ML-based QoE Estimation in 5G Networks Using Different Regression Techniques

Susanna Schwarzmann*, Clarissa Cassales Marquezan †,
Riccardo Trivisonno †, Shinichi Nakajima*‡§, Vincent Barriac¶,Thomas Zinner‖
* Technische Universität Berlin, 10587 Berlin, Germany
† Huawei Technologies, Munich, Germany
‡ Berlin Big Data Center, 10587 Berlin, Germany
§RIKEN Center for AIP, Tokyo, Japan
¶ Orange Innovation/Networks, Lannion, France
‖NTNU, 7041 Trondheim, Norway

***Abstract*—Monitoring and providing customers with a satisfying Quality of Experience (QoE) is a crucial business incentive for mobile network operators (MNOs). While the MNO is capable of monitoring a vast amount of network-related key performance indicators (KPIs), it typically does not have access to application-specific performance metrics. Among others, this is due to practical obstacles, such as missing standardized interfaces between the network and the application. Existing QoE models allow to map collected KPIs to the user-perceived quality. However, they are not dynamic, cumbersome to obtain, and often rely on application-level information, such as the stalling duration in the case of video streaming.**

The 5G networking architecture provides new features which can potentially overcome current limitations of in-network QoE monitoring. More specifically, the Application Function (AF) provides a standardized interface for communicating between 5G systems and third parties, such as application providers. The Network Data Analytics Function (NWDAF) is capable of collecting a vast number of network statistics from other 5G network functions and is dedicated to training and deploying Machine Learning (ML) models. This opens new possibilities, unimaginable for earlier mobile network generations, to dynamically learn the relationship between network KPIs and QoE by utilizing ML. Besides elaborating on how the new capabilities introduced with 5G can support an ML-based QoE estimation, we perform a simulation-based feasibility study which evaluates the estimation accuracy of different state-of-the-art regression techniques. In addition, we discuss them with respect to various qualitative aspects from an MNO's point of view.

***Index Terms*—5G, Mobile Networks, Machine Learning, QoE, HAS, VoD, VoIP**

I. Introduction

Providing a good (QoE) [1] to customers is of utmost importance to both, application providers (APs) and mobile network operators, to avoid user churn in the constantly growing market. APs, such as YouTube, have capabilities to monitor QoE-relevant application-level metrics, e.g., delivered video quality or interruption times, and to collect user ratings. An MNO, however, has only access to network-level data, unless it simultaneously acts as an AP. Although models exist, which allow a mapping from QoS metrics to QoE, they are typically only available for a limited number of services, cumbersome to design and update, and focus only on key network features or aggregated monitoring information. While they allow identifying certain root causes for service degradation, their general applicability for root cause analysis or fine-grained resource control is limited. Consequently, these traditional models cannot be used, for example, for self-driving networks, which are capable of implementing control-loops that automatically trigger QoE-aware network control actions.

With the introduction of the 5G networking architecture comes a vast number of new capabilities, among others supporting more intelligence in the network, potentially eliminating the current limitations of QoE monitoring activities carried out by MNOs. That is, (i) the interaction with externals, such as third party application or content providers, allowing to communicate ground-truth QoE or QoE-relevant metrics to the 5G system. Next, (ii) enhanced capabilities for data collection, processing, and exposure, which allow for correlating network KPIs and QoE information, i.e., training ML-based models. Finally, 5G facilitates (iii) the integration of ML into the network, such as deploying a trained model which is capable of obtaining the QoE from network KPIs. The 5G system is thus the first one to provide all necessary abilities for deploying ML-assisted QoE monitoring in the network at scale. While the collaboration between network and application providers has so far been hindered by the practical obstacle of missing interfaces and network entities [2], 5G overcomes this limitation by providing such a standardized interface. Moreover, its enhanced analytics capability guarantees the required computational resources and collects and supplies a sufficient set of statistics [3], potentially standardized, so as to support ML-based QoE estimation within multi-vendor networks.

In this work, we propose an approach exploiting these new features, i.e., the interaction with externals, the sharing of information between network functions (NFs), and the enhanced data processing capabilities. We discuss the relevant research questions for introducing ML-based QoE estimation into the network from the viewpoint of an MNO and perform a simulation-based feasibility study to show the applicability of such an approach with the data available in 5G systems. In the scope of this study, we analyze the performance of a representative set of five different regression techniques, ranging from simple linear models to highly complex neural networks. Thereby, we also retrieve potential factors influ-

encing the estimation accuracy in the heterogeneous mobile environment, where users run different applications and have varying mobility patterns. Moreover, we analyze the ML techniques' complexities in both, a theoretical and a practical manner, and compare their resource requirements as well as the duration to train and test them. Finally, we discuss the set of regression techniques in a qualitative manner, taking into account factors such as their comprehensibility or provided built-in features.

The rest of this work is structured as follows. We introduce the used regression techniques and present related work in Section II. Next, we describe in Section III how 5G can support ML-driven QoE estimation and how we address the research questions focused in this manuscript. Afterwards, we present the applied methodology in Section IV and describe our ground-truth data set in Section V. The evaluation results are given in Section VI, followed by a discussion on the study's limitations and the lessons learned in Section VII. Finally, Section VIII concludes the paper.

II. Background and Related Work

This section introduces the key working principles of the studied regression techniques and presents related works in the field of ML for QoE estimation and for mobile networks.

A. Background on Regression Techniques

Regression models learn relationship between *features* and (non-binary) *responses* from training data, so that they can predict unknown responses from features at deployment. Features can be raw observations or their predefined transformation. In our case, the features are all network-related variables, transformed using statistical metrics, and made available at the NWDAF, while the response is the QoE expressed as Mean Opinion Score (MOS), a scalar variable in a range from 1 to 5. We evaluate two linear models, where the response is expressed as a linear function of the features, and three non-linear models. We briefly introduce the applied regression techniques in the following.

Least absolute shrinkage and selection operator (LASSO) [4] is a linear regression model trained with the L1 regularizer, the sum of absolute values of the weights (or the regression coefficients), in addition to the mean squared error (MSE) of the prediction. The L1 regularizer is known to induce *sparsity*, meaning that many of the learned weights are zero, and therefore the corresponding features are completely neglected when estimating the response (QoE in our case). The number of zero weights can be tuned by the regularization parameter λ, which controls the strength of shrinkage and sparsity. By doing so, LASSO does not only help to reduce over-fitting, but can also be used for feature selection, as done in our previous work [5]. This can also help to make the model less complex and thus easier to understand by humans.

Linear Ridge Regression (LRR) [6] is a linear regression model trained with the L2 regularizer, the sum of the squares of the weights. While its working principle in general is corresponding to that of LASSO, the L2 regularizer shrinks the weights to avoid overfitting, but it hardly shrinks any weight to zero. It cares more about driving big weights to small values, and tends to give small but well distributed weights. By doing so, LRR tends to provide better prediction accuracy than LASSO, while it cannot be directly used for reducing the number of used features.

Depending on the complexity of the data, linear models might not be sufficient to capture the peculiarities of the problem. In such cases, non-linear regression models such as the **Kernel Ridge Regression (KRR)** [6] should be used. KRR is a kernel regression model trained with L2 regularizer. It is equivalent to the linear regression applied to a high (possibly infinite) dimensional space into which the original features are non-linearly mapped. By choosing an appropriate kernel, this model can approximately express any smooth function and thus learn non-linear relationships between features and response. The kernel trick allows to operate in the original feature space without computing high dimensional mapping, offering a more efficient and less expensive way to non-linearly transform features into high dimensional space. We use the Gaussian kernel with the bandwidth parameter γ.

Support Vector Regression (SVR) [7] is a similar model to KRR, but trained with the ϵ-insensitive loss, instead of the MSE. The parameter ϵ defines an acceptable error margin - the errors smaller than the margin are ignored during training, resulting in a small number of support vectors. We can tune ϵ to achieve the required accuracy. Since the ϵ-insensitive loss is less sensitive to outliers than MSE, SVR is robust against outliers. Similar to KRR, we apply SVR with the Gaussian kernel to be capable of modeling non-linear relationships. Models learned with SVR are sparse and thus typically faster when they are deployed, compared to the non-sparse models learned with KRR.

Neural Networks (NN) [6] are models consisting of artificial *neurons*. A neuron converts a given input to the output by applying a linear transformation with learned weights and then a non-linear transformation, called *activation*. Typically, thousands of neurons form a *layer*, and multiple layers are stacked, where the features correspond to the input of the first layer and the responses correspond to the output of the final layer. The non-linearity of the activation allows to model non-linear relationship between the features and the responses. Actually, NNs with just a single intermediate layer (with sufficiently many nodes) can approximately express any function. Deep NNs with many layers showed excellent performance in many classification and regression tasks, where the raw data (e.g., natural images) without any manual feature engineering are directly fed into the network [8]. This is possible because the first layers (the most layers except the few final layers) work as an automatic non-linear feature extractor, if a deep NN with a reasonable architecture is trained appropriately. The advantages coming with a deep NN are the autonomous generation of features from raw input and their optimal tuning, which are transferable to related problems.

B. Related Work

1) Estimating Video QoE from Encrypted Traffic: With the increasing adoption of network encryption, in-network

QoE estimation is facing new challenges, as conventional approaches, such as deep packet inspection (DPI) are not applicable anymore. To overcome the issue, several approaches have been proposed in literature to estimate the QoE from encrypted network traffic. The work in [9] presents a framework which is able to extract the key influence factors for video streaming QoE, namely stallings along with their duration, the visual quality as determined by the resolution and video encoding bitrate, as well as the quality fluctuations throughout the play back. Based on a ground truth data set comprising nearly 400,000 streaming sessions, the authors train a Random Forest algorithm to classify the respective QoE influence factors into a set of three pre-defined classes each. A similar approach is followed in [10] for YouTube video streaming. In addition to classifying specific QoE influence factors, such as stallings and average quality, the authors also classify the video sessions according to their overall QoE score. Therefore, the authors train seven different types of classification models and evaluate their performance by means of four different data sets, gathered from a controlled WiFi lab environment as well as from an operational mobile network. Besides the high accuracy that can be achieved by the classification models, the work also shows that despite the models have been trained in a lab environment, they are still applicable to operational networks.

2) QoE Estimation in (5G) Mobile Networks: Mobile video streaming is getting more and more popular. Due to additional network-related KPIs, such as the channel quality, and additional characteristics of clients, e.g., their movement, QoE assessment in mobile environments needs dedicated evaluations. Therefore, [11] focuses on mobile networks for predicting whether stalling occurs during video streaming. They train a Generalized Linear Model (GLM) as well as a Support Vector Machine (SVM) to perform a prediction based on the wireless channel conditions and the number of active users. Thereby, the authors distinguish moving and static users and find that predictions are harder to perform for those users, who are moving during streaming. The movement of users for estimating the QoE is also addressed in our previous work [5], which shows that the relevance of features differs for moving and static user equipments (UEs). For moving ones, features expressing variability gain importance and as obtaining those features requires to monitor the network with finer granularity, the costs for QoE estimation could increase with user movement.

Another study on video QoE estimation using Machine Learning in mobile networks is presented in [12]. The authors rely on in-smartphone measurements describing the radio link of the LTE connection. Accordingly, the feature set includes statistics related to, e.g., the Channel Quality Indicator (CQI), Reference Signal Received Power (RSRP), or the Carrier to Interference Noise Ratio (CINR). Using these statistics, a Random Forest model is trained in order to obtain the MOS and different QoE influence factors, such as a video's blurriness or frame skips. Besides the evaluation of the trained model's accuracy, the authors study the correlation between the different features and the MOS, and additionally perform a root cause analysis to understand the model's decisions.

3) Usage of New 5G NFs: The exploitation of the newly introduced NFs in 5G systems for intelligent networking has recently been proposed in literature. For example, it is examined how the NWDAF can be used for predicting abnormal as well as expected behavior for a group of UEs, and for forecasting the network load in an area of interest [13]. The proposed architecture connects the NWDAF with other NFs via the Service Based Interface (SBI) to allow mutual data transfer. Using both, time series data and generated features available at the NWDAF, different ML models are examined with respect to their feasibility for the given problems. The conducted study shows that NNs outperform LRR models when it comes to network load prediction and that tree-based XGBoost yields better classification performance compared to logistic regression in the anomaly detection use-case.

Due to the huge amount of devices expected to be connected to 5G systems, moving all data to a centralized unit for analytics is inefficient. The need for running ML algorithms in a distributed manner is discussed in [14]. Only then, a fast decision making which minimizes the network response time to user requests and which fulfills the latency requirement of 5G, can be guaranteed. The outlined proposal for a distributed analytics architecture considers one centralized NWDAF instance and several distributed NWDAF instances, which can be co-located with other NFs and only collect data gathered from that co-located NFs. Our previous work [15] proposed a work flow for integrating ML-based QoE estimation in 5G networks by means of utilizing the NWDAF and the AF. Thereby, the latter is used to communicate ground-truth QoE information from a third party application provider to the MNO. The NWDAF, on the other hand, collects and processes network telemetry data and trains an ML algorithm, so that at later stages, the QoE can be estimated from the KPIs monitored in the network.

The studies carried out in this manuscript differ from previous works with respect to several distinct aspects. Instead of estimating the value of specific QoE influence factors, our goal is to estimate the overall MOS score of a completed session. While this has been addressed in some previous works as well, we want to emphasize that we do not make use of classification, but solve a regression problem. Instead of categorizing the QoE into a pre-defined, limited set of classes, we estimate real numbers in the continuous range from 1 to 5. This allows for a wider range of use-cases, including those where the QoE needs to be available on a fine-grained level. In terms of the used features, we extend previous works by combining the usage of radio-related statistics with end-to-end flow statistics, instead of only using either one or relying on very fine-grained packet-level statistics that are hard to collect at scale. Our study is based on those monitoring data, that can actually be collected by an MNO within the 5G system according to 3GPP specifications. Apart from that, our evaluations go beyond estimation accuracy by including further qualitative and quantitative aspects which are relevant from an operator's point of view, such as the computational complexity, the trackability, or the comprehensibility of ML models. Finally, we note that the generalizability of our results are enhanced by considering two distinct service types, i.e.

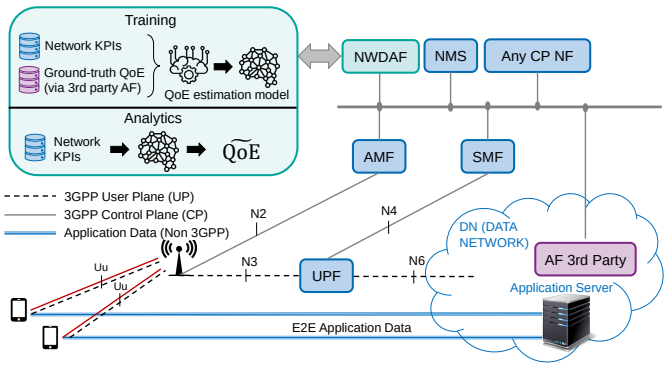


Fig. 1: Integration of AF and NWDAF in the 5G architecture to support third party information exchange and data analytics.

Voice over IP (VoIP) and Video on Demand (VoD), which have significantly different QoS-QoE relationships.

III. Contributions

In the scope of this manuscript, we first propose the usage of two new NFs introduced with 5G, dedicated for data analytics and third party information exchange, so to enable an ML-based QoE estimation. In a next step, we conduct a practical study showing the feasibility of such an approach in mobile networks and compare the suitability of different regression techniques in both, a quantitative and a qualitative manner.

A. Data Analytics-driven QoE Estimation in 5G Networks

Based on Figure 1, we describe two newly introduced NFs, which are relevant for our work. The first one is the AF, a 5G core network function, which can be owned and customized by third parties. It is connected to the SBI, and thus allows the communication between 5G control plane NFs (owned by the MNO) and content or application providers, e.g., YouTube or Netflix, in a standardized manner. The second one is the NWDAF, which is also connected to the SBI and capable of collecting and processing statistics from other 5G control plane NFs, i.e., the Access and Mobility Management Function (AMF), Session Management Function (SMF), or the Network Management System (NMS). Besides, the NWDAF can also obtain data from third parties via the AF, e.g., information about the QoE. As the entity where all information is accumulated, the NWDAF can be seen as the brain of the 5G system, where ML algorithms may run and which provides intelligent services to all NFs by exposing analytics, which other NFs can invoke. Dedicated to an ML deployment, the NWDAF can be decomposed into two logical functions: *training* and *analytics*. The NWDAF which holds the logical function for *training* takes care for the development of the model, including is initial training and regular re-training with new training data, to keep it up to date. The NWDAF containing the *analytics* function applies the trained model (i.e., it estimates the QoE in our case).

In the scope of this manuscript, we assume the following workflow: The AP, which is aware of any application-related information, provides the QoE scores via the AF. The MNO, which can collect any network-related KPIs from the NFs connected via the SBI, statistically processes the monitored data at the NWDAF, i.e., it generates features from the monitored data. The logical function of the NWDAF dedicated to *training* is applied to learn the relationship between the generated features and the QoE. Once the model is capable of obtaining a satisfactory estimation accuracy, it can be deployed via the NWDAF's logical *analytics* function, allowing the MNO to assess the QoE also in the absence of information provided by the AP via the AF. Such a mechanism consequently allows the MNO to perform QoE monitoring from its own collected network telemetry data.

B. Research Questions and Practical Study

An MNO planning to integrate