσίες και εφαρμογές όχημα-σε-οτιδήποτε για την εκτέλεση των απαιτούμενων προσαρμογών. Η αξιολόγηση του προτεινόμενου μηχανισμού βασίζεται σε ένα ρεαλιστικό, προσομοιωμένο περιβάλλον όχημα-σε-οτιδήποτε που αποδεικνύει τη βιωσιμότητα και την εγκυρότητα μιας τέτοιας προσέγγισης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: 5G Δίκτυα

ΚΛΕΙΔΙΑ: 5G, Αυτοματοποιημένη Κινητικότητα, Πρόβλεψη Ποιότητας Υπηρεσίας, V2X, Προσομοιωτής Δικτύου 3

This page intentionally left blank

AKNOWLEDGMENTS

I would like to deeply thank my supervisors, prof. Nancy Alonistioti and PhD candidate Panagioti Kontopoylo for the continuous support during the development of this thesis. Their contribution was really valuable for me in order get results of better quality.

CONTENTS

LIST OF FIGURES

LIST OF IMAGES

LIST OF TABLES

PREFACE

The current thesis has been conducted for the bachelor's program degree offered by the department of Informatics and Telecommunication from the National and Kapodistrian University of Athens. The main study of this thesis concerns the development of an innovative QoS prediction scheme for V2X communications, namely PreQoS, which is able to accurately predict predefined QoS metrics, such as ul/dl delay or data rate, ultimately enabling the network and involved applications to perform the required adaptations for avoiding service interruption. In the context of the present work, the proposed system has been implemented using SUMO and NS3 for the Software-Defined-Networking (SDN) scenario and Jupyter along with Python for the related algorithms and methods, as well as for the visualization of the experimental results. The choice of this topic is due to my interest in the field of 5G Networks, Machine Learning and its numerous applications.

1. INTRODUCTION

Beyond 5G network intelligence is already considered as one of the cornerstones for the next generation of wireless and mobile systems. Architecture enhancements for 5G System (5GS) to support network data analytics services in recent releases [1], already pave the way for the implementation of diverse Machine Learning (ML) and Artificial Intelligence (AI)-based resource management, security-related and application-/service-oriented enhancements. The Network Data Analytics Function (NWDAF) introduced by 3GPP and ETSI [2], is able to interact with different network entities for different purposes, such as data collection based on subscription to events, retrieval of information from data repositories, on demand provision of analytics to consumers, etc. In parallel, the extreme service requirements introduced already by 5G are further defined and standardized in 3GPP Release 16 [3]. In this specification, 5G Quality of Service (QoS) Identifiers (5QIs) are mapped to specific QoS characteristics, in relation to the respective resource type, such as Guaranteed Bit Rate/Non-Guaranteed Bit Rate, priority level, packet error rate thresholds, etc.

One of the most challenging and at the same time broadly investigated use cases for 5G networks and beyond, i.e., Connected and Automated Mobility (CAM), along with its respective communication services, namely Vehicle-to-Everything (V2X) and Cellular V2X (C-V2X), is already progressing via several architectural and service-oriented enhancements from the standardization organizations, such as 3GPP and ETSI, both from the 5G Core (5GC), as well as the Radio Access Network (RAN) and Edge aspects [4] [5] [6] [7] [8]. 3GPP has defined the main use cases (UCs) for V2X scenarios, namely Vehicles Platooning, Advanced Driving, Extended Sensors, Remote Driving and Vehicle QoS Support. The 5G Automotive Association (5GAA) has also defined a number of more fine-grained CAM use cases [9], namely Tele-operated driving (ToD), Anticipated Cooperative Collision Avoidance, High-density platooning, Hazardous location warning, lane merge, Software update and Infotainment. In [10], the use cases, requirements, and design considerations for vehicle-to- everything communications are presented. Also, the authors in [11] describe the current challenges, focusing on the 5G crossborder V2X operations for CAM, providing also an overview of the proposed technologies and solutions.

CAM applications rely on the network reliability and QoS in order to address requirements expressed in terms of ubiquitous coverage, minimum uplink/downlink and sidelink data rates, acceptable packet loss ratio, maximum allowed packet delay, etc. Towards this direction, 5GAA has very recently introduced the concept of predictive QoS [12] that refers to the mechanisms enabling mobile networks to provide notifications about predicted QoS changes to interested consumers in advance. Mobile/Multiple Access Edge Computing (MEC), -which is considered one of the essential technologies for 5G-, is also a key enabler for CAM and V2X. In [13], the automotive use cases that are relevant for MEC are showcased, providing insights into the technologies specified and investigated by the ETSI MEC ISG. ETSI, -in the context of MEC- has also very recently introduced the notion of predictive QoS support in the context of the MEC framework [8]. In this context, the prediction of potential handovers leading to the estimated QoS performance is described as the key solution. This will enable the UEs/vehicles to proactively identify the MEC hosts and base stations, which will be able to support the relevant V2X application requirements without any service interruption. Also, in [14] ETSI specification, the API resource, along with the detailed data model for the QoS prediction of a vehicular UE are provided.

Based on the above, it becomes thus obvious that network intelligence, based on state-of-the-art AI and ML algorithms, towards the enhancement of V2X services can prove of utmost importance. The stringent requirements of the majority of the CAM use cases ask for proactive resource allocation approaches in order to ensure that the V2X communications are adequately supported and satisfy the specific reliability, end-to-end (E2E) delay and data

rate (both in the downlink, as well as the uplink) requirements. On top of that, the promising network edge capabilities may prove valuable for bringing network intelligence closer to the network nodes and in a distributed manner, enabling thus even more efficient and low latency proactive network management operations to apply.

1.1 Primary Contributions

Inspired by the above-mentioned challenges and topics, for the current thesis, I focus on QoS prediction for V2X services and the modeling of the overall V2X network and environment towards an efficient and viable predictive QoS solution. My primary contributions are summarized as follows:

- A predictive QoS algorithm is presented, namely PreQoS, which processes contextual information such as vehicle mobility information, radio and network parameters and load conditions, as well as application-specific information and generates predicted QoS-related values, such as uplink/downlink end-to-end delay, data rate, etc. Towards proactive service adaptation for service continuity in stringent V2X services.
- The overall problem is thoroughly analyzed in terms of spatial, temporal and system modeling; a novel approach is presented, namely Map-as-a-Grid (MaaG), which offers higher performance of the QoS prediction in terms of computing and memory requirements, highlighting the potential of the proposed mechanism from the scalability perspective.
- The capability of the proposed mechanism to integrate and operate via exploiting diverse machine learning algorithms is demonstrated, such as Deep Neural Networks, Random Forests, Distributed Gradient Boosting schemes, etc., on top of the spatio-temporal modeling of the MaaG approach. A detailed evaluation procedure is presented, which demonstrates the gains of the MaaG approach, correlates the accuracy error and coefficient determination metrics in relation to the spatial modeling options and volume of training data, while also provides insights in terms of the performance of different ML models, which are exploited by the PreQoS framework.

1.2 What is 5G

5G is a new global wireless standard for broadband cellular networks and is the successor to the 4G networks. 5G enables a new kind of network that is designed to connect virtually everyone and everything together including machines, objects, and devices [15]. Like its predecessors, 5G networks are cellular networks, in which the service area is divided into small geographical areas called *cells*. All 5G wireless devices in a cell are connected to the internet and telephone networks by radio waves through a local antenna in the cell. The main advantage of the new networks is that they will have greater bandwidth, giving higher download speeds, eventually up to 10 gigabits per second (Gbit/s). In addition to 5G being faster than existing networks, 5G has higher bandwidth and can thus connect more different devices, improving the quality of Internet services in crowded areas. Due to the increased bandwidth, it is expected the networks will increasingly be used as general internet service providers (ISPs) for laptops and desktop computers, competing with existing ISPs such as cable internet, and also will make possible new applications in internet-of-things (IoT) and machine-to-machine areas. Cellphones with 4G capability alone are not able to use the new networks, which require 5G-enabled wireless devices.

Since 5G started being deployed worldwide autonomous driving stopped being just a vision in a science fiction movie. Autonomous driving means that the car is fully Autonomous means that the car is fully independent in making decisions and responding to situations, including emergencies; no driver and no external intervention are needed. Enabling an even faster connection between transport systems, the 5G network will offer new application options advancing the development of autonomous cars. Not only will they be able to make

autonomous decisions in the future, they will also communicate and cooperate with each other. Automated driving is the term used to describe a scenario where a fully interconnected and intelligent road transport system is created as a result of these capabilities.

Thanks to wireless technology and internet connection, connected cars with their digital and location-related services can greatly improve our driving comfort. The car relies on regular data updates for navigation, e.g., detailed road maps, plus updates in unexpected traffic situations, such as congestion, rain, or black ice. In combination with apps for the driver and cloud systems, information for maintenance or other status reports can be retrieved and sent. Thanks to mobile edge computing, these functions are already realized today, based on (LTE) at a transmission rate of up to 300 megabit per second and latencies of less than 100 milliseconds, even in emergencies or remote-controlled driving at low speeds. 5G will offer even higher quality for many digital in-car services in the future.

Image 1: 5G Visualization

One huge benefit of 5G is what is known as network slicing. The wireless network is subdivided into virtual network levels. One network level is then used only for automated driving, for instance. This ensures that safety-relevant notifications to self-driving cars will not end up in a traffic jam on the data highway and will be given priority over other infotainment services used in parallel.

Humans, in general, use automated functions in their everyday life mainly in order to save time. It is obvious that making use of mobile networks will be life-changing and a great improvement in the driving aspect.

1.3 VoX

V2X stands for 'Vehicle-to-Everything and refers to passing information from a vehicle to any other entity that may affect the vehicle and vice versa.[17]

Table 1: V2X technology includes:

Vehicle to Infrastructure (V2I)	The exchange of data between a car and equipment installed alongside roads.

Vehicle to Network (V2N)	Vehicle's access of the network for cloud-based services.
Vehicle to Vehicle (V2V)	The exchange of data between vehicles.
Vehicle to Pedestrian (V2P)	The exchange of data between the car and pedestrians

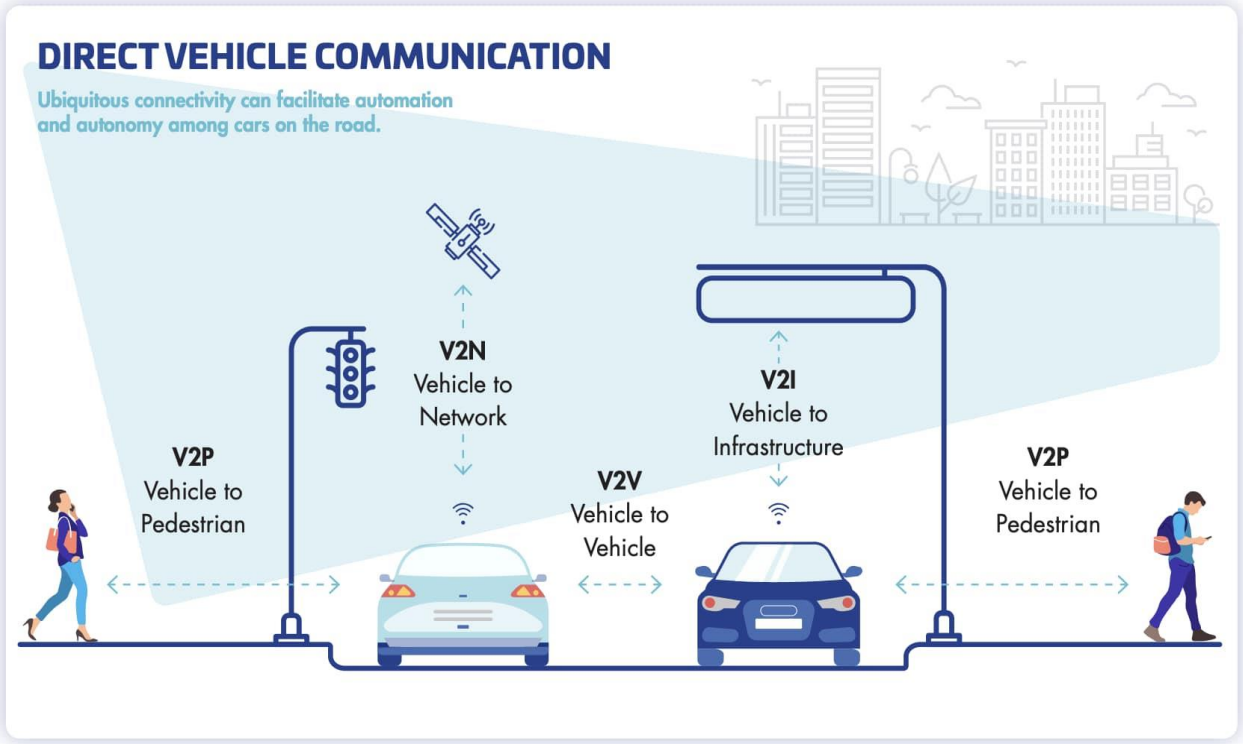


Image 2: All vehicle communication categories

The main motivations for V2X are road safety, traffic efficiency and energy savings. The U.S National Highway Traffic Safety Administration estimates a minimum of 13% reduction in a year [18].

2. RELATED WORK

The notion of QoS prediction has been studied in the literature in different contexts and focusing on different segments of the end-to-end communication. Numerous works have been proposed that attempt to predict the QoS –from diverse perspectives-, as well as several QoS-aware schemes, especially for V2X communications [19] [20] [21], but also in the domain of mobile networks, from a broader perspective [22] [23] [24] [25].

In [17], the authors focus on a MEC-enabled architecture for evaluating two V2X applications, namely, Advanced Driving (safety-related application) and Emergency Brake Light. Based on collected measurements, they focus on a classification problem, targeting to predict threshold-driven QoS classes. A Neural Network (NN)-based approach is proposed, combined with a Maximum Dependency (MD) algorithm for feature selection. The authors focus on predicting the expected end-to-end delay and the results are compared to other ML solutions, namely a Recurrent Neural Network with Long Short-Term Memory neurons, a Random Forest, and a Support Vector Machine. According to the authors, no significant prediction accuracy gains are observed in the alternative solutions, able to justify the noticeable increased cost of training, compared to the much simpler NN.

In [20], the authors employ supervised learning, as well as the auto-regressive integrated moving average (ARIMA) models. The specific work considers a typical urban scenario for C-V2X communication, namely Manhattan Grid [26]. The authors collect numerous radio-related measurements such as the reference-signal-received-quality/power (RSRQ/RSRP), the signal-to-noise ratio (SINR), the channel quality indicator (CQI), the user past averages throughput, delay, etc. and use them as inputs towards calculating the prediction accuracy and f1-score metrics. Although in most scenarios the ARIMA model performs relatively satisfactory, for a high number of UEs, the accuracy of the model drops considerably.

In [21], a latency-prediction framework is described, tailored for delay-sensitive V2X applications. The proposed scheme integrates ML, i.e., a Long short-term memory (LSTM) network, along with a k-medoids clustering algorithm to predict data that follow a trackable trend over time, with statistical approaches, i.e., a combination of Epanechnikov Kernel and moving average functions for predicting data that behave like random noise. The evaluation shows that the specific approach reduces the prediction error to half of a standard deviation of the raw data.

In [22], a predictive energy-efficient scheduling scheme is proposed, that optimizes the user equipment (UE)'s bits/joule metric subject to QoS constraints in downlink orthogonal frequency-division multiple access (OFDMA) systems. Hammad et al. were able to achieve that by minimizing the number of wake-up transmission time intervals, where the UE receiver circuit is ON, in a long-time horizon. The proposed predictive scheduler is supported by a ray-tracing (RT) engine that increases the scheduler's knowledge on users' characteristics long-term information. Authors in [23] studied the problem of application rate allocation over different radio interfaces, and addressed the issue of different delay requirements of applications using the discounted-rate framework. They propose two online predictive algorithms in order to handle the intermittence of secondary interface(s). The proposed algorithms' performance is presented consistently near-optimal using small prediction windows. Another work [24] performs prediction-based resource allocation focusing on application data rate; the specific work proposes the Threshold Percentage Dependent Interference Graph (TPDIG) using a Deep Learning-based resource allocation algorithm for city buses mounted with moving small cells. A comparative analysis of resource allocation approaches is presented, using TPDIG, Time Interval Dependent Interference Graph, and Global Positioning System Dependent Interference Graph, in terms of Resource Block (RB) usage and average

achievable data rate of mIoT-mSC network. Performance evaluation evidence is presented in order to confirm the gains achieved by the proposed contribution. In [25], a Cognitive Neural Network Delay Predictor for high-speed mobility in 5G C-RAN cellular networks is proposed, for compensating the transmission and acquisition delay of the Channel State Information working simultaneously, along with the conventional prediction technique for predicting the time variations of the communication channel. The results demonstrate a significant enhancement in the data rate of the network with the proposed approach.

Apart from the application of QoS prediction on mobile networks, extended research has taken place as well on other network service and application perspectives, highlighting the use of neural networks [27] [28] [29] [30]. In [27], a probability distribution detection-based hybrid ensemble approach is proposed, in order to achieve high prediction accuracy for QoS-Aware web service recommendations. Specifically, Li et al. propose an enhanced collaborative filtering (CF) based approach as the basis of the prediction model. The authors propose a distribution detection algorithm in order to calculate the probability confidence weights, based on the results of a set of other basic prediction models. Zhou et al. in [28] propose two neural models for the task of spatio-temporal context-aware QoS prediction, by considering invocation time and multiple spatial features both on the service-side and the user-side. The presented evaluation indicates that the models achieve a performance improvement by 10.9–21.0% in terms of Mean Absolute Error (MAE) and Normalised MAE in comparison with the baseline methods. The work in [29] presents a deep neural model, intended to achieve multiple QoS predictions based on context. It provides a framework to realize multi-attribute QoS prediction, and it manages to achieve high prediction accuracy in terms of MAE; the specific work also includes strategies that achieve substantial results in making use of contextual information. In [30], Yin et al. propose a method of combining an auto-encoder with CF, model-based CF and neighborhood-based CF approaches. The proposed auto-encoder deals with sparse inputs by pre-computing an estimate of the missing QoS values and obtains the effective hidden features by capturing the complex structure of the QoS records. In addition, authors propose a novel computation method, based on Euclidean distance, that aims to address the overestimation problem, to further improve prediction accuracy. Finally, they propose two models to produce the final QoS prediction results from user side and service side respectively, based on a real-world dataset that verifies the effectiveness of their method. Due to the high impact of cloud computing in the field of scientific and business technology domains, QoS Prediction has drawn the attention of researchers in this domain as well [31] [32] [33] [34] [35] [36]. In [31] Li et al. collect real cloud computing environment data, obtain the correlation between the hardware/software resource data and QoS attributes of the Cloud service and propose a novel QoS prediction approach based on Bayesian Network Model. In [32], Chen et al. propose a self-adaptive resource allocation framework composed of feedback loops, each of which goes through a designed iterative QoS prediction model and a Particle Swarm Optimization based run time decision algorithm. The work in [33] proposes a model for predicting end-to-end QoS values of cloud-based software solutions composed of services from multiple cloud layers. It relies on the internal features of services and end users such as location, network configuration and user profiles, in order to calculate service similarity. In [34], the authors attempt to predict the QoS by utilizing the historical QoS records of similar users on the Internet. A novel approach that combines a clustering-based algorithm and trust-aware collaborative filtering (CF) is proposed, aiming to predict the accuracy and recommendation quality. In [35][34], the authors propose a Matrix Factorization based approach for making context-aware QoS-prediction of specific cloud services. Luo et al. in [36] present a novel data-driven QoS prediction scheme using Kernel Mean Least Square (KLMS) for the purpose of achieving a higher accuracy. Via the trained KLMS they can predict the unknown QoS entries with their corresponding relevant QoS values.

As network management and QoS support are becoming more and more challenging with the increase in network traffic, size, and service requirements, a lot of research has taken place in order to develop ML-based models to meet these challenges [37] [38]. In [37], Vasilev et al. utilize ML in order to demonstrate how QoS metrics can be exploited to accurately estimate and predict key QoE factors. They propose a Bayesian Network model to predict the re-buffering ratio and then they derive their own novel Neural Network search method to prove that the Bayesian Network correctly captures the discovered stalling data patterns. They show that hidden variable models based and context information boost performance for all QoE related measures. Lastly, in [38], Lekhala et al. propose a software-defined and ML-based intelligent QoS framework called PIQoS, which pushes link failure recovery at the data plane in order to improve the delay and throughput. The proposed work offers two supervised ML models for efficient network state diagnosis and respective management policies selection.

As it can be inferred from the above state of the art analysis, a plethora of existing proposals on QoS prediction is already present, that focuses on diverse network aspects. The proposed work primarily relates to [17] [20] [21], and secondarily to [22] [23] [24] [25], which although tackle similar research questions, do not focus on V2X and/or CAM scenarios, which is the key use case of our work.

The work in [17] firstly tackles the problem as a classification problem; in our case, we attempt to avoid handling the Key Performance Indicator (KPI) metrics as discretized variables (QoS classes), but rather as a regression problem, in order to retain a more fire-grained approach. In [20], the authors employ a simple moving ARIMA, which fails to cope with complex scenarios, high number of UEs, etc. The proposed scheme in [21] is the closer work in terms of design decisions, combining LSTM with clustering approaches; nevertheless, complexity analysis - which is tackled in the current paper- is completely missing, while the evaluation scenario makes at no point any links to any realistic V2X application or use case, contrary to the proposed work, which provides a detailed simulation environment for the ToD use-case.

Furthermore, the work in [22] is limited by the selection of the RT-based prediction technique, which relies solely on the physical layer, and more specifically on the wireless communication channel propagation characteristics. The work in [23] focuses on the rate allocation problem as a convex optimization problem. It does not employ or validate any ML algorithms and focuses on the optimization of rate allocation from the scheduler’s perspective. The specific work focuses on the management and offloading of flows with different delay and rate requirements among different radio interfaces of a user device. Also, the predictability of wireless connectivity is realized for a small look-ahead window, while our work is capable of extending the prediction window ahead, as the volume of the training data increases over time. The work in [24] does not attempt to directly predict QoS-related metrics, which may exhibit considerably unexpected behavior, but locations of vehicles (road segments), which are afterwards correlated to determine the experienced interference. Moreover, only physical layer aspects are taken into account. Last but not least, the work in [25] does not perform direct QoS metric predictions; instead, it focuses on the acquisition delay in the Channel State Indicator metric, which is indirectly linked to the variation of the communication channel conditions.

In this work, we extend the current state of the art, by proposing a novel predictive QoS scheme, tailored for 5G CAM use cases. To our knowledge, predictive QoS in CAM and V2X scenarios is still in a very primitive stage, with very limited prior work that attempts to exploit AI and ML approaches towards predicting the QoS for vehicles in 5G and beyond scenarios. As a result, this is the first predictive QoS scheme for V2X communications, that takes into account the latest 5G standardization guidelines and implements an end-to-end solution towards the proactive notification of connected vehicles, in a realistic CAM scenario.

Most importantly, this is the first work that provides in detail a computational complexity and network overhead analysis, assessing the viability of the proposed framework. Additionally, this is the first QoS prediction scheme that relies on a dynamic and flexible map grid/cell clustering technique, - which takes into account the correlation of QoS behavior among cell clusters -, towards minimizing the computing resources' utilization. The last feature is realized via a novel Map-as-a-Grid model, which is presented in the following section, along with the rest of the framework details and algorithmic aspects. Also, no inputs from other network segments that affect the end-to-end QoS are taken into account. Last but not least, this work attempts to explore the prediction performance and application-specific capabilities of a considerable number of ML algorithms and approaches, with and without deploying the proposed map-as-a-grid approach.

3. THE PREDICTIVE QoS MECHANISM

This work follows the concepts and terminology introduced by [15][13]. As already discussed in the introduction of this work, the ultimate goal of predictive QoS for CAM is to exploit in-network intelligence for the proactive calculation of the relevant QoS aspects for the specific V2X services that are each time active. The prediction is realized in a proactive manner and ultimately aims towards the generation and transmission of the QoS-related notifications messages to the involved vehicles, for taking the needed application-layer actions (service parameters adaptation, switching between automated-manual operation, etc.). The last part, related to the vehicle-side application adaptation aspects, is out of context of this particular work.

3.1 Input Parameters

The proposed predictive QoS scheme, namely PreQoS, is based on a Fusion Machine Learning approach, which is able to process contextual information from diverse data sources and different layers of the network (Table I), in order to predict with high accuracy, the expected QoS for different V2X services with diverse requirements; the QoS that is provided to the vehicles depends on different factors, namely the availability of radio and network resources, the environment characteristics (e.g., physical obstacles such as buildings, blind spots, etc.), as well as the mobility characteristics of the vehicles (e.g., a high velocity vehicle will potentially require consecutive handovers from the network, which will impact the QoS). The QoS metric that is studied each time is tailored in accordance with the specific V2X application requirements, namely uplink or downlink data rate, end-to-end delay, reliability, packet loss, etc.

On the one hand, in order to adequately assess the provided QoS for the respective V2X service and be able to accurately predict it for future time windows, the mechanism needs to correlate information from the different network segments, which comprise the end-to-end communication path. As also described in the respective 3GPP's study on application layer support for V2X services [31], this end-to-end communication path depends on the type of the V2X application/service, i.e., V2V, V2I, V2N or V2P. On the other hand, besides the air/link propagation aspects, additional delay-introducing components contribute to the E2E network performance, such as device buffers, computing Virtual-Network-Functions / Physical-Network-Functions (VNFs/PNFs), backhaul link capacities, as well as core network components processing resources and load. The input context parameters, which can be utilized to train the ML model the proposed scheme is grouped in five main categories, as illustrated in the following table (Table I):

Table 2 The five categories of input context parameters of PreQoS

Category	Description	Input Metrics
Mobility Information	Vehicle/UE mobility-related data. The optional mobility information is only required in the case of Trajectory prediction of the UE.	Required: latitude, longitude and timestamp for each location. Optional: velocity and acceleration vectors, heading, predicted path, trajectory constrains (e.g., road limits), etc.
Radio Parameters	Radio-related, passive measurements provided via the UE-based measurement	Received Signal Strength Indicator, RSRP, RSRQ, CQI, SINR, client (Global Position

	reporting to the gNB/eNB. These metrics are complementary, in order to enhance the prediction.	System, velocity, heading), geolocation map.
RAN / facilities layer	Latency-introducing network components related to queues, computing resources' capacity, etc.	Roadside unit (RSU)-related network load, RSU-queues load, Number of UEs associated to Base Station, availability of MEC/cloud computing resources.
E2E network performance	Network measurements related to the E2E service, including transport and core network measurement information.	Transmission/queueing delays, backhaul link capacities, VNF processing delays, etc.
Application- specific information	Tailored, application-oriented information that influences the performance of V2X service (e.g., V2X group- based communications, such as platooning).	Service priorities group / cluster-based communication type, cluster head nodes, etc.

It is highlighted that the above table illustrates the information item types that can be processed during the offline training process, if available. The operation of the described algorithm follows this flexible approach enabling the processing of the data and the generation of the respective prediction, depending each time on the specific network deployment, interfaces, and available real-time data sources, which often comprise only a sub-set of the afore-presented information items.

3.2 System Model

We consider a prediction communication system, in which a number of $u \in U$ mobile users (i.e., vehicles) are consuming a number of different V2X services $s \in S$ and are notified by the network about a predefined set of QoS-related KPIs $k \in K$ such as uplink/downlink data rate, packet error rate, or end-to-end delay. Hence, each time the system must perform a prediction for u x k KPIs, which are then processed by the respective V2X applications for possible required adaptations.

The prediction approach can be defined to be performed in two possible ways: a) in a predefined (according to the V2X service specific requirements) periodic manner, b) upon change of the predicted QoS values/value classes (i.e., as the user moves along a path with heterogeneous radio propagation characteristics, or the network conditions change -for example a high number of new vehicles enter the specified area).

We assume that the actual computation tasks are performed by the 5G Core Network's Prediction Function (PF), which is assumed to be a module, part of the wider NWDAF [2]; the PF can be deployed either at a MEC server or a cloud center, as part of the rest of the Core Network functions and entities. In the case of a MEC system, the delays and packet losses in backhauls and core networks are considered equal to zero, according to the input parameters modelling presented in the previous subsection.

1. Geographical space as a grid: The considered geographical space (map) is modelled based on grid-tile approach, namely Map-as-a-Grid (MaaG), comprising rectangular cells towards

applying the prediction in a discrete manner (Figures 1a and 1b). It should be highlighted that from now on, the term grid cell is not to be confused with the base station notion of cells, in the context of cellular networks; grid cells will refer to the geographical grid-based model of the proposed algorithm. Based on a recursive QoS metric assessment (i.e., SINR, ul/dl average delay or data rate, etc.), mapped to the discrete map grid cells, the second step is the clustering of grid cells, in a way that the clusters demonstrate similar behavior in terms of QoS metrics for all the vehicles/UEs' QoS measurements in the specific clustered cell. The detailed algorithmic steps are presented in the next subsection.

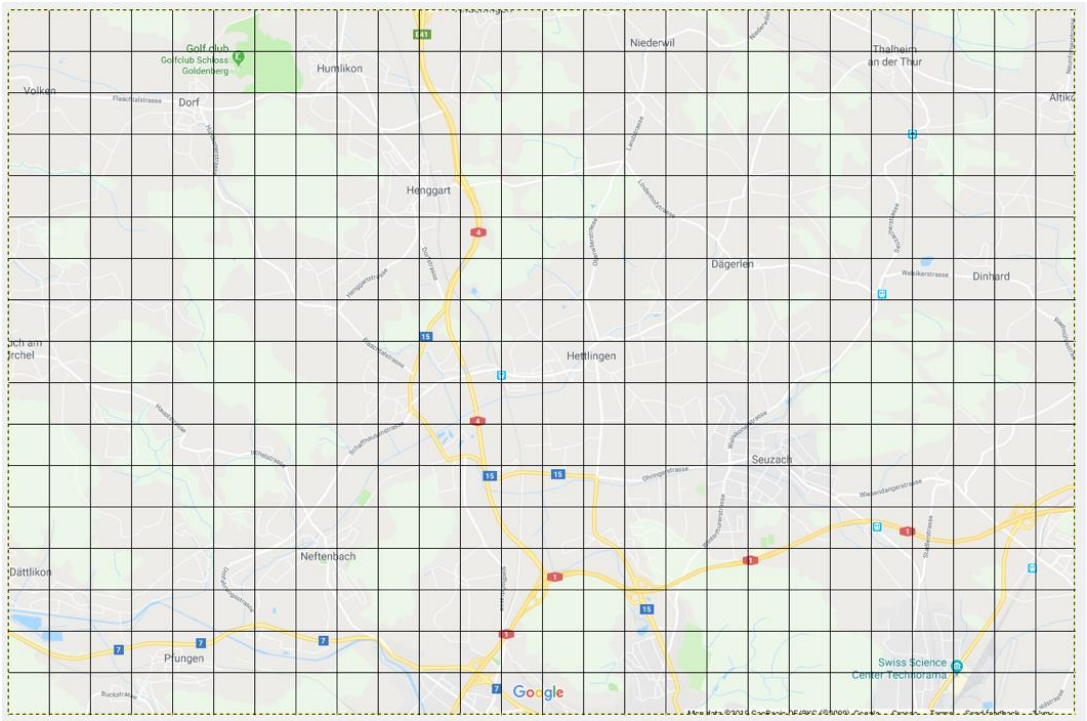
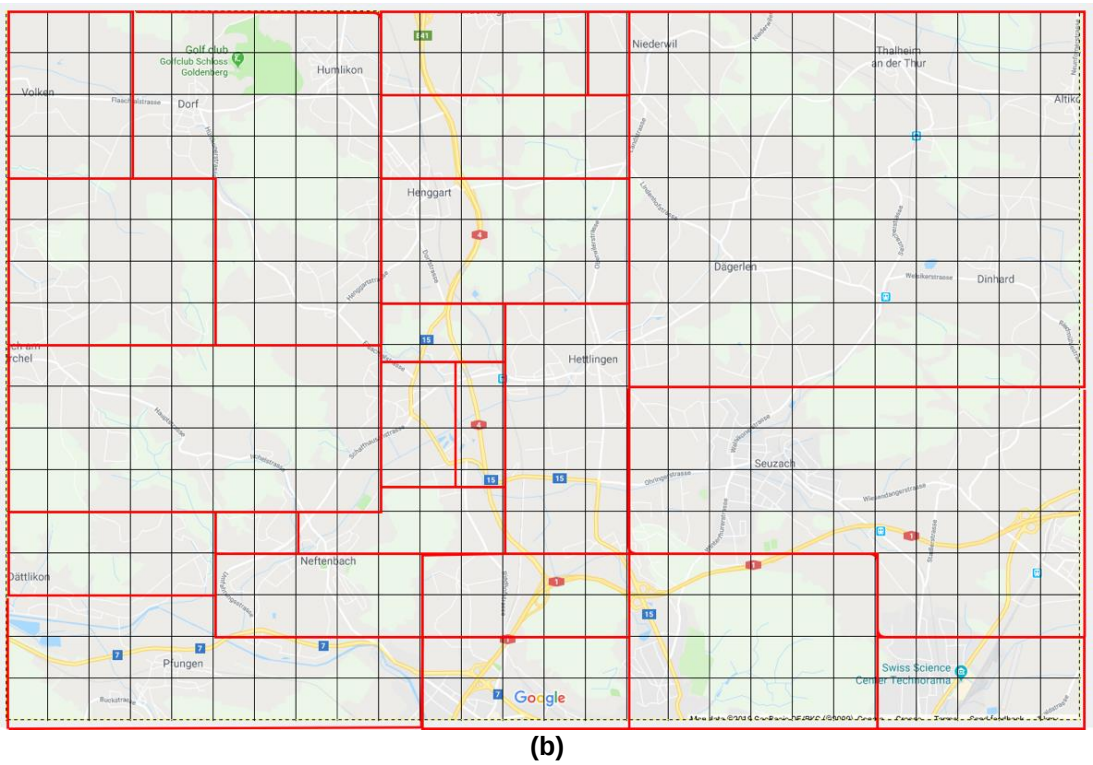


Figure 1. (a) MaaG – Step 1: Initial grid modeling.



(b)

Figure 1. (b) MaaG – Step 2: Grid after the QoS-based correlation clustering.

The major advantage of this approach is the minimization of the computational overhead, as the prediction is applied in a discrete manner, i.e., per cluster, rather than for the continuous coordinate system's space or single users. In other words, the geolocation information (i.e., latitude, longitude) of each measurement is transformed via the MaaG mapping to one specific grid cell or grid cell cluster. This results in a reduction of the required computing tasks from u x k predicted values, to c x k, where $c \in C$, the defined clusters of the system and $c \leq u$.

2. Time-based modeling: The time is modeled using periodicity features based on the rationale that -besides the network resources' availability or the radio propagation characteristics- the QoS is also radically influenced by the actual users' load, i.e., the road traffic volume and density characteristics, which in turn relate to specific weekly vehicle mobility patterns (i.e., weekdays, weekends, etc.). This approach relies on the intuition that the service traffic-related data follow a seasonal pattern on a weekly interval and therefore during training (Figure 3) it is able to capture that seasonality. Moreover, the model can be flexibly adapted in order to perceive other types of seasonality characteristics, such as annual or monthly, which can be taken into account (e.g., different road traffic patterns during Holidays). Furthermore, specific vehicle volume/mobility characteristics can be identified on single day-basis, meaning that the traffic follows specific patterns on a daily basis (rush hours during morning, lower traffic during night, etc.). To this end, the time dimension of the prediction horizon is discretized into T slots of a predefined duration, namely QoS window, for which the QoS metrics of a specific grid cell exhibit a low to near-zero variance; the QoS window is considered as the prediction horizon for each single prediction. An example value with fair granularity for the QoS window for a single cell/cluster could be defined at 1 minute with a weekly seasonality; this translates to 60 min x 24 hours x 7 days = 10,080 slots for a weekly-based prediction model. The second step is to normalize the time dimension, depending on the periodicity of the model to be generated (e.g., time is normalized from 0 to 1 for one-week duration using min/max normalization or standardization). It should be highlighted that the described prediction horizon refers to the temporal length of a specific prediction model for a specific Grid Cell and is different from the prediction granularity, which is at the level of ms.

Based on the aforementioned analysis, regarding the seasonality of the data, the interval could be daily, weekly, monthly, annual, etc. In order to choose the appropriate interval, during data pre-processing i) the interval that suggests a cyclic pattern should be chosen and ii) the training data must be sufficient for each timestamp.

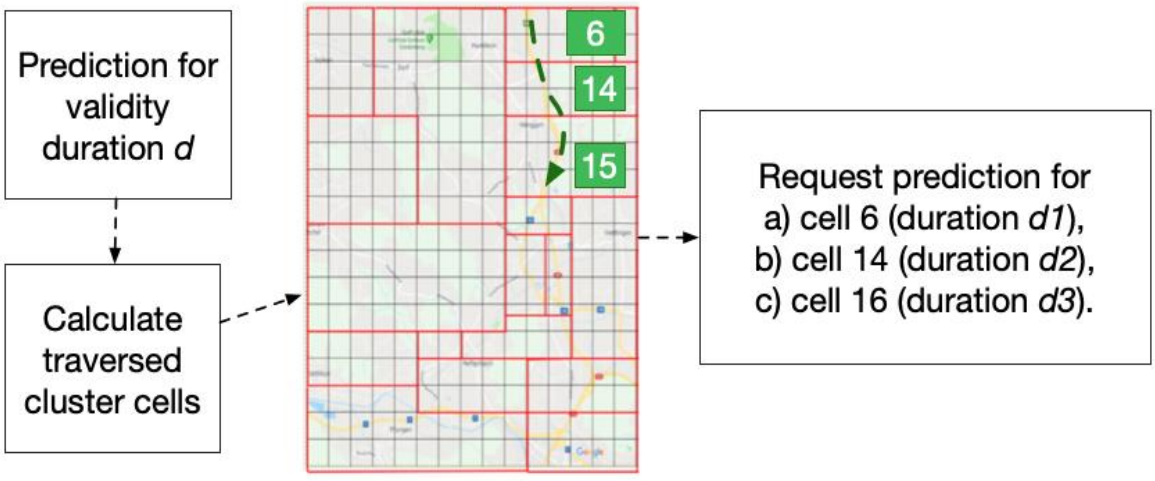


Figure 2 Prediction validity duration via calculation of traversed cells

According to the validity duration requested by each specific V2X application, the predicted data rate/latency/packet error rate values are extracted based on a one-to-one mapping, depending on the cluster cells that are traversed during this time duration. The traversed cells are computed based on the vehicle's position and mobility characteristics (i.e., velocity and heading). Figure 2 illustrates an example of the afore-described concept. Let us denote prediction validity duration, for which the vehicle path comprises three adjacent cells and where d = d1 + d2 + d3.

3.3 PreQoS Algorithmic Framework

The workflow of the PreQoS algorithm is described in Figure 3. The main algorithmic steps comprise the training of the data, the application of the regression model, the cell clustering and the extraction of the respective ML model

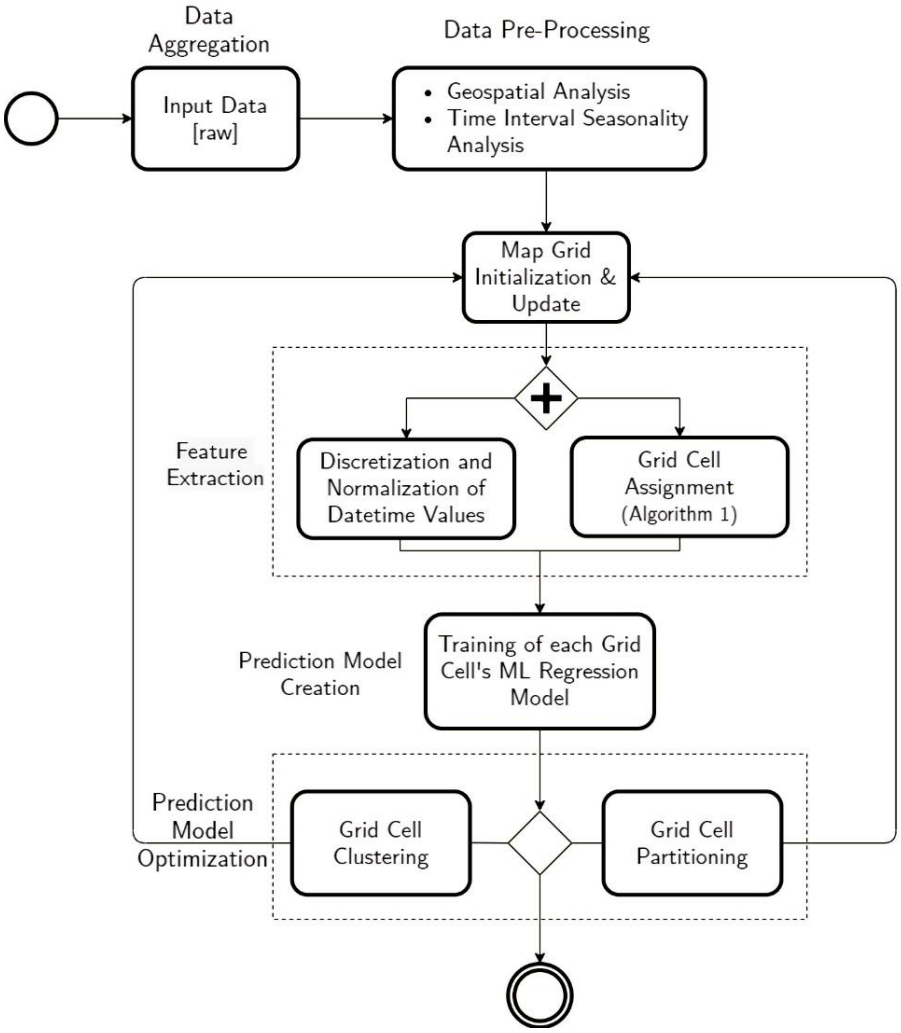


Figure 3. Overview of PreQoS Algorithmic Framework

$num\_cells \leftarrow num.\ rows/columns\ of\ the\ Grid$
procedure GET_CELL_ID(lat, lon)
    $tmp = \frac{max\_lat - min\_lat}{num\_cells}$
    $x = \left\lfloor \frac{lat - min\_lat}{tmp} \right\rfloor$
    $tmp = \frac{max\_lon - min\_lon}{num\_cells}$
    $y = \left\lfloor \frac{lon - min\_lon}{tmp} \right\rfloor$
    $cell\_row \leftarrow (num\_cells - 1) - x$
    $cell\_column \leftarrow y$
    return $C_{x,y}$ ▷ The Grid Cell Id of the sample
end procedure
for *sample* s `in Dataset` **do**
    $s.cell\_id$ =`GET_GRID_CELL_ID` $(s.lat, s.lon)$
end for

Figure 4. Algorithm 1 Latitude and Longitude to Grid Cell Id

As illustrated in Figures 3 and 4, the first steps of the algorithm comprise i) the aggregation of the different input data types, as presented in Table I, as well as ii) the pre-processing of the input data, where a geospatial and a data seasonality analysis is performed, towards determining the Grid Size (num. of grid cells) and the time interval respectively.

As a part of the PreQoS workflow, Algorithm 1 presents the steps for determining the grid cell ID of a sample, given its latitude and longitude. The grid cell ID follows a two-dimensional array-modeled indexing [row, column] in ascending order [top to bottom, left to right]. Each unique identifier of a geographical cell is used in order to intuitively perceive the location of a Grid Cell in the Geographical space, as well as, generate a Regression model for each cell and save it to memory. At this point, the samples contained within a single Grid Cell are used as input for the ML model of this specific Cell, as shown in Figure 5. Given that the spatial aspect of the data has already been captured, by the MaaG scheme, the timestamp of each sample is used as the independent variable input for each Regression Model, with each QoS metric being the dependent variable.

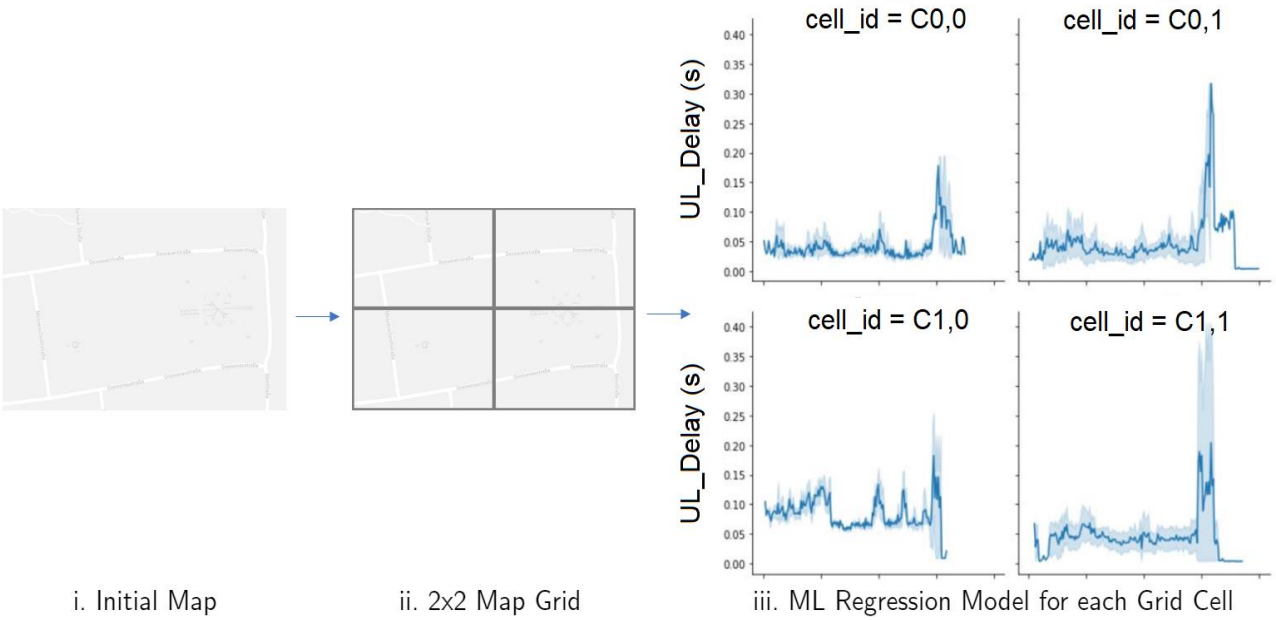


Figure 5. Regression Model for each Grid Cell

As it is described in higher detail in the Evaluation Section, a set of different well established regression ML models have been considered and compared. A confidence interval in the prediction is considered, in order to capture the uncertainty in our predictions (e.g., upper bound of an end-to-end delay prediction, lower bound of a data rate prediction, etc.). For the definition of the prediction confidence interval, the Quantile Loss Function (QLF) is employed (Equation 1), where Quantile-based regression aims to estimate the conditional “quantile” of a response variable given certain values of predictor variables. For example, if the requested Confidence Interval is 90\%, then the α (alpha) of the QLF in the ML model will be set equal to 0.1, in order to estimate the 10- and 90-quantiles.

$$Ł(\xi_i|\alpha)= \begin{cases} \alpha\xi_i & if\ \xi_i \geq 0, \\ (\alpha-1)\xi_i & if\ \xi_i < 0 \end{cases} \quad (1)$$

, where ξ_i is each sample’s real y value.

The last step of the algorithm is the Optimization of the Prediction Model in terms of Grid Cell Clustering and Partitioning (Figure \ref{ml1}). Intuitively, a larger number of Grid Cells results in higher granularity and higher prediction accuracy; at the same time, this results in a larger number of Grid Cell models, which must be accommodated in memory, leading to performance deterioration. Cell Clustering is performed, as a final step of the proposed algorithm in order to evaluate the afore-described trade-off; detailed results are presented in the Evaluation section. This means that cells with correlated QoS behavior over time are clustered together into a Grid Cluster. Pearson correlation is used (Equation 2), in order to measure the linear correlation between two time series, where if the absolute value of the Pearson correlation coefficient (PCC) is greater than a predefined threshold (e.g., 0.85, 0.90, 0.95 etc.), the cells now reference to the same ML model, thus forming a cluster. The choice of the correlation threshold value, is determined during analysis, requiring that the average accuracy error doesn't increase significantly, given the trade-off with the memory and CPU consumption.

$$\rho = \frac{\sum_{i=1}^{n}(x_{ai}-x_{bi})\cdot(y_{ai}-y_{bi})}{\sqrt{\sum_{i=1}^{n}(x_{ai}-x_{bi})^2}\cdot\sqrt{\sum_{i=1}^{n}(y_{ai}-y_{bi})^2}} \quad (2)$$

, where we calculate the PCC between two different Grid Cell Regression models a) and b). Moreover, the sample size value n is an equidistant sequence in the x axis, derived from the predefined time interval, with each y_i being the QoS metric prediction for the specific x_i.

The Grid Cell Partitioning is also illustrated in the context of the Prediction Model Optimization step in Figure 3. In the same rational with the Grid Cell Clustering, if a Grid Cell is not able to fully capture the spatial aspect of the Data, the Variance in the QoS features within the Cell will be very high. Therefore, the coefficient of determination ($R^{2)}$ for each ML model is calculated, and if R^2 is lower than 0.5, the cell is further partitioned into 4 sub-cells, in order to increase the robustness of the input data to the respective ML models.

4. EVALUATION

In order to evaluate the performance of the proposed mechanism, we implemented a real-world simulated mobility scenario using Simulation of Urban Mobility (SUMO) [39], which is an open source, highly portable, microscopic and continuous traffic simulation package designed to handle large networks. It allows for intermodal simulation including pedestrians and comes with a large set of tools for scenario creation. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center.; afterwards, the extracted mobility patterns were imported into NS-3 discrete-event Network Simulator [40], which is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. NS-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use. NS-3 was used for performing a complete, end-to-end communication scenario for a specified time duration, exploiting the 5G mmWave module, introduced in [41]. The simulated geographical area that was selected for the performed evaluations is located in Munich, Germany near the Huawei Munich Research Center.

In Figure 6a, we present the real location in Munich as it has been retrieved via Google Maps, along with the specific deployment location of the two Base Stations (BSs), while in Figure 6b the SUMO-based transformation into the virtual scenario is illustrated. Overall, Table 3 presents a summary for all the parameters used in the mobility simulation.

Table 3. Parameters Used in the Mobility Simulation Environment

Number of UEs	Velocity Range (km/h)	Acceleration (m/s)	Deceleration (m/s)	Scenario Duration (s)
50	[0,20]	0.8	0.8	200

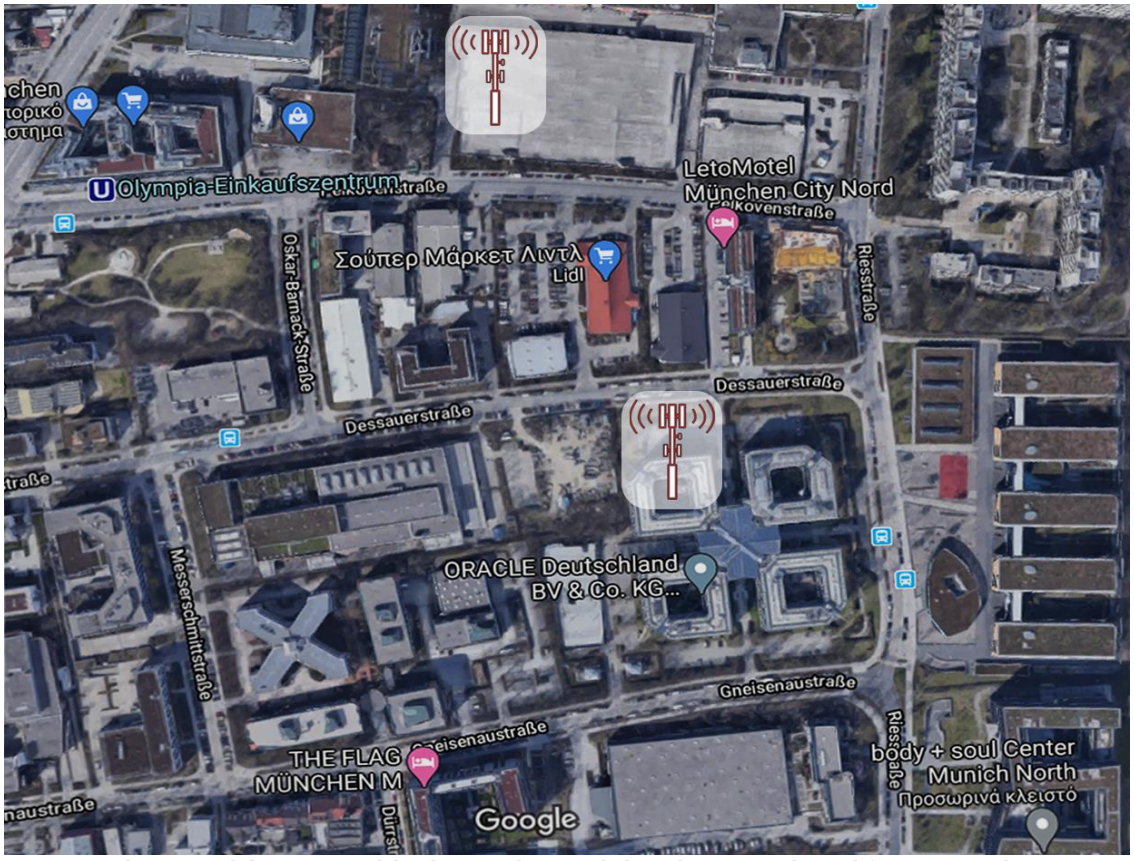


Figure 6. (a) Geographical Location as it has been retrieved from Google

Figure 6. (b) The Geographical Location converted in SUMO mobility simulation tool

ToD is the V2X use case that has been modeled in the specified simulation scenario. The goal of the ToD use case is to enable a remote driver to remotely control a vehicle in the case that the driver of the vehicle cannot drive in an efficient and safe manner e.g., due to a health issue or in the case that an autonomous vehicle may detect a situation that's uncertainty is high and cannot make the appropriate decision for a safe and efficient maneuver. At the uplink interface the vehicle provides to the remote driver video streams of high quality and status information of the HV. The Remote Driver based on the received information builds her situation awareness and taking into account the destination point selects the maneuver instructions. The vehicle receives from the Remote Driver (downlink) the maneuver instructions and adjusts its trajectory, speed, acceleration. Feedback is provided to the Remote Driver in parallel with the execution of the maneuver. The uplink and downlink data rate of each remote-driven vehicle (UE) is 50Mbps and 500kbps, respectively.

The simulation scenario comprises of 50 different moving ToD UEs, with variable speeds in the range [0, 20km/h] with 0.8m/s of acceleration and 0.8m/s of deceleration. The duration of each executed simulation is 200 seconds and the sampling frequency of QoS data is 1 Hz. The UEs experience different channel conditions according to their line-of-sight/non-line-of-sight (LOS/NLOS) positions and respective distance from the BSs, while each moment being associated to a single BS. Horizontal, X2-based handover is enabled in the scenario, based on the RSRQ measurement reporting of the UEs. Overall, Table 4 presents a summary of the simulation configuration parameters.

Table 4. Parameters Used in the NS-3 Simulation Scenario

Parameter Description	Default Value
5G NR / LTE Scheduler	Propotional Fair Scheduler
5G NR / LTE eNBs' Height	24m
UEs' Height	1.6m
Uplink Data Rate	50 mbps
Uplink Packet Size	1400 bits
Downlink Data Rate	500 kbps
Downlink Packet Size	1400 bits
5G NR Frequency Used	28 GHz
LTE Uplink Frequency	1920 MHz
LTE Downlink Frequency	2110 Mhz
Service Level Latency	40 ms
LTE Transmission Power	46 dBm
5G New Radio (NR) Transmission Power	30 dBm
UEs' Transmission Power	20 dBm
LTE Downlink and Uplink Bandwidh	20 MHz
5G NR Downlink and Uplink Bandwidth	1 GHz

The metric that was selected for the evaluation of the proposed algorithm is the downlink end-to-end delay, which is a crucial QoS KPI for the successful realization of the ToD use case [41]. Initially, a comparison was performed between a low and a high network load scenario-in terms of associated vehicles/UEs-, in order to assess the relative load for the particular environment and network set up, and how each selection influences the downlink delay KPI. As it is shown in Figure 7, in the case of a low load scenario with 5 ToD UEs low and stable downlink end-to-end delay is observed due to the system's abundance of resources. In the case of a high load scenario, where 50 ToD UEs are driving there is an increase of the observed downlink delay. An accurate and early prediction of an expected increase of the downlink delay is important for the efficiency of the ToD service, since this will enable an efficient adaptation of the ToD application and/or of the network side. The high load scenario is used for the rest of the evaluation section to show the prediction performance in as much realistic and challenging conditions as possible.

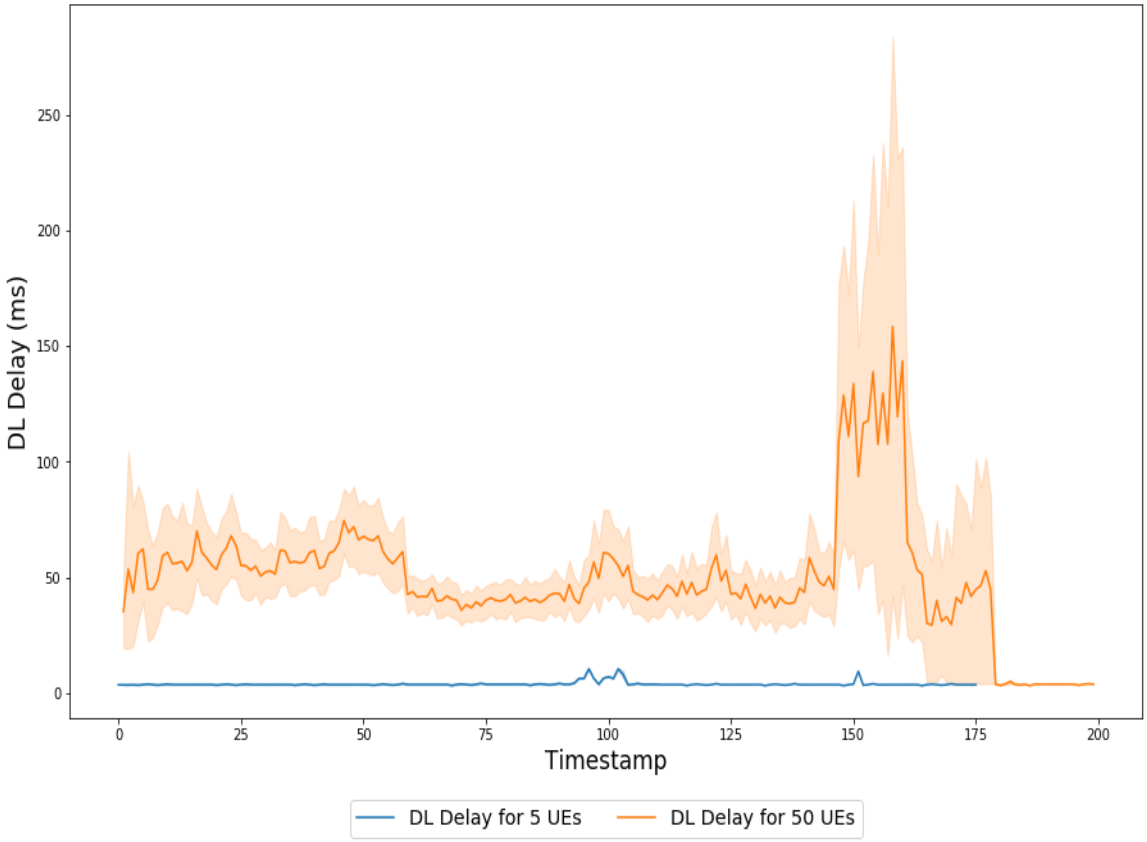


Figure 7. Average system downlink end-to-end delay (ms) for the two scenarios (5 and 50 UEs respectively)

In the rest of the section, we present the outcomes of the evaluation of the aforepresented prediction scheme, differentiating between two main ML algorithm groups: the one is described as non-MaaG, i.e., an approach, in which we do not employ the Map-as-a-Grid spatiotemporal algorithm presented earlier, and which comprises two different ML models, namely a Random Forest Regressor Model (RF-R) and a K-Nearest Neighbor Regressor (KNN-R); in this first group, the spatial input follows a simple map coordinates format (timestamp, longitude, latitude). The second approach follows the MaaG model, and three different ML algorithms are applied on top, namely a Deep Neural Network (DNN), a Gradient Boosting Machine (XGBoost), as well as a Support Vector Regression (SVR) model. Table 5 presents an overview of the training parameters that were used for each one of the above ML models.

Table 5. ML Algorithms Configuration

Algorithm	Training Parameters
RF-R	Number of Trees (Estimators): 300, Features per split: 3 (latitude, longitude, timestamp), Maximum Depth of a Tree: 110
KNN-R	Number of Neighbors: 5, Weight function: Distance
DNN	Layers 64x64x32x1, Activation Function Relu, Loss: Quantile Loss function, Training Parameters: Optimization Adam, Initial Learning Rate: 0.01, Epochs: 150, Bath size: 150
XGBoost	Estimators: 1550, Depth: 7, Learning Rate: 0.001, Sample Threshold: 100, Loss: Least Squares
SVR	Polynomial Transformation of data: Degree 7, Kernel: RBF, C penalty: 0.01

The Accuracy Error, which is illustrated as the primary evaluation metric in the following figures refers to the prediction accuracy of the downlink delay KPI, using the Quantile Loss

Function with alpha = 0.5, which we define employing the Mean Absolute Error (MAE) metric:

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (3)$$

, where n is the number of samples in the testing dataset, y_i is the actual value of the metric (e.g., DL delay) and y_i is the predicted value from the ML Regression model. The Accuracy Error is calculated as the average of the total Grid Cell MAEs for a specific Grid.

Towards evaluation, -and for all the results that are presented henceforth- we apply a 10-fold cross-validation method for each grid cell's ML Regression Model; the results from the folds are then averaged, to produce a single estimation for evaluation metrics such as prediction accuracy (error), F-test and the coefficient of multiple correlation R^2, which is explained in higher detail in the following paragraphs. The prediction accuracy from the 10-fold cross-validation of each model is used, in order to find the optimal parameters of each ML Model, as shown in Table 5. Moreover, the accuracy error of the following figures refers to the prediction accuracy of the downlink delay KPI, measured in milliseconds (ms), using the aforementioned MAE loss function. Finally, the dataset is shuffled between each fold, in order to test points spanned across the various time interval, given that the model has to be fitted adequately for Interpolation, with no need for adequate extrapolation capabilities.

The data used for the testing of each model, consist of a chunk of data that the model had never seen before during the training phase. Using the 10-fold cross-validation method, the testing data was different in each of the ten iterations, having the overall error calculated by taking the average error (measured by the MAE loss function) of all iterations. This process is then repeated for every Grid Cell's ML Regression model, where finally, the average of all the trained models' errors results in the final evaluation value.

In order to assess the behavior of the map grid modeling, we perform as a proof-of-concept six different grid cell models (i.e., number of correlated prediction location cells), applying different spatial granularity of the prediction models.

Figure 8 illustrates the average DL Delay prediction accuracy error, based on the averaged MAE values, calculated for each Grid Cell, for each MaaG-enabled ML regression model, for different Grid size options. The dataset size of this experiment is 50.000 samples. The number of Grid Cells is initialized with 1 (meaning no grid at all in this case) and it is increased gradually to a 6x6 Grid (i.e., 36 Grid Cells); accordingly, it can be observed that the average error value decreases in an exponential manner from ~ 9 ms (due the high variance in the data, by not capturing any geo-spatial information within the data) down to ~ 2 ms for Grid size = 36. This is the direct result of the geospatial aspect of the samples being taken into account, mitigating the overall prediction error. All three ML models exhibit an almost identical performance.

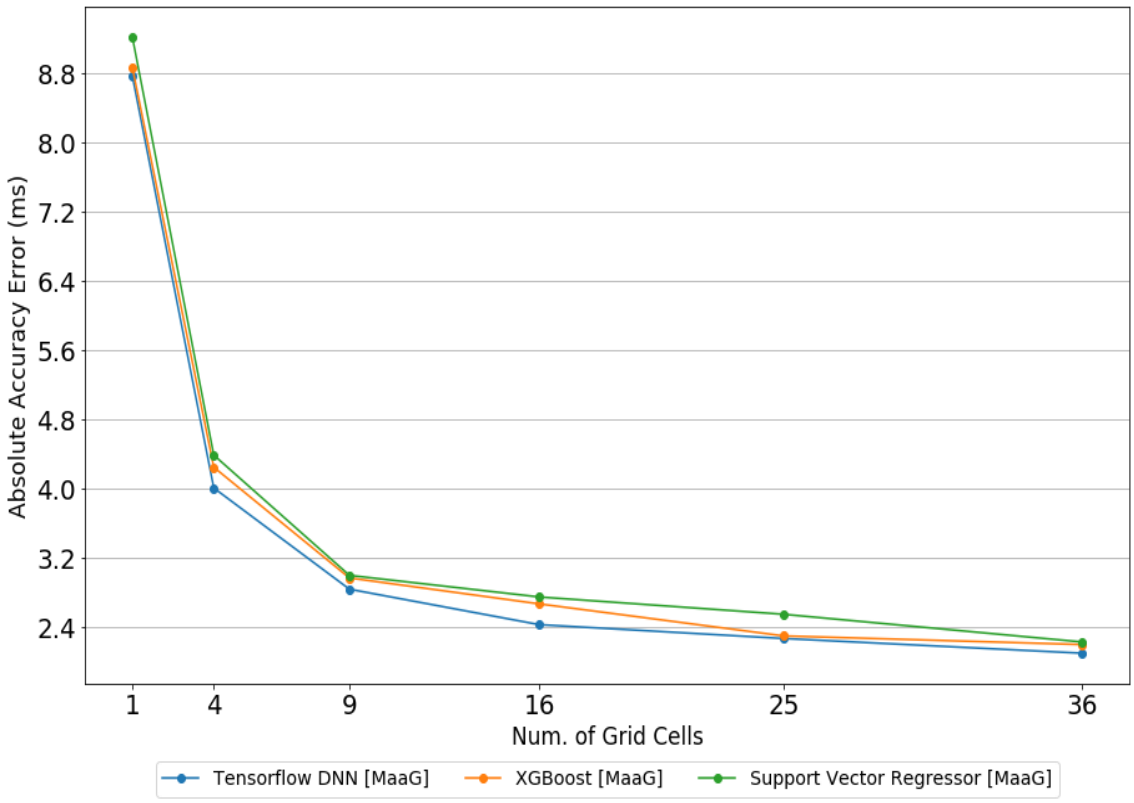


Figure 8. Absolute Accuracy Error based on the number of Grid Cell

Figure 9 illustrates the evaluation results for the aforementioned five different algorithms (non-MaaG- and MaaG-based), indicating the mean absolute error (MAE) of the DL delay QoS metric, in ms. The Grid size for the MaaG-featured experiment is 16x16, hence generating 256 Grid Cells and regression models respectively (each one per single Grid Cell); the Error illustrated in the y axis is calculated as the average of the 256 MAE values of the afore described Grid configuration.

For smaller training data volumes available from the network, the non-MaaG algorithms exhibit a better performance; for very small samples of the order of 1k raw measurements, the MAE of the non-MaaG algorithms is 2.2, while the MaaG-enabled schemes exhibit an almost double MAE of 3.9-4.6 (mean 4.1); on the contrary, as the input training data volume increases, the MaaG algorithms' performance gradually increases; for sample sizes of 100k measurements or more, MaaG algorithms outperform the non-Maag, reaching an optimal MAE of 2.2 down to 1.6, for the MaaG-enabled DNN algorithm.

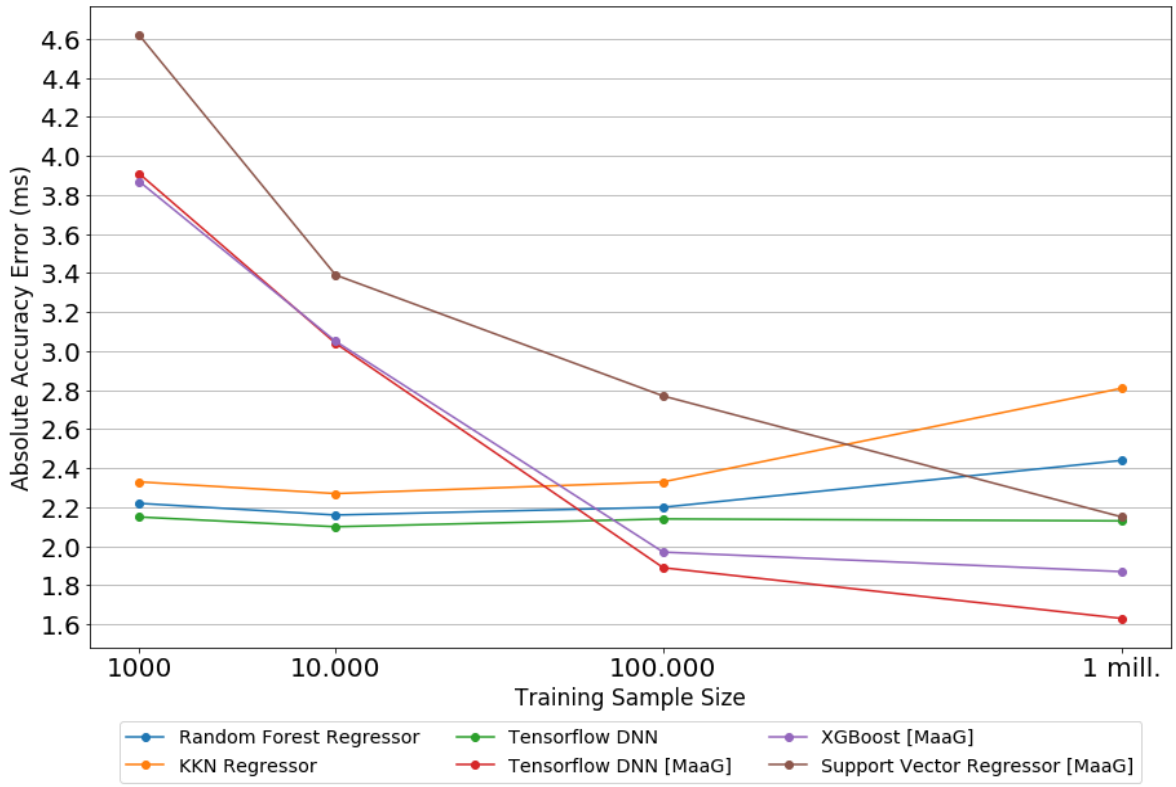


Figure 9. Absolute Accuracy Error per Sample Size

As mentioned earlier, the superiority of the MaaG model is exhibited, when applied on a sufficiently extensive sample size, in order to fully capture the spatial, as well as, the temporal aspects of the available dataset. The size of the dataset required, in order to minimize the error, is directly proportional to the size of the geographical space examined, as well as, the time interval chosen, based on the seasonality the model is trained to capture. Moreover, it is worth noting that the time interval chosen in this evaluation, is a daily interval, thus, in the absence of adequate samples in the temporal domain, the model needs to extrapolate to a considerable extent, whereas the model performs better for interpolation.

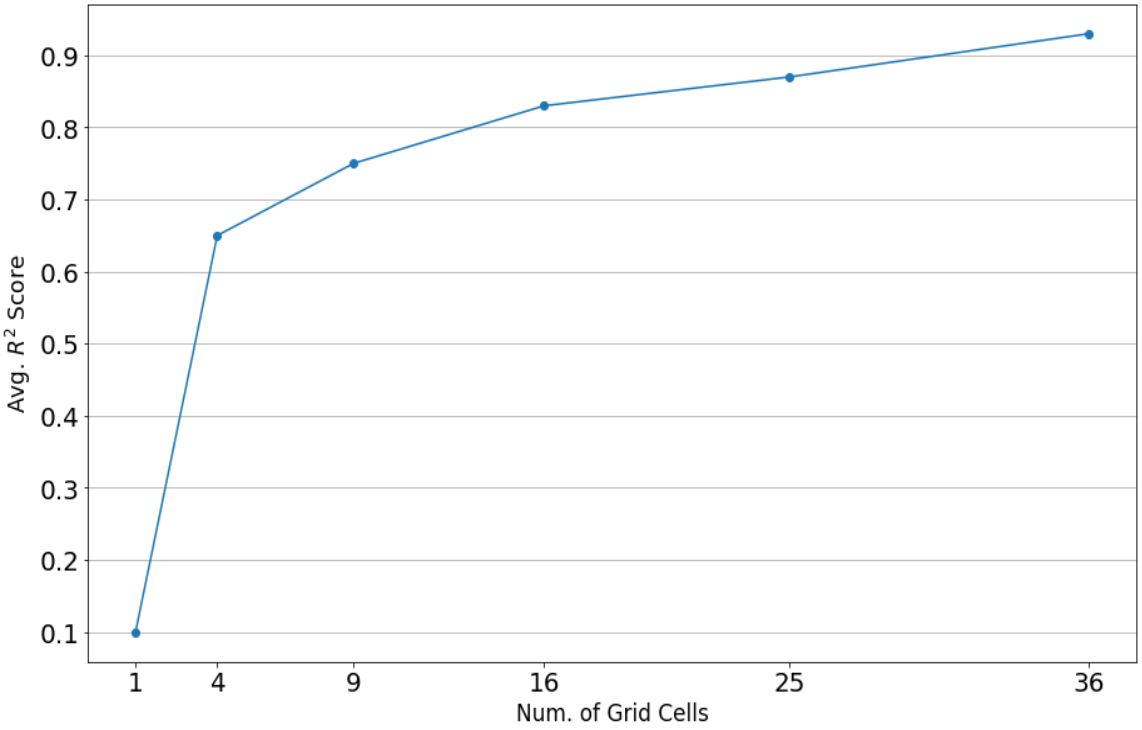


Figure 10. Coefficient of Determination based on the number of Grid Cells

The gains obtained by the MaaG model are shown when evaluating the coefficient of determination (Equation 4), or the coefficient of multiple correlation R^2 (Figure 9).

$$R = \frac{n \cdot \sum x \cdot y - \sum x \cdot \sum y}{\sqrt{[n \cdot \sum x^2 - \sum x] \cdot [n \cdot \sum y^2 - \sum y]}} \quad (4)$$

, where “n” is the number of samples, with each sum including the whole dataset size [0, n].

The specific figure illustrates how the R^2 metric increases, as the spatial granularity of the MaaG model increases as well, leading to more accurate prediction results. It should be highlighted that the specific values result without the last step of the grid cell clustering. The coefficient of determination is the correlation between the variable's values and the best predictions that can be computed linearly from the predictive variables, or more simply, it's a statistical measure of how close the data are to the fitted regression line. Moreover, R^2 values lie in the [0,1] range, where a higher value translates to better correlation and indicates that the model explains the variability of the response data around its mean more accurately. Figure 10 illustrates how increasing the number of the grid cells, increases the R^2 score, hence, the robustness of the prediction. This is due to the fact that the spatial aspect of the data is further captured by the algorithm, thus, outliers in the temporal domain are eliminated. As illustrated, without using MaaG (grid size = 1), the R^2 score is significantly low, due to the large data dispersion caused by the geospatial variance of the samples. Thus, the fitted line of the tested models, although trained to provide the prediction with the optimal error, will provide a poor prediction accuracy, given the direct correlation between the prediction error and the R^2 score.

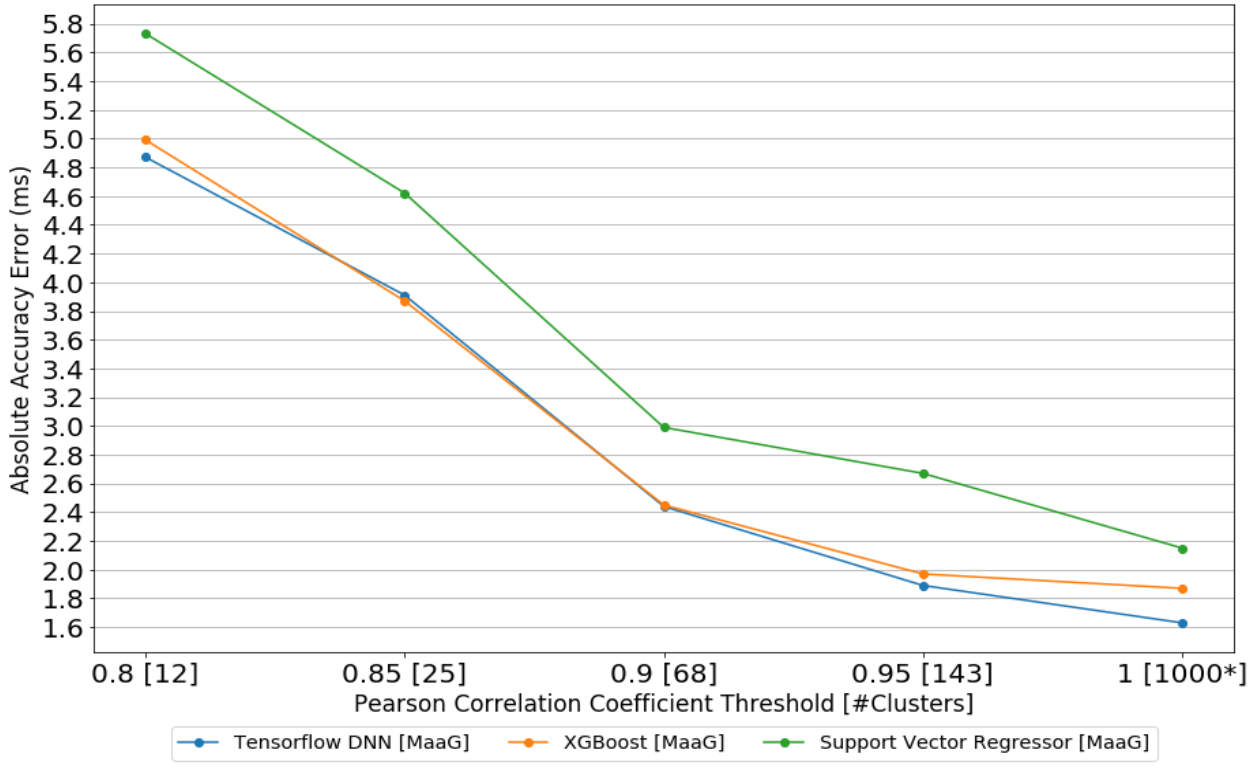


Figure 11. Absolute Accuracy Error per Cluster Size

For the evaluation of the last algorithmic step, i.e., the Grid Cell clustering using the Pearson Correlation method, Figure 11 illustrates the average prediction accuracy error for different clustering parameters. In more detail, the Pearson Correlation Coefficient (PCC) threshold (Equation 2) is evaluated for different values, namely ranging from 0.80 to 1. Higher PCC threshold values translate to higher ("strict") correlation between different Grid Cells, as well as higher number of Grid Cells (i.e., higher granularity of the Grid model). Accordingly, for lower PCC threshold values, the correlation between Grid Cells becomes "looser", hence,

more Grid Cells are clustered together, resulting in fewer Grid Clusters. As already discussed in the previous section, a trade-off is created between the model granularity/accuracy error, and the memory consumption -and as a result the overall performance- of the predictive QoS scheme.

As it is intuitively expected according to the above, all three algorithms exhibit optimal prediction accuracy for the highest granularity (no clustering performed), however, the number of the models that have to be accommodated is maximum (i.e., 1000 in the specific scenario), resulting in a low performance in terms of prediction execution time. The DNN and XGBoost algorithms exhibit a very similar performance, while the SVR algorithm exhibits slightly inferior performance (0.2 to 0.8 ms of accuracy error in absolute values). All in all, as it is can be inferred by Figure \ref{eval8}, significant gains are reported by performing a controlled Cell Clustering approach: For example, a PCC value selection equal to 0.9, can result in a 93\% reduction in the overall number of Grid Clusters (leading thus, to a significantly higher execution performance), while the cost in the Accuracy Error will be less than 1 ms in absolute values, which is negligible for the specific ToD use case.

5. CONCLUSION

This thesis paper presented a QoS prediction scheme for V2X communications, namely PreQoS, is able to accurately predict predefined QoS metrics, such as uplink / downlink delay or data rate, ultimately enabling the network and involved applications to perform the required adaptations for avoiding service interruption. A detailed system model was presented, describing the spatial and temporal modeling of the solution. Additionally, an extensive analysis of the machine learning methods that were applied was presented. Last but not least, an extensive evaluation was performed via a real world-based simulated network deployment that proves the viability and validity of the proposed scheme for the foreseen challenging V2X and CAM use cases.

ABBREVIATIONS – ACRONYMS

QoS	Quality of Service
V2X	Vehicle to Everything
PreQoS	Prediction Quality of Service
ul	uplink
dl	downlink
SUMO	Simulation of Urban Mobility
NS3	Network Simulation 3
SDN	Software-Defined-Networking
ML	Machine Learning
AI	Artificial Intelligence
NWDAF	Network Data Analytics Function
5QIs	5G Quality of Service Identifiers
CAM	Connected and Automated Mobility
C-V2X	Cellular – Vehicle to Everything
5GC	5G Core
RAN	Radio Access Network
UC	Use Case
ToD	Tele-operated driving
MEC	Mobile/Multiple Access Edge Computing
E2E	End-to-End
MaaG	Map-as-a-Grid
Gbit/s	Gigabits per Second
ISP	Internet Service Provider
IoT	Internet-of-Things
LTE	Long Term Evolution
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2V	Vehicle to Vehicle
V2P	Vehicle to Pedestrian
NN	Neural Network
MD	Maximum Dependency
ARIMA	Auto-Regressive Integrated Moving Average
RSRQ/RSRP	Reference-Signal-Received-Quality/Power
SINR	Signal-to-Noise Ratio

LSTM	Long Short-Term Memory
CQI	Channel Quality Indicator
OFDMA	Orthogonal Frequency-Division Multiple Access
RT	Ray-Tracing
TPDIG	Threshold Percentage Dependent Interference Graph
mSC	Moving Small Cell
RB	Resource Block
CF	Collaborative Filtering
KLMS	Kernel Mean Least Square
KPI	Key Performance Indicator
VNF	Virtual-Network-Function
PNF	Physical-Network-Function
RSU	Roadside Unit
PF	Prediction Function
QLF	Quantile Loss Function
PCC	Pearson Correlation Coefficient
BS	Base Station
NR	New Radio
RF-R	Random Forest Regressor Model
KNN-R	K-Nearest Neighbor Regressor
DNN	Deep Neural Network
XGBoost	Gradient Boosting Machine
SVR	Support Vector Regression
MAE	Mean Absolute Error

ANNEX I

In order to take advantage of the functions that NS3 – version 27 - can provide us, there is a list of packages that needed to be installed [40].

- Minimal requirements for C++ users:

```
apt install g++ python3
```

- Additional minimal requirements for Python:

```
apt install python3-setuptools git
```

- Netanim animator: qt5 development tools are needed for Netanim animator; qt4 will also work but we have migrated to qt5:

```
apt install qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools
```

- Support for ns-3-pyviz visualizer:
  For Ubuntu 18.04 and later, python-pygoocanvas is no longer provided. The ns-3.29 release and later upgrades the support to GTK+ version 3, and requires these packages:

```
apt install gir1.2-goocanvas-2.0 python3-gi python3-gi-cairo python3-
pygraphviz gir1.2-gtk-3.0 ipython3
```

- Support for MPI-based distributed emulation

```
apt install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
```

- Support for bake build tool:

```
apt install autoconf cvs bzr unrar
```

- Debugging:

```
apt install gdb valgrind
```

- Support for utils/check-style.py code style check program

```
apt install uncrustify
```

- Doxygen and related inline documentation:

```
apt install doxygen graphviz imagemagick
apt install texlive texlive-extra-utils texlive-latex-extra texlive-
font-utils dvipng latexmk
```

- The ns-3 manual and tutorial are written in reStructuredText for Sphinx (doc/tutorial, doc/manual, doc/models), and figures typically in dia (also needs the texlive packages above):

```
apt install python3-sphinx dia
```

- To read pcap packet traces:

```
apt install tcpdump
```

- Database support for statistics framework

```
apt install sqlite sqlite3 libsqlite3-dev
```

- Xml-based version of the config store (requires libxml2 >= version 2.7)

```
apt install libxml2 libxml2-dev
```

- Support for generating modified python bindings

```
apt install cmake libc6-dev libc6-dev-i386 libclang-dev llvm-dev au-
tomake python3-pip
python3 -m pip install --user cxxfilt
```

Note: Ubuntu versions (through 19.04) and systems based on it (e.g., Linux Mint 18) default to an old version of clang and llvm (3.8), when simply 'libclang-dev' and 'llvm-dev' are specified. The packaging on these 3.8 versions is broken.
Users of Ubuntu will want to explicitly install a newer version by specifying 'libclang-6.0-dev' and 'llvm-6.0-dev'. Other versions newer than 6.0 may work (not tested).

ANNEX II

Build and installation for SUMO tool for MS Windows [39]:

By default, SUMO provides pre-compiled binaries and CMake files to generate Visual Studio projects and can supply the dependent libraries using:

```
git clone --recursive https://github.com/DLR-TS/SUMOLibraries
```

Prerequisites:

- A Visual Studio Community, Professional or Enterprise 2015 or later installation.
- CMake for Windows.
- Python 3.X.
- SUMO sources (either an unpacked src zip or a git clone, see Getting the source code).
- Installed Libraries (Xerces-C, Proj, Fox) preferably by cloning the aforementioned repository.
- Make sure that the SUMO_LIBRARIES environment variable points to your cloned directory.

REFERENCES

[1] 3GPP, TS 23.288, V16.2.0 (2019-12), Architecture enhancements for 5GSystem (5GS) to support network data analytics services (Release 16), December 2019.

[2] ETSI TS 129 520 V15.0.0 (2018-07), 5G System; Network Data Analytics Services.

[3] 3GPP, TS 23.501, V16.3.0 (2019-12), System Architecture for the 5GSystem (Release 16), December 2019.

[4] 3GPP, TS 22.185, V14.4.0 (2018-06), Architecture enhancements forV2X services (Release 14), June 2018.

[5] 3GPP, TS 22.186, V16.2.0 (2019-06), Enhancement of 3GPP support for V2X scenarios (Release 16), June 2019.

[6] 3GPP, TS 23.285, V16.2.0 (2019-12), Architecture enhancements forV2X services (Release 16), December 2019.

[7] 3GPP, TS 23.287, V16.1.0 (2019-12), Architecture enhancements for 5GSystem (5GS) to support Vehicle-to-Everything (V2X) services (Release16), December 2019.

[8] ETSI MEC 022 V2.1.1, Multi-access Edge Computing (MEC); Study on MEC Support for V2X Use Cases, September 2018.

[9] 5GAA White Paper, C-V2X Use Cases, Methodology, Examples and Service Level Requirements.

[10] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, “Connected roads of the future: Use cases, requirements, and design considerations for Vehicle-to-Everything communications,” IEEE Vehicular Technology Magazine, 2018.

[11] A. Kousaridas, A. Schimpe, S. Euler, and et al., “C5G Cross-Border Operation for Connected and Automated Mobility: Challenges and Solutions,” MDPI Future Internet 12, 5, 2020.

[12] 5GAA White Paper, Making 5G Proactive and Predictive for the Auto-motive Industry, November 2019.

[13] 5GAA TR A-190176: Architectural Enhancements for Providing QoS Predictability in C-V2X.

[14] ETSI MEC, Study on MEC Support for V2X Use Cases, ETSI GR MEC022 V2.1.1 (2018-09).

[15] Telecom company web-site. [Online]. Available: https://www.telekom.com/en/company/details/5g-network-as-foundation-for-autonomous-driving-561986

[16] Market business news web-site. [Online]. Available: https://marketbusinessnews.com/why-5g-is-crucial-for-autonomous-vehicles/262328/

[17] Thales group web-site. [Online]. [Available]: https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/industries/automotive/use-cases/v2x

[18] Wikipedia web-site. [Online]. Available: https://en.wikipedia.org/wiki/Vehicle-to-everything

[19] L. Torres-Figueroa, H. F. Schepker, and J. Jiru, “QoS evaluation and prediction for c-v2x communication in commercially-deployed lte and mobile edge networks,” in 2020 IEEE 91st Vehicular Technlogy Conference (VTC2020-Spring), 2020, pp. 1-7.

[20] D. C. Moreira, I. M. Guerreiro, W. Sun, C. C. Cavalcante, and D. A. Sousa, “Qos predictability in v2x communication with machine learn ing,” in 2020 IEEE 91st Vehicular Technology Conference (VTC2020- Spring), 2020, pp. 1–5.

[21] W. Zhang, M. Feng, M. Krunz, and H. Volos, “Latency prediction for delay-sensitive v2x applications in mobile cloud/edge computing systems,” in GLOBECOM 2020 - 2020 IEEE Global Communications Conference, 2020, pp. 1–6.

[22] K. Hammad, S. L. Primak, M. Kalil, and A. Shami, “QoS-aware energy efficient downlink predictive scheduler for OFDMA-based cellular de vices,” IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 1468–1483, 2017.

[23] S. ElAzzouni, E. Ekici, and N. B. Shroff, “QoS-aware predictive rate allocation over heterogeneous wireless interfaces,” in 2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2018, pp. 1–8.

[24] S. Zafar, S. Jangsher, O. Bouachir, M. Aloqaily, and J. Ben Othman, “QoS enhancement with deep learning-based interference prediction in mobile IoT,” Computer Communications, vol. 148, pp. 86 – 97, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366419306620

[25] A. M. Mahmood, A. Al-Yasiri, and O. Y. Alani, “Cognitive neural network delay predictor for high-speed mobility in 5G C-RAN Cellular Networks,” in 2018 IEEE 5G World Forum (5GWF), 2018, pp. 93–98.

[26] 3rd Generation Partnership Project (3GPP), “Study on LTE-based V2X Services (Rel. 14),” Technical Specification (TS) 36.885, June 2016.

[27] J. Li and J. Lin, “A probability distribution detection based hybrid ensemble QoS prediction approach,” Information Sciences, vol. 519, pp. 289 – 305, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002002552030058X

[28] Q. Zhou, H. Wu, K. Yue, and C.-H. Hsu, “Spatio-temporal context-aware collaborative QoS prediction,” Future Generation Computer Systems, vol. 100, pp. 46 – 57, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18332473

[29] H. Wu, Z. Zhang, J. Luo, K. Yue, and C. Hsu, “Multiple attributes QoS prediction via deep neural model with contexts,” IEEE Transactions on Services Computing, pp. 1–1, 2018.

[30] Y. Yin, W. Zhang, Y. Xu, H. Zhang, Z. Mai, and L. Yu, “QoS prediction for mobile edge service recommendation with auto-encoder,” IEEE Access, vol. 7, pp. 62 312–62 324, 2019.

[31] W. Li, P. Zhang, H. Leung, and S. Ji, “A novel QoS prediction approach for cloud services using bayesian network model,” IEEE Access, vol. 6, pp. 1391–1406, 2018.

[32] X. Chen, H. Wang, Y. Ma, X. Zheng, and L. Guo, “Self-adaptive resource allocation for cloud-based software services based on iterative QoS prediction model,” Future Generation Computer Systems, vol. 105, pp. 287 – 296, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X19302894

[33] R. Karim, C. Ding, and A. Miri, “End-to-end QoS prediction of vertical service composition in the cloud,” in 2015 IEEE 8th International Conference on Cloud Computing, 2015, pp. 229–236.

[34] J. Liu and Y. Chen, “A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing,” Knowledge-Based Systems, vol. 174, pp. 43 – 56, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705119300930

[35] H. Wu, K. Yue, B. Li, B. Zhang, and C.-H. Hsu, “Collaborative QoS pre diction with context-sensitive matrix factorization,” Future Generation Computer Systems, vol. 82, pp. 669 – 678, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17304570

[36] X. Luo, J. Liu, D. Zhang, and X. Chang, “A large-scale web QoS prediction scheme for the industrial internet of things based on a kernel machine learning algorithm,” Computer Networks, vol. 101, pp. 81 – 89, 2016, industrial Technologies and Applications for the Internet of Things. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616000189

[37] V. Vasilev, J. Leguay, S. Paris, L. Maggi, and M. Debbah, “Predict ing QoE factors with machine learning,” in 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–6.

[38] U. Lekhala and I. Haque, “PIQoS: A Programmable and Intelligent QoS Framework,” in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2019, pp. 234–239.

[39] Official SUMO website. [Online]. Available: http://sumo.sourceforge.net/

[40] Official ns-3 website. [Online]. Available: https://www.nsnam.org/

[41] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, “End-to-end simulation of 5G mmWave networks,” IEEE Communications Surveys Tutorials, vol. 20, no. 3, pp. 2237–2263, 2018.

[42] 5GCroCo Deliverable D2.1, Test Case Definition and Trial site Description Part 1, 2020. [Online]. Available: https://5gcroco.eu/images/templates/rsvario/images/5GCroCoD21v2:pdf