**NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS**

**SCHOOL OF SCIENCE**
**DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION**

**INFORMATION AND DATA MANAGEMENT**

**MSc THESIS**

# AI-Driven, Predictive QoS for V2X Communications in 5G and beyond.

**Nikolaos E. Maroulis**

**Supervisors:** **Athanasia Alonistioti,** Associate Professor
**Sokratis Barmpounakis,** PhD

**ATHENS**

**October 2021**

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΔΙΑΧΕΙΡΙΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΔΕΔΟΜΕΝΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

# Πρόβλεψη της Ποιότητας Υπηρεσίας για V2X επικοινωνίες στο 5G και περαιτέρω, μέσω αλγορίθμων Τεχνητής Νοημοσύνης.

**Νικόλαος Ε. Μαρούλης**

**Επιβλέποντες:** **Αθανασία Αλωνιστιώτη,** Αναπληρώτρια Καθηγήτρια
**Σωκράτης Μπαρμπουνάκης,** Διδάκτωρ

**ΑΘΗΝΑ**

**ΟΚΤΩΒΡΙΟΣ 2021**

**MSc THESIS**

# AI-Driven, Predictive QoS for V2X Communications in 5G and beyond.

**Nikolaos E. Maroulis**
S.N: M1605

**Supervisors** **Athanasia Alonistioti,** Associate Professor
**Sokratis Barmpounakis,** PhD

October 2021

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

# Πρόβλεψη της Ποιότητας Υπηρεσίας για V2X επικοινωνίες στο 5G και περαιτέρω, μέσω αλγορίθμων Τεχνητής Νοημοσύνης.

**Νικόλαος Ε. Μαρούλης**
Α.Μ: M1605

**ΕΠΙΒΛΕΠΟΝΤΕΣ:** **Αθανασία Αλωνιστιώτη,** Αναπληρώτρια Καθηγήτρια
**Σωκράτης Μπαρμπουνάκης,** Διδάκτωρ

Οκτώβριος 2021

# 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 thesis 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**: Machine Learning

**KEYWORDS**: Timeseries, 5G, V2X, Forecasting, Predicition, Geo-spatial Data, CAM, Quality of Service prediction, Artificial Intelligence, Machine Learning

# ΠΕΡΙΛΗΨΗ

Με το ξεκίνημα της εποχής της συνδεδεμένης και αυτοματοποιημένης κινητικότητας με δυνατότητα 5G, έχουν προκύψει καινοτόμες υπηρεσίες Vehicle-to-Everything προς ασφαλέστερη και αυτοματοποιημένη οδήγηση. Οι απαιτήσεις που απορρέουν από αυτές τις υπηρεσίες θέτουν πολύ αυστηρές προκλήσεις στο δίκτυο κυρίως όσον αφορά την καθυστέρηση από άκρο σε άκρο και την αξιοπιστία των υπηρεσιών. Ταυτόχρονα, η τεχνητή νοημοσύνη εντός δικτύου που αναδύεται, αποκαλύπτει μια πληθώρα νέων δυνατοτήτων του δικτύου να ενεργεί με προληπτικό τρόπο προς την ικανοποίηση των προαναφερθέντων μεγάλων απαιτήσεων. Αυτή η διατριβή παρουσιάζει το PreQoS, έναν προγνωστικό μηχανισμό Ποιότητας Υπηρεσιών που εστιάζει στις υπηρεσίες Οχήματος-προς-Όλα (V2X). Το PreQoS είναι σε θέση να προβλέψει έγκαιρα συγκεκριμένες μετρήσεις Ποιότητας Υπηρεσιών, όπως ο ρυθμός δεδομένων uplink and downlink και η καθυστέρηση από άκρο σε άκρο, προκειμένου να προσφέρει το απαιτούμενο χρονικό διάστημα στο δίκτυο για την πιο αποτελεσματική κατανομή των πόρων του. Επιπλέον, η προληπτική διαχείριση αυτών των πόρων επιτρέπει στις αντίστοιχες υπηρεσίες και εφαρμογές του Οχήματος προς Όλα να εκτελούν εκ των προτέρων τυχόν ενδεχόμενες προσαρμογές που σχετίζονται με την Ποιότητα Υπηρεσιών. Η αξιολόγηση του προτεινόμενου μηχανισμού με βάση ένα ρεαλιστικό, προσομοιωμένο, συνδεδεμένο και αυτοματοποιημένο περιβάλλον κινητικότητας αποδεικνύει τη βιωσιμότητα και την εγκυρότητα μιας τέτοιας προσέγγισης.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ**: Μάθηση Μηχανής

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**: Χρονοσειρές, 5G, V2X, Πρόβλεψη, Πρόγνωση, Γεω-χρονικά Δεδομένα, CAM, Πρόγνωση Ποιότητας της Υπηρεσίας, Τεχνητή Νοημοσύνη, Μάθηση Μηχανής

*This page intentionally left blank*

# ACKNOWLEDGEMENTS

I would like to deeply thank my supervisors, and especially prof. Nancy Alonistioti for the continuous support and guidance during the development of my thesis. Dr. Sokratis Barmpounakis' guidance was really valuable for me, in order to get access to the necessary tools and data in order to complete my thesis. Finally, I would like to thank Lina Magoula (PhD Candidate) and Nikolaos Koursioubas (PhD Candidate) for all the support both in the theoretical, as well as, the practical implementations of this thesis.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# PREFACE

The current thesis has been conducted for the master'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 a timeseries prediction algorithm which is based on 5G QoS KPIs, as well as geo-spatial data, using machine learning techniques. In the context of the present work, the proposed system has been implemented using Python, Tensorflow, Scikit-Learn, Pandas, Seaborn, Jupyter Notebook and more 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 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 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 BitRate/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].

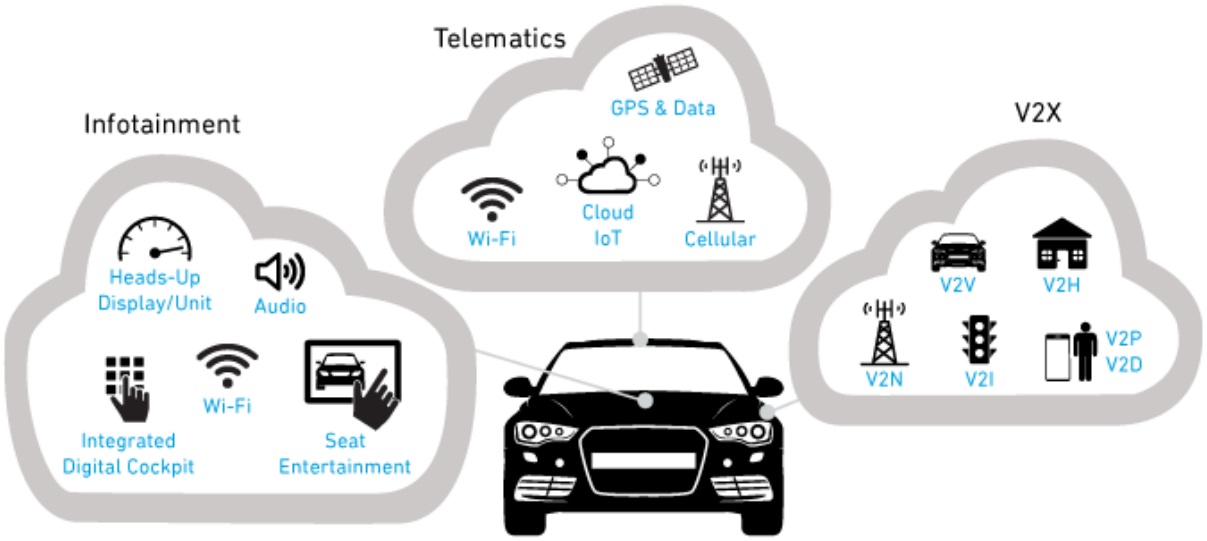


**Figure 1 V2X Architecture Overview**

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 cross-border 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.

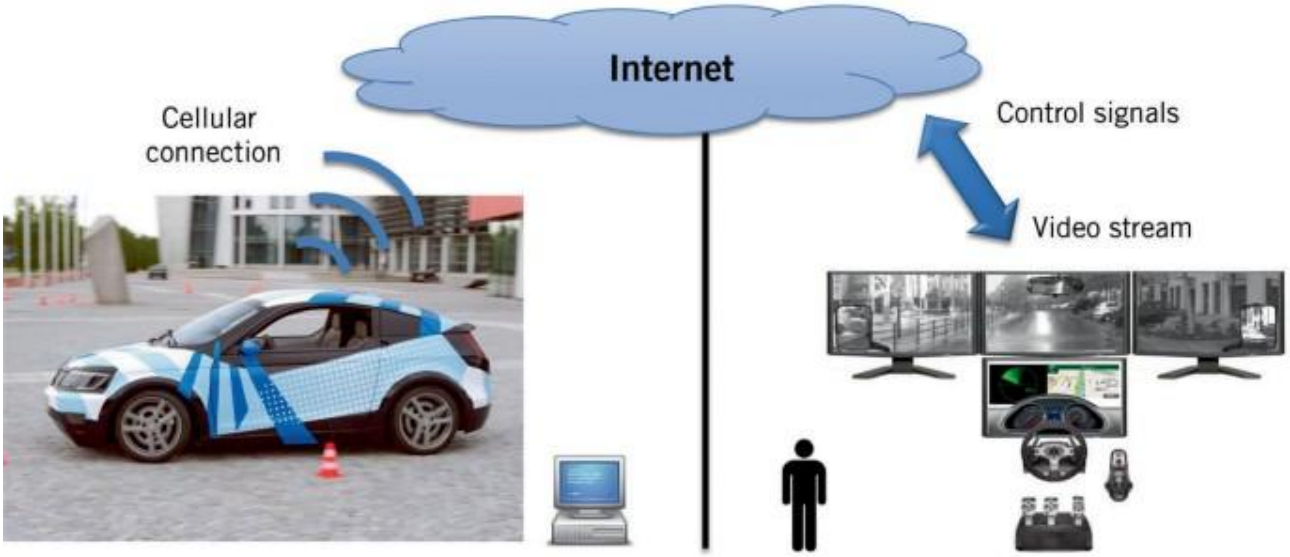


**Figure 2 Tele-operated Driving Deployment**

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 very recently also 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 hostsand 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, alongwith 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 delay and data rate requirements. ***Predicting QoS KPIs*** in advance will offer major gains to the operation of such V2X applications, in the sense that, predicting in advance a potential deterioration on the utilized communication channel and/or offered communication or computing resources will enable the respective application to adapt itself in order to seamlessly continue operating and -most importantly- exclude any chance of jeopardizing the safety of the involved humans. 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. Inspired by the above-mentioned challenges, in this 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. This thesis' primary contributions are summarized as follows:

- A predictive QoS algorithm is presented, namely **PreQoS**, which processes contextual information towards proactive service adaptation for service continuity in stringent V2X services. This contextual information comprises data, such as vehicle mobility information, radio and network parameters, as well as service/application-specific information. Its ultimate target is to predict specific QoS-related values, such as uplink/downlink end-to-end delay,data rate, etc.
- 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. Via the MaaG paradigm, this thesis highlights the scalability perspective of the proposed mechanism. On top of this, a comprehensive analysis of the computational complexity and network overhead is introduced for the proposed framework.
- 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. The evaluation also correlates the *accuracy error* and *coefficient determination* metrics for each ML model, in relation to the spatial modeling options and volume of training data. The evaluation section provides insights in terms of the performance of different ML models, which are exploited by the PreQoS framework.

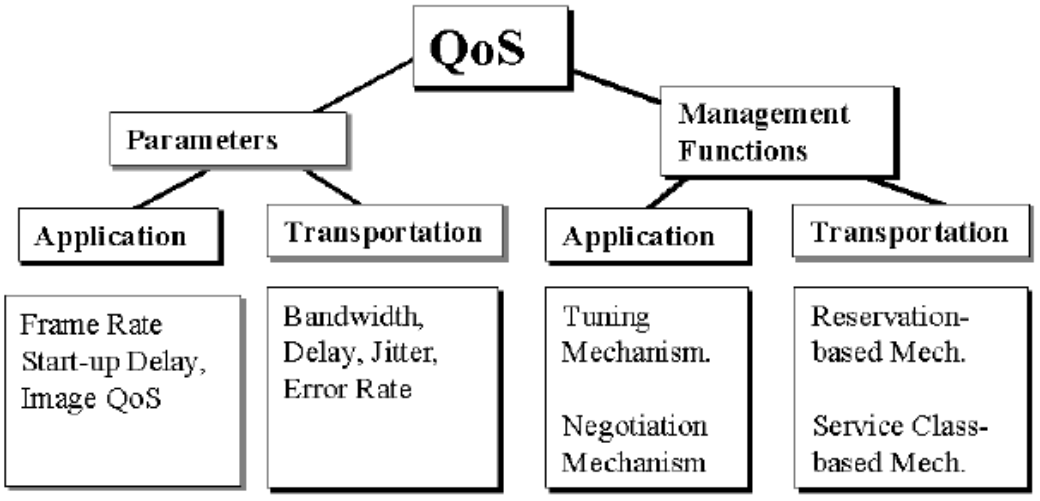


**Figure 3 QoS Parameters and Functions**

The rest of the thesis is organized as follows: Section 2 presents an overview of the existing work in the literature on proactive QoS approaches. Section 3 presents a comprehensive description on the system model, design and algorithmic details of the

proposed QoS scheme, while Section 4 presents the evaluation scenarios and assessment of results. Finally, Section 5 concludes this thesis.

# 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 [15] [16] [17], but also in the domain of mobile networks, from a broader perspective [18] [19] [20] [21]. In [15], 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* (as shown in Figure 4) 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.

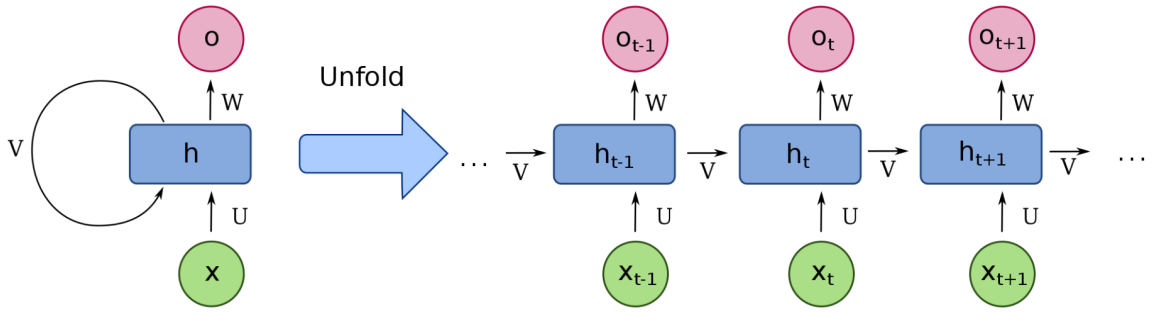


**Figure 4 RNN - LSTM HIgh Level Architecture**

In [16], 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 [22]. 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 [17], a latency-prediction framework is described, tailored for delay-sensitive V2X applications. The proposed scheme integrates ML, i.e., an 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 [18], a predictive energy-efficient scheduling scheme is proposed, that optimizes the user equipment (UE)'s bits/joulemetric subject to QoS constraints in downlink orthogonal frequency-division multiple access (OFDMA) systems. Ham-madet al. were able to achieve that by minimizing the number of wake-up transmission time intervals (TTIs), 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 [19] 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 [20] 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 (mSCs). A comparative analysis of resource allocation approaches is presented, using TPDIG, Time Interval Dependent Interference Graph (TIDIG), and Global Positioning System Dependent Interference Graph (GPSDIG), 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 [21], a Cognitive Neural Network Delay Predictor (CNNDP) for high speed mobility in 5G C-RAN cellular networks is proposed, for compensating the transmission and acquisition delay ofthe Channel State Information (CSI) 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 [23] [24] [25] [26]. In [23], a probability distribution detection-based hybrid ensemble approach (DHEM) 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), as shown in Figure 5, based approach as the basis of the prediction model. The authors propose a distribution detection algorithm in order to calculate the probability confidence weights (PCWs), based on the results of a set of other basic prediction models.

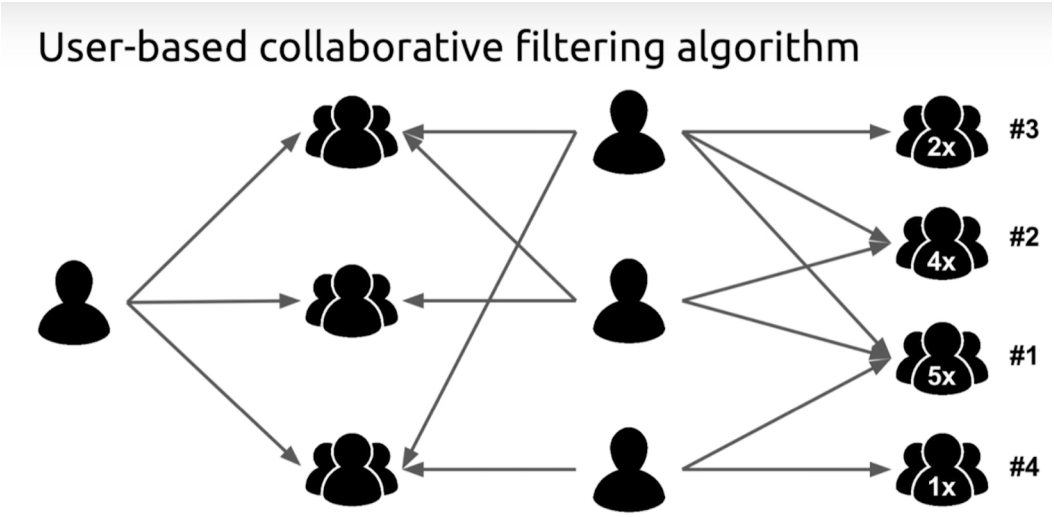


**Figure 5 Collaborative Filtering Overview**

Zhou et al.in [24] 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 (NMAE) in comparison with the baseline methods. The work in [25] 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 [26], Yin et al. propose a method of combining an auto-encoder with collaborative filtering (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 [27] [28] [29] [30] [31] [32]. In [27] 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 (BNM).

In [28], 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 (PSO)-based run time decision algorithm. The work in [29] 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 [30], 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 [31], the authors propose a Matrix Factorization (MF)-based approach for making context-aware QoS-prediction of specific cloud services. Luo et al.in [32] present a novel data-driven QoS prediction scheme using KLMS for the purpose of achieving a higher accuracy. Via the trained Kernel Mean Least Square (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 [33] [34] . In [33], 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 rebuffering 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 [34], 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 thesis primarily relates to [15] - [17], and secondarily to [18] -[21], which although tackle similar research questions, do not focus on V2X and/or CAM scenarios, which is the key use case (UC) of this thesis. The work in [15] firstly tackles the problem as

a classification problem; in my case, I 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 [16], the authors employ a simple moving average regression approach (ARIMA), which fails to cope with complex scenarios, high number of UEs, etc. The proposed scheme in [17] is the closer work in terms of design decisions, combining LSTM with clustering approaches; nevertheless, complexity analysis, which is tackled in the current thesis, 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 [18] is limited by the selection of the Ray-Tracing (RT)-based prediction technique, which relies solely on the physical layer, and more specifically on the wireless communication channel propagation characteristics.The work in [19] 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 the algorithm presented in this thesis is capable of extending the prediction window ahead, as the volume of the training data increases over time. The work in [20] 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 [21] does not perform direct QoS metric predictions; instead it focuses on the acquisition delay in the Channel State Indicator (CSI) metric, which is indirectly linked to the variation of the communication channel conditions. In this thesis, I extend the current state of the art, by proposing a novel predictive QoS scheme, tailored for 5G CAM use cases. To my 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 thesis 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 thesis follows the concepts and terminology introduced by [35]. 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 thesis.

## 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 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). One of the main advantages of the proposed framework is that the QoS metric that is studied is tailored each time, in accordance with the specific V2X application requirements, namely uplink or downlink data rate, end-to-end delay, reliability, packet loss, jitter, 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 [36], this end-to-end communication path depends on the type of the V2X application/service, i.e., Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I),Vehicle-to-Network (V2N) or Vehicle-to-Pedestrian (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 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 are grouped in five main categories, as illustrated in the following table (Table I):

**Table 1 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 constraints (e.g. road limits), etc. |
| *Radio parameters* | Radio-related, passive measurements provided via the UE-based measurement reporting to the gNB/eNB. These metrics are complementary, in order to enhance the prediction. | RSSI, RSRP, RSRQ, CQI, SINR, client (GPS, velocity, heading), geolocation map |
| *RAN/facilities layer* | Latency-introducing network components related to queues, computing resources' capacity, etc. | RSU-related network load, RSU-queues load, Number of UEs associated to BS, availability of MEC/cloud computing resources |
| *E2E network performance* | Network measurements related to the E2E service, including transport and core network measurement information. | transmission/queuing delays, backhaul link capacities, VNF processing delays, etc. |
| *Application-specific information* | Tailored, application-oriented information that influences the performance of a 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 all potential information item types that can be processed during the offline training process, if available. The flexibility of the proposed scheme enables the processing of the data and the extraction of the respective prediction, depending each time on the specific network deployment, interfaces, and available real-time data sources, which often comprise only a subset of the afore-presented information items.

## 3.2 System Model

This thesis considers a prediction communication system, in which a number of $u \in U$ users (i.e., vehicles) are consuming a number of different V2X services $s \in S$. Users are notified by the network via a QoS prediction message in relation to 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 $u * k$ KPIs, which are then processed by the respective V2X applications for possible required adaptations.The prediction inference process can be defined to be performed in two possible ways:

A. in a pre-defined (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).

I 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 Network Data Analytics Function (NWDAF) [2]. The PF can be deployed either at a mobile edge computing (MEC) server or a cloud center, as part of the rest of the Core Network functions and entities. Intuitively, in the case of a MEC-enabled 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.

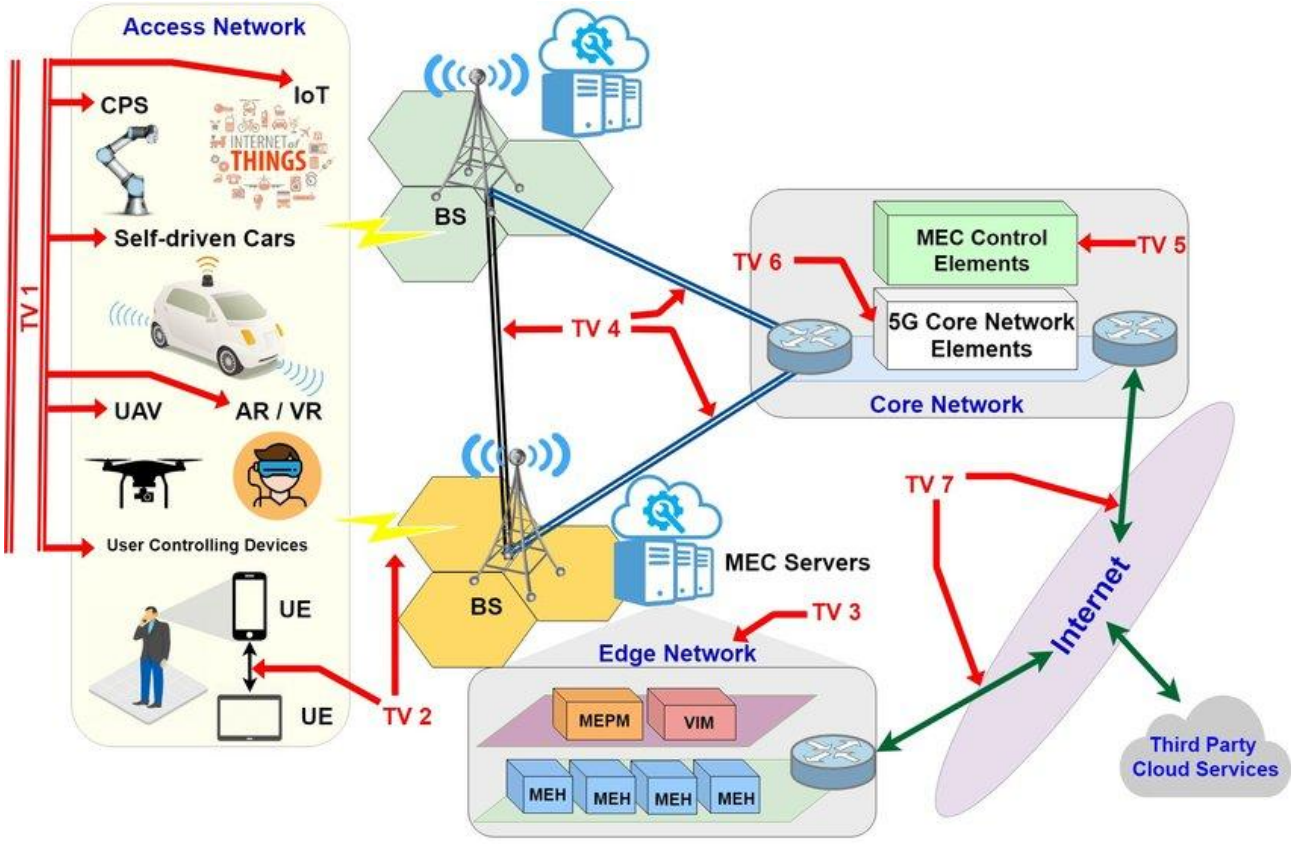


**Figure 6 MEC Deployment Architecture Overview**

### 3.2.1 Geographical space as a grid

The considered geo-graphical 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 7a and 7b). 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 and for the same time horizon. It should be highlighted that as long as the correlation is realized only based upon the QoS values, the members of one Grid Cluster do not need to be adjacent cells (see for example Figure 7b $c_2$ and $c_3$). The detailed algorithmic steps are presented in subsection 3.4.

### 3.2.2 Computational complexity and signaling overhead analysis

The major advantage of the described 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., the [latitude,longitude] tuple) of each measurement is transformed via the **MaaG** mapping to one specific Grid Cell or Grid Cells Cluster, identified by a unique ID. This results in a reduction of the required computing tasks from $u*k$ predicted values, to $c*k$, where $c \in C$, the defined clusters of the system and $c \leq u$. More specifically, I model the end-to-end computational delay for the system, in order for the UEs to send a prediction request towards the Prediction Function, the request to be processed, the prediction to be extracted by the respective ML model and -finally- the prediction output to be transmitted back to the respective UE.

### 3.2.3 System Model Implementation

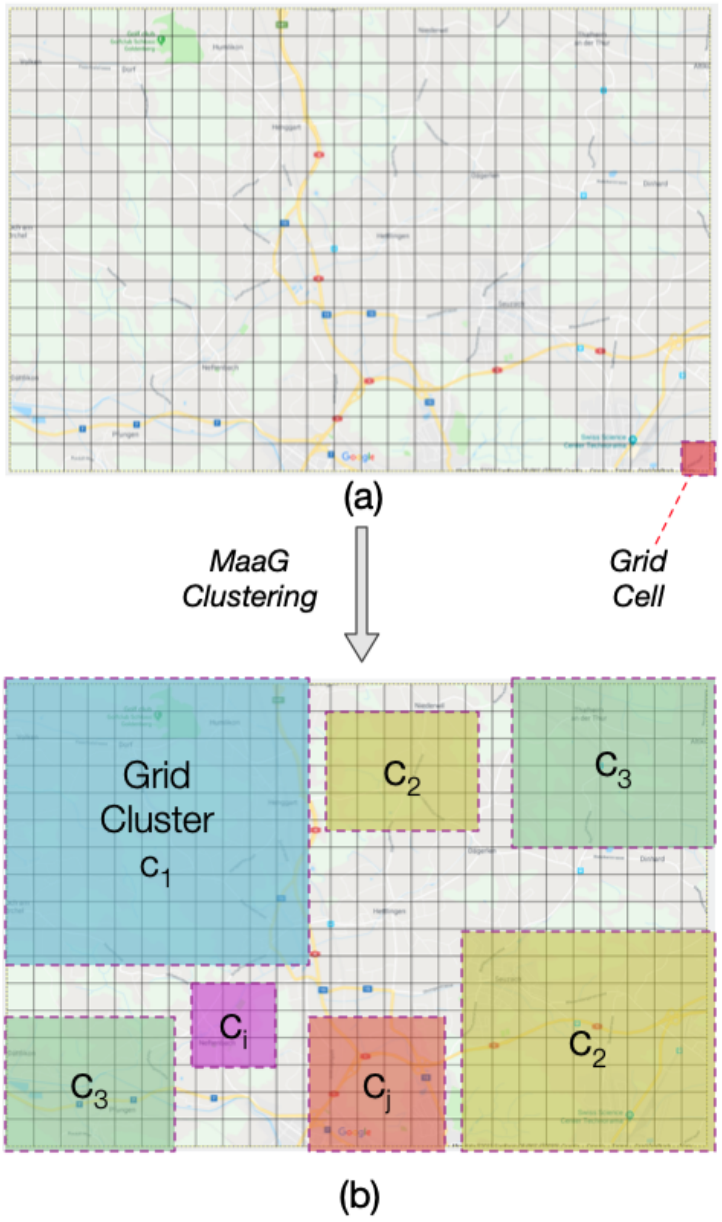


**Figure 7 (a) MaaG - Step 1: Initial grid modeling, (b) MaaG - Step 2: Grid after the QoS-based correlation clustering**

The overall prediction delay $D_{pred}$ for one KPI (e.g., for the uplink throughput QoS metric), for the ***non-Maag*** enabled system is defined as the sum of the transmission delay of the prediction request for all the vehicles of the system, the computing delay at the Prediction Function server for all requests, as well as the transmission delay of the prediction response back to the UEs, and is defined as follows:

$$D_{pred} = \sum_{i=1}^{u} d_{ul\text{-}req} + \sum_{i=1}^{u} d_{comp} + \sum_{i=1}^{u} d_{dl\text{-}resp},$$

where $u$ is the overall number of vehicles of the system, while $d_{ul\text{-}req}$, $d_{comp}$ and $d_{dl\text{-}resp}$ are the respective prediction request transmission, prediction computing and prediction response transmission delays, respectively. Taking into account that the data size of the predictive QoS request and response messages is fixed [12] and directly

related to the specific online/offline prediction operation -which I assume common for this comparison analysis-, the respective transmission delay is defined as follows:

$$d_{ul\text{-}req} = S_{msg\text{-}request}/R_{i,j}$$

where $S_{msg\text{-}request}$ is size of the prediction request message and:

$$R_{i,j} = B_l * \log_2((1 + P_l * h)/\sigma^2 + I)$$

is the data rate (in bits per second) between *UEi* and its associated base station *j*, $B_l$ is the bandwidth of the link, $P_l$ is the transmission power of the mobile device, σ2 is the noise power of mobile devices and *I* is the inter-cell interference. In the same way, the response message transmission delay $d_{dl\text{-}resp}$ is given. I assume that the propagation and queuing delays are equal to zero, without affecting the comparison analysis. Similarly, I estimate the overall prediction delay $D_{pred}$´ when deploying the MaaG algorithm, presented earlier:

$$D_{pred}' = \sum_{i=1}^{h} d_{ul\text{-}req} + \sum_{i=1}^{c} d_{comp} + \sum_{i=1}^{u} d_{dl\text{-}resp},$$

where *h* is the number of vehicles, which transmit a prediction request, with $c \leq h \leq u$ and *c* is the number of the formulated MaaG clusters. In the MaaG model, if a vehicle, inside one of the Grid Clusters *c* realizes a prediction request, the computed prediction output applies for all vehicles, which are moving in the same Grid Cluster. Also, the prediction computation is realised only for *c* times -one per cluster-, rather than for *u* times, in the non-MaaG approach. Thus, from these four previous equations we get that:

$$D_{pred}' \leq D_{pred}$$

for c≥1, or in other words, if at least one clustering operation has taken place and there are more than 1 Grid Clusters. The above analysis for the prediction end-to-end delay is directly related -besides the computational complexity- also to the network overhead minimization, as the prediction requests, along with the transmitted prediction request message sizes are considerably lower.

### 3.2.4 Time-based Modeling

The time dimension 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 networks' traffic load, which is directly linked to the road traffic volume and density characteristics, i.e., the number of vehicles requesting resources within a specific cell. Those parameters in turn are assumed to follow 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 9) it is able to capture that seasonality. The model can be adapted in order to perceive other types of seasonality characteristics

as well, such as annual or monthly. Specific vehicle volume/mobility characteristics are identified also 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 pre-defined 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 may be defined at one minute (60 seconds) 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.

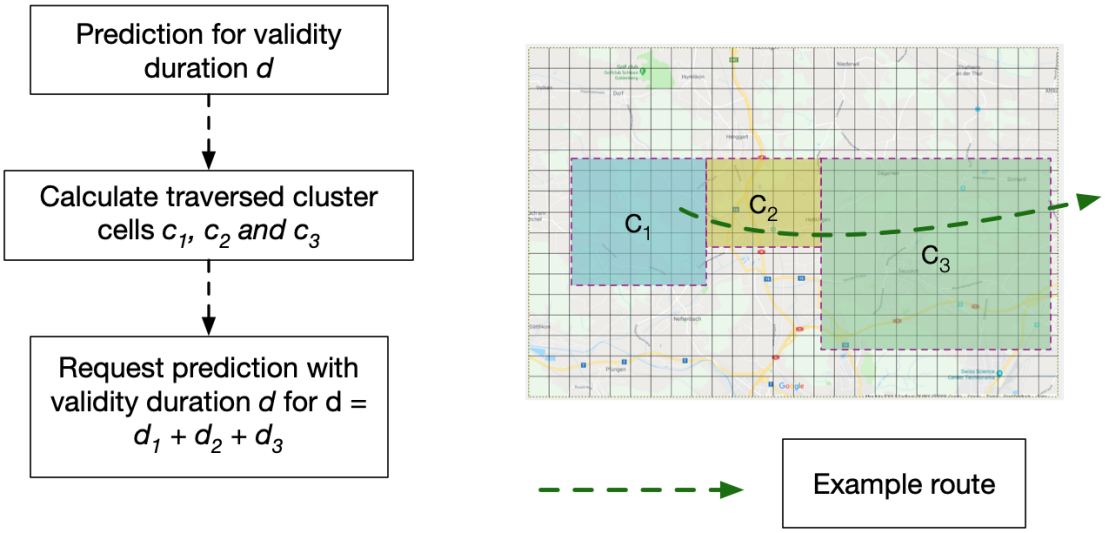


**Figure 8 Prediction validity duration via calculation of traversed cells**

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 8 illustrates an example of the afore-described concept. Let me denote prediction validity duration *d*, for which the vehicle path comprises three adjacent cells $C_1$, $C_2$, and $C_3$. Intuitively, the overall prediction validity is equal to the sum of the duration that each cell is traversed:

$$d = \sum_{i=1}^{n} d_i,$$

for *n=3*.

### 3.3 PreQoS algorithmic framework

The workflow of the training part of the PreQoS algorithm is described in Figure 9. The main algorithmic steps comprise the data preprocessing, the training of the data, the application of the regression model, the cell clustering/partitioning, as well as the extraction of the respective ML model. As illustrated in Figure 9, the first steps of the training part of the algorithm comprise:

1. The aggregation of the different input data types, as presented in Table I
2. 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.

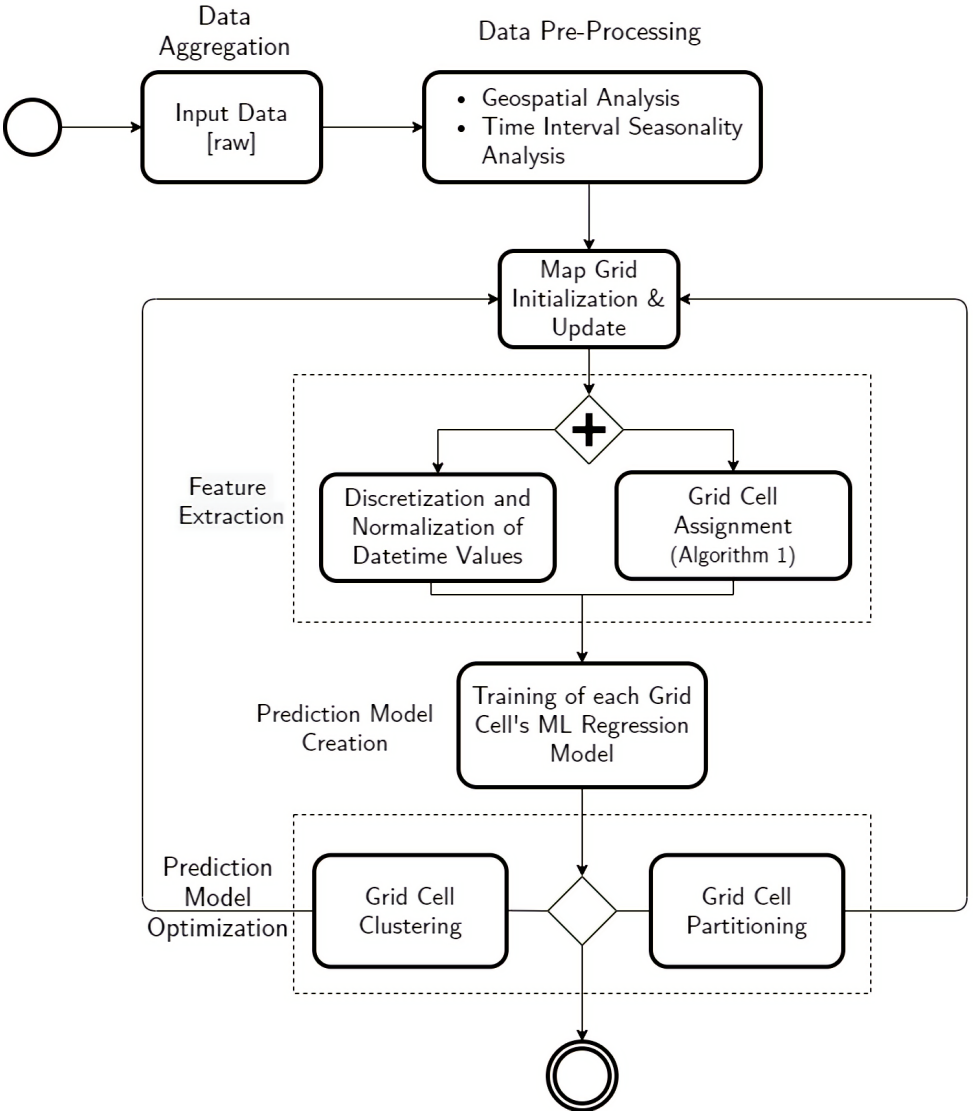


**Figure 9 Data preprocessing, Model Training and MaaG Optimization (Clustering/Partitioning)**

As a part of the PreQoS workflow, Algorithm 1 (Figure 10) describes the steps for determining the Grid Cell ID of a vehicle, given its latitude and longitude. The Grid Cell ID -which is used as a unique identifier for the specific cell- follows a two dimensional array-modeled indexing [row,column] in ascending order [top to bottom, left to right].For

each single Grid Cell, a Regression model is created and saved to memory for the inferencing part. 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 11.

**Algorithm 1** Latitude and Longitude to Grid Cell Id as part of the PreQoS Inferencing part

$num\_cells \leftarrow num.\ rows/columns\ of\ the\ Grid$
**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 10 Grid Cell Assignment Algorithm (Spatial Transformation)**

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. This part of the algorithm is used both during the pre-processing -prior to the training of the model-, as well as the inferencing process, for responding to the vehicles'/clients' predictive QoS requests. The Regression method was considered, as only interpolation is needed for the prediction, with no need for extrapolation over the examined time horizon.

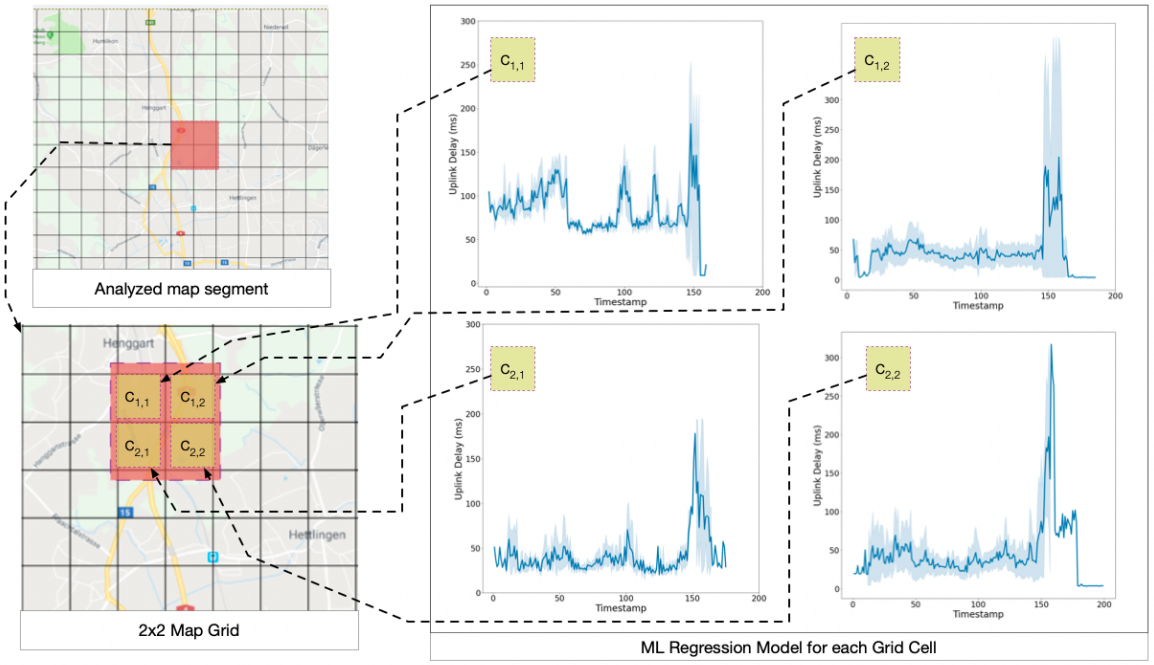


**Figure 11 Regression Model for each Grid Cell**

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 my 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 (shown in the equation below), where Quantile-based regression aims to estimate the conditional ***"quantile" (α)*** 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.

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

, where $\xi_i$ is each sample's real y value. The last step of the algorithm is the optimization of the prediction model via the methods of the Grid Cell Clustering and Partitioning. The clustering and the partitioning are two inverse and complementary methods applied at each iteration of the offline training phase of the model, depending on the cells'/clusters' QoS correlation calculation, via the ***Pearson Correlation Coefficient (PCC)***. In most cases, a subset of the clusters may be clustered, or others may be partitioned into smaller clusters/cells, depending on the temporal variance of the QoS measurements. Cell Clustering is applied in order to group (cluster) Grid Cells with correlated QoS behaviour, into Grid Clusters. Pearson correlation is used (Equation shown below), in order to measure the linear correlation between two time series, where if the absolute value of the PCC ***ρ*** is greater than a predefined threshold (e.g., 0.85, 0.90, 0.95 etc.), the cells now reference a single ML model, thus forming a cluster. The data from all the Grid cells that comprise the Cluster, are merged in order to form the dataset which is used for the re-training of the ML Model of the newly created Cluster, hence the *n* ML

models (where *n* is the number of the Clustered Grid cells) are discarded, ending up with a single ML model for the Cluster. Finally, 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.

$$\rho = \frac{\sum_{i=1}^{n}(x_{a\,i}-x_{b\,i})\cdot(y_{a\,i}-y_{b\,i})}{\sqrt{\sum_{i=1}^{n}(x_{a\,i}-x_{b\,i})^2}\cdot\sqrt{\sum_{i=1}^{n}(y_{a\,i}-y_{b\,i})^2}}$$

, where I 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 9. In the same rationale with the Grid Cell Clustering, if a Grid Cluster at some point in time does not fully capture the spatial aspect of the data, the QoS feature variance within the Cell will increase. Therefore, the coefficient of determination $R^2$ for each ML model is calculated, and if $R^2$ is lower than the defined PCC value, the Cluster/Cell is re-partitioned into its previous state (i.e., to the comprising Grid Cells of the specific Cluster). Towards, retaining the Cell-Cluster mapping and applying partitioning and clustering in an effective manner, the algorithm models the MaaG entity as a stateful data structure, i.e., the list of Clusters, along with the previous states (list of comprising cells): each time clustering is applied the state is increased (for a particular cluster) by one. If at some point in time, partitioning is applied, the state is reduced by one, meaning that we return to the previous state by one. Both in the Clustering, as well as in the Partitioning cases, a revised ML model is deployed, which is retrained based on the updated cluster-/cell-specific data, that results after the respective (clustering/partitioning) action. With regard to the model's performance, 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. In the next section, this assumption is thoroughly investigated and respective results are discussed.

# 4. EVALUATION

## 4.1 Simulation Architecture and Setup

In order to evaluate the performance of the proposed mechanism, I implemented a real-world simulated mobility scenario using Simulation of Urban Mobility (SUMO) [37], which is an open source road simulation software. The extracted mobility patterns were imported into NS-3 discrete-event Network Simulator [38] for performing a complete, end-to-end communication scenario for a specified time duration, exploiting the 5G mmWave module introduced in [39]. The simulated geographical area that was selected for the performed evaluations is located in Munich, Germany near the Huawei Munich Research Center. In Figure 12a, I 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 12b the SUMO-based transformation into the virtual scenario is illustrated. Overall, Table 2 presents a summary for all the parameters used in the mobility simulation.

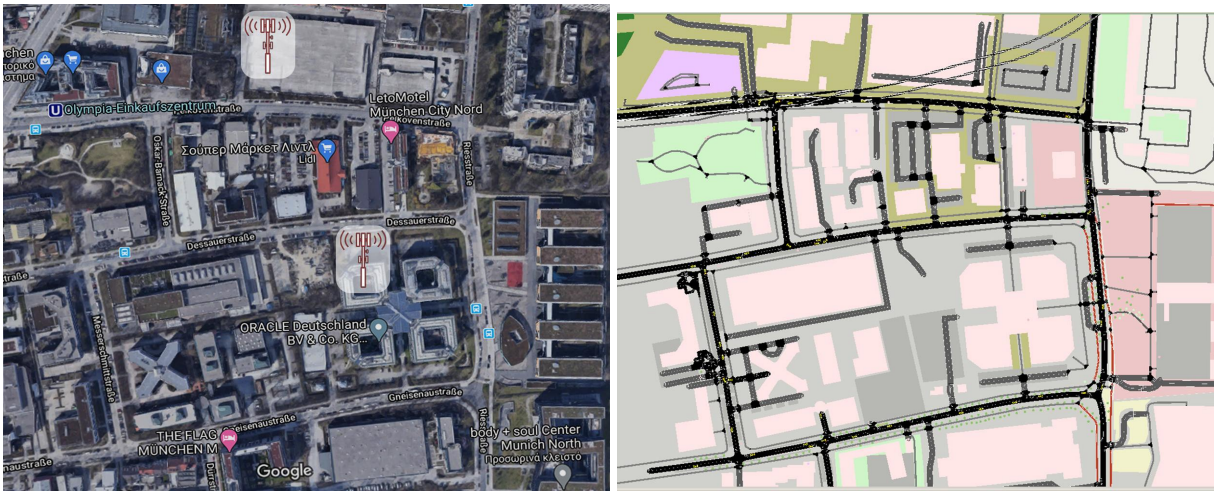


**Figure 12 (a) Geographical Location as it has been retrieved from Google Maps and (b) Converted in SUMO mobility simulation tool**

Tele-operated Driving (ToD) is the V2X use case that has been modeled in the specified simulation scenario. The goal of the ToD use case is applied both to human-operated vehicles, as well as to connected and autonomous vehicles: in the first case, in order 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; in the automated mobility scenario, 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 [40]. 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 the maneuver instructions via the downlink and adjusts its trajectory, speed and acceleration. Acknowledgement feedback is then 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. More information about the QoS requirements of the ToD use cases is available in [40]. The importance of the QoS

prediction for the specific, ToD use case, is highlighted by the fact that in case the required QoS - in terms of uplink throughput for the video stream transmission, as well as the downlink delay for the remote commands' transmission from the Remote Driver to the vehicle- cannot be met, the ToD application is required to perform application-specific adaptation, to safely and seamlessly continue operating, -even for degraded QoS-. Such an example would be that the quality of the vehicle-transmitted video stream is smoothly reduced in advance, in case the predicted uplink throughput is lower. The simulation scenario assumes 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 diverse 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.

**Table 2 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 |

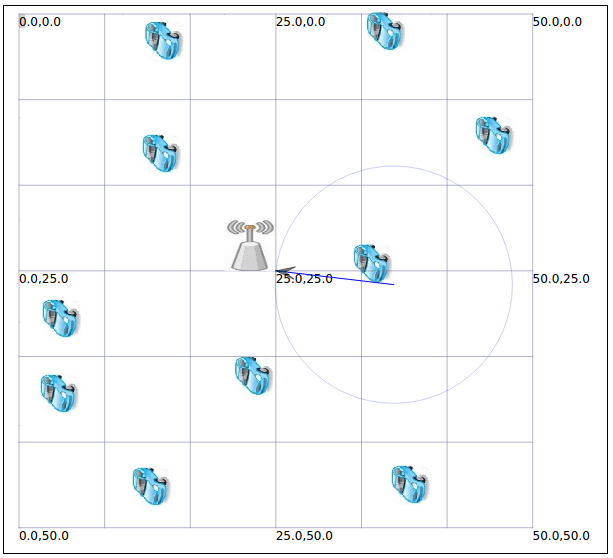


**Figure 13 NS-3 Simulation GUI Example**

Overall, Table 3 presents a summary of the simulation configuration parameters. 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 [40]. The ToD use case involves at least two application endpoints, namely the ToD client (on the vehicle side) and the ToD server (at the Vehicle Control Center, on the cloud/MEC side). The end-to-end measurement refers to the network latency between those two endpoints on the application layer. 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 setup, and how each selection influences the downlink delay KPI.

**Table 3 Parameters used in the NS-3 simulated scenarios**

| Parameter Description | Default Value |
|---|---|
| 5G NR / LTE scheduler | Proportional Fair scheduler |
| 5G NR / LTE eNBs' Height | 24 m |
| UEs' Height | 1.6 m |
| 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 NR Transmission Power | 30 dBm |
| UEs' Transmission Power | 20 dBm |
| LTE Downlink and Uplink Bandwidth | 20 MHz |
| 5G NR Downlink and Uplink Bandwidth | 100 MHz |

As it is shown in Figure 14, 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 many realistic and challenging conditions as possible.

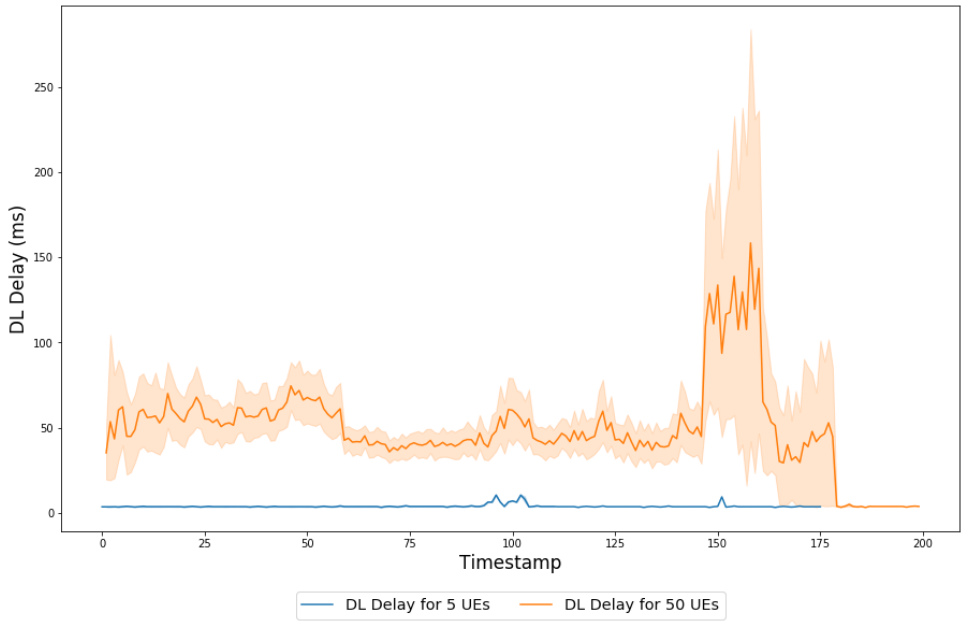


**Figure 14 Average system downlink end-to-end delay (ms) for the two scenarios(5 and 50 UEs respectively)**

### 4.2 PreQoS Performance Evaluation

In this section, I present the outcomes of the evaluation of the afore-presented prediction scheme, differentiating between two main ML algorithm groups: the one is denoted as non-MaaG, i.e., an approach, in which I do not employ the Map-as-a-Grid spatiotemporal algorithm presented earlier, and which is evaluated based on 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 4 presents an overview of the training parameters that were used for each one of the above ML models.

**Table 4 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 I define employing the Mean Absolute Error (MAE) metric:

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

, 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 $\hat{y}_i$ 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- I 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 4. 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 intervals, 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, I 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.

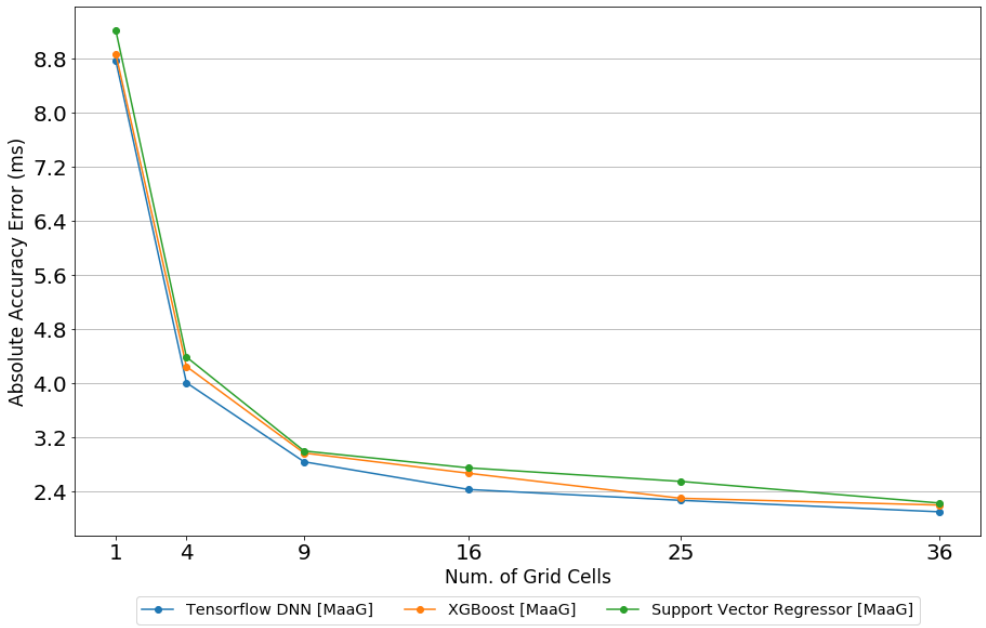


**Figure 15 Accuracy Mean Absolute Error based on the number of Grid Cells**

Figure 15 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 50.000samples. 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 ~9ms (due the high variance in the data, by not capturing any geospatial information within the data) down to ~2ms 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. The same trend is followed for the Mean

Absolute Percentage Error KPI, illustrated in Figure 16. The specific results illustrate the Mean DL delay Prediction Accuracy Percentage Error (MAPE), based on the averaged MAPE values. The formula for MAPE is the following:

$$MAPE = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \cdot 100$$

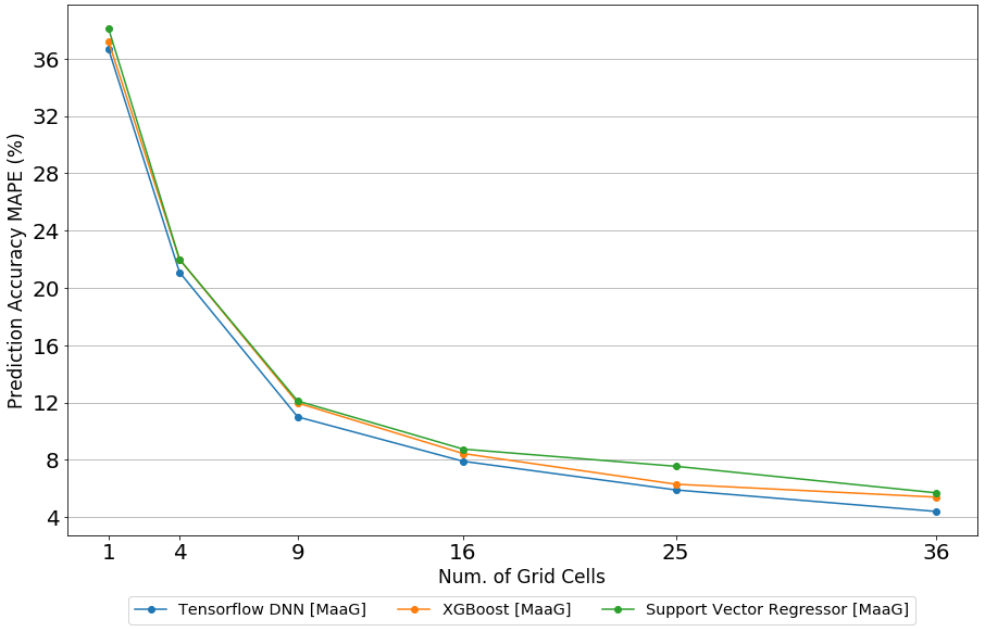


**Figure 16 Accuracy Mean Absolute Percentage Error based on the number of Grid Cells**

Figure 17 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 ofthe 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.

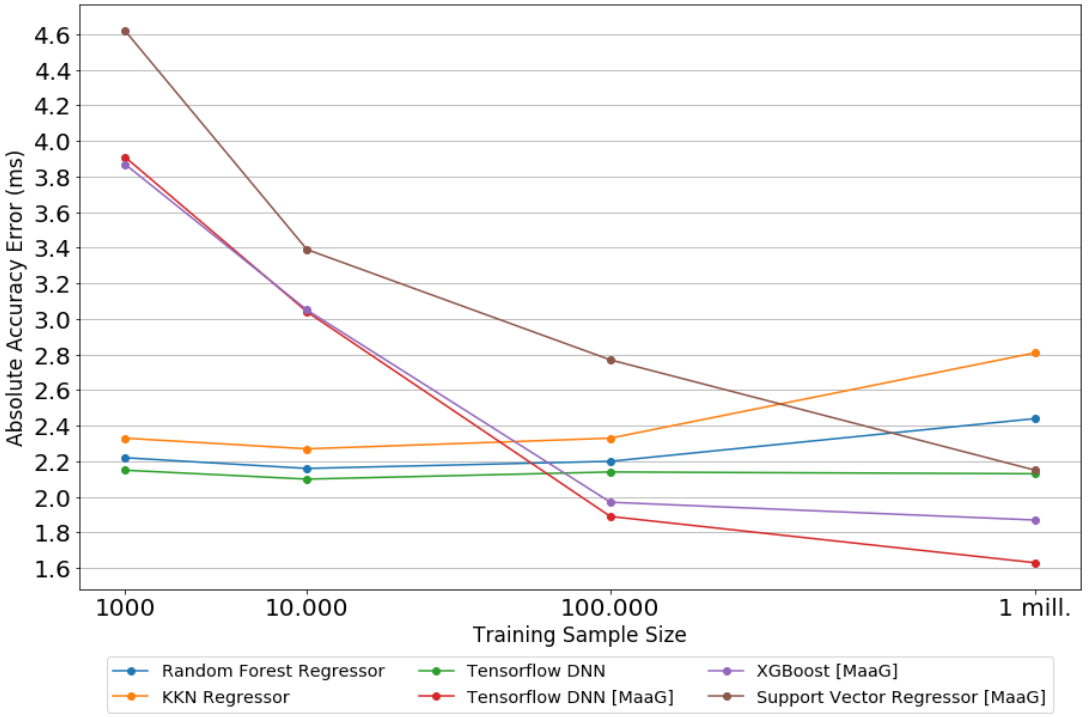


**Figure 17 Absolute Accuracy Error per Sample Size**

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.

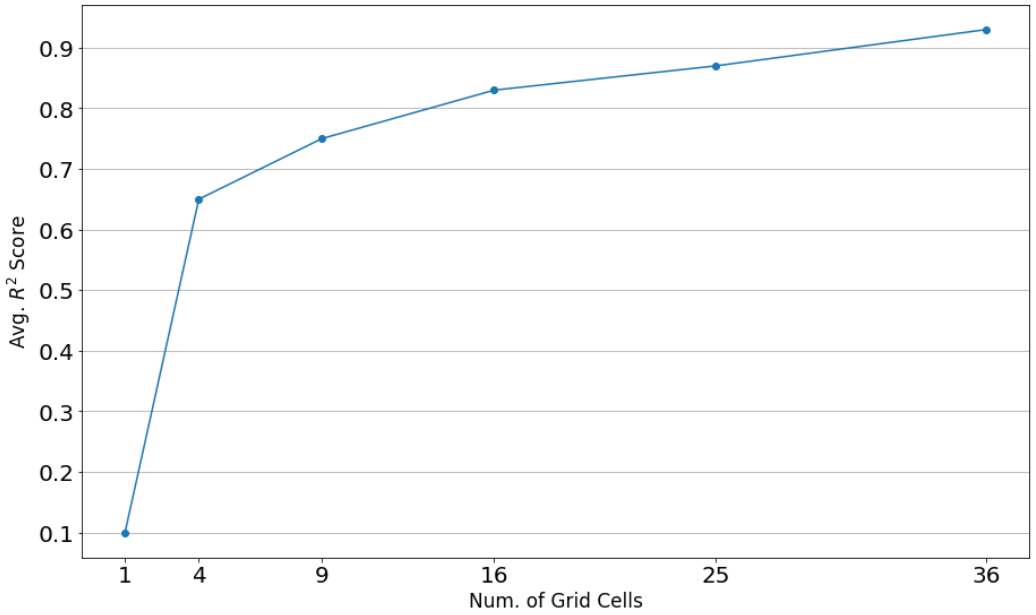


**Figure 18 Coefficient of Determination based on the number of Grid Cells**

The gains obtained by the MaaG model are shown when evaluating the coefficient of determination (as show in the equation below), or the coefficient of multiple correlation $R^2$ (Figure 18)

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

, 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'sa 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 18 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.

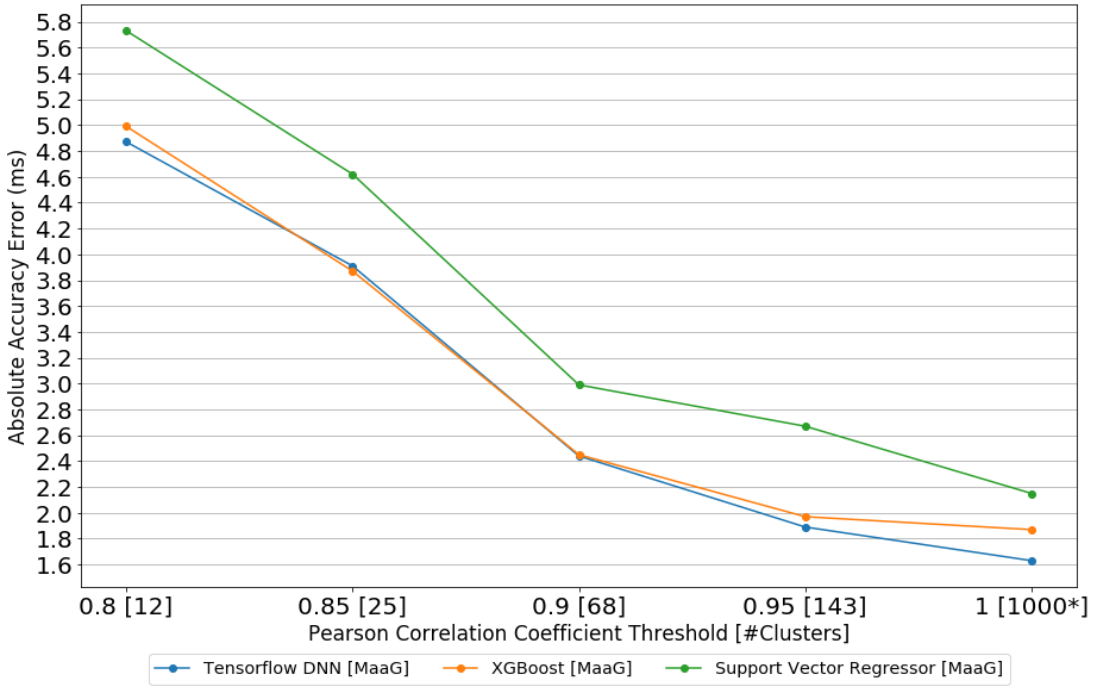


**Figure 19 Absolute Accuracy Error per Cluster Size**

For the evaluation of the last algorithmic step, i.e., the Grid Cell clustering using the Pearson Correlation method, Figure 19 illustrates the average prediction accuracy error for different clustering parameters. In more detail, the Pearson Correlation Coefficient (PCC) threshold is evaluated for different values, namely ranging from 0.80 to1. 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 more ”loose”, 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 0.8ms of accuracy error in absolute values). All in all, as it can be inferred by Figure 19, 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. CONCLUSIONS

This thesis presented 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. 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.

One of the important next steps of this thesis will be to further evaluate the tradeoff function that describes the MaaG granularity/prediction accuracy and clustering intensity relation, and perform an extensive evaluation of the different clustering approaches. New ML algorithms will be also integrated in the framework, primarily focusing on an LSTM-based solution, capable of performing online predictions, evaluating real-time incoming radio information and -in combination with the offline training solutions- ensemble Machine Learning and statistical approaches will be also explored. Last but not least,the proposed mechanism will be evaluated in the context of other well-established CAM use cases, such as the Anticipated Cooperative Collision Avoidance.

# ABBREVIATIONS - ACRONYMS

| | |
|---|---|
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| BS | Base Station |
| CAM | Connected and Automated Mobility |
| DNN | Deep Neural Network |
| k-NN | k - Nearest Neighbors |
| KPI | Key Performance Indicator |
| MAE | Mean Absolute Error |
| MAPE | Mean Absolute Percentage Error |
| MaaG | Map as a Grid |
| MEC | Mobile Edge Computing |
| ML | Machine Learning |
| MLP-BP | Multilayer Perceptron Back-Propagation |
| NB | Naïve Bayes |
| PCA | Principal Component Analysis |
| PCC | Pearson Correlation Coefficient |
| QoS | Quality of Service |
| RF | Random Forest |
| RF-R | Random Forest Regressor |
| RT | Ray Tracing |
| SiNR | Signal-to-noise ratio |
| SVM | Support Vector Machines |
| SVR | Support Vector Regressor |
| SUMO | Simulation of Urban Mobility |
| ToD | Tele-operated Driving |
| UC | Use Case |
| V2N | Vehicle to Network |
| V2I | Vehicle to Infrastructure |
| V2P | Vehicle to Pedestrian |
| V2V | Vehicle to Vehicle |
| V2X | Vehicle to Everything |

# ANNEX I

**Pearson Correlation:**
In statistics, the ***Pearson correlation coefficient*** (PCC), also known as Pearson's r, the Pearson product-moment correlation coefficient (PPMCC), the bivariate correlation, or colloquially simply as the correlation coefficient is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations; thus it is essentially a normalised measurement of the covariance, such that the result always has a value between −1 and 1. As with covariance itself, the measure can only reflect a linear correlation of variables, and ignores many other types of relationship or correlation. As a simple example, one would expect the age and height of a sample of teenagers from a high school to have a Pearson correlation coefficient significantly greater than 0, but less than 1 (as 1 would represent an unrealistically perfect correlation).

**Coefficient of Multiple Correlation:**
In statistics, the coefficient of multiple correlation is a measure of how well a given variable can be predicted using a linear function of a set of other variables. It is the correlation between the variable's values and the best predictions that can be computed linearly from the predictive variables.The coefficient of multiple correlation takes values between 0 and 1. Higher values indicate higher predictability of the dependent variable from the independent variables, with a value of 1 indicating that the predictions are exactly correct and a value of 0 indicating that no linear combination of the independent variables is a better predictor than is the fixed mean of the dependent variable. The coefficient of multiple correlation is known as the square root of the coefficient of determination, but under the particular assumptions that an intercept is included and that the best possible linear predictors are used, whereas the coefficient of determination is defined for more general cases, including those of nonlinear prediction and those in which the predicted values have not been derived from a model-fitting procedure.

**Standard Deviation:**
In statistics, the **standard deviation** is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

The standard deviation of a random variable, statistical population, data set, or probability distribution is the square root of its variance. A useful property of the standard deviation is that, unlike the variance, it is expressed in the same units as the data.

# REFERENCES


[1] 3rd Generation Partnership Project (3GPP), “Architecture enhancements for 5G System (5GS) to support network data analytics services (Rel.16),” Technical Specification (TS) 23.288, December 2019.

[2] ETSI, “5G System; Network Data Analytics Services,” Technical Spec-ification (TS) TS 129 520 V15.0.0, July 2018.

[3] 3rd Generation Partnership Project (3GPP), “System Architecture for the5G System (Rel. 16),” Technical Specification (TS) 23.501, December 2019.

[4] 3rd Generation Partnership Project (3GPP), “Architecture enhancements for V2X services (Rel. 14),” Technical Specification (TS) 22.185, June 2018.

[5] 3rd Generation Partnership Project (3GPP), “Enhancement of 3GPP support for V2X scenarios (Rel. 16),” Technical Specification (TS) 22.186, June 2019.

[6] 3rd Generation Partnership Project (3GPP), “Architecture enhancements for V2X services (Rel. 16),” Technical Specification (TS) 23.285, December 2019.

[7] 3rd Generation Partnership Project (3GPP), “Architecture enhancements for 5G System (5GS) to support Vehicle-to-Everything (V2X) services (Rel. 16),” Technical Specification (TS) 23.287, December 2019.

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

[9] 5GAA, “C-V2X Use Cases, Methodology, Examples and Service Level Requirements,” White Paper, July 2019.

[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 Vehic-ular Technology Magazine, vol. 13, no. 3, pp. 110–123, 2018.

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

[12] 5GAA, “Making 5G Proactive and Predictive for the Automotive Industry,” White Paper, November 2019.

[13] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante,W. Featherstone, and D. Munaretto, “Multi-access edge computing: The driver behind the wheel of 5g-connected cars,” IEEE Communications Standards Magazine, vol. 2, no. 3, pp. 66–73, 2018.

[14] ETSI, “Multi-access Edge Computing (MEC); V2X Information Service API ,” Technical Specification (TS) MEC 030 V2.1.1, April 2020.

[15] 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 Technology Conference (VTC2020-Spring), 2020, pp. 1–7.

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

[17] 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.

[18] 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.

[19] S. El Azzouni, E. Ekici, and N. B. Shroff, “QoS-aware predictiverate allocation over heterogeneous wireless interfaces,” in 2018 16th International Symposium on Modeling and Optimization in Mobile, AdHoc, and Wireless Networks (WiOpt), 2018, pp. 1–8.

[20] 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

[21] 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,” in2018 IEEE 5G World Forum (5GWF), 2018, pp. 93–98.

[22] 3rd Generation Partnership Project (3GPP), “Study on LTE-based V2X Services (Rel. 14),” Technical Specification (TS) 36.885, June 2016.

[23] 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

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

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

[26] 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.

[27] 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.

[28] 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

[29] 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.

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

[31] 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

[32] 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

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

[34] 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.

[35] 5GAA, “Architectural Enhancements for Providing QoS Predictability in C-V2X,” Technical Report, May 2020.

[36] 3rd Generation Partnership Project (3GPP), “Study on application layer support for V2X services (Rel. 16),” Technical Specification (TS) 23.795, December 2018.

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

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

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

[40] “EU H2020 5GCroCo project , Deliverable D2.1, Test Case Definition and Trial site Description Part 1,” 2016. [Online]. Available: https://5gcroco.eu/images/templates/rsvario/images/5GCroCoD21.pdf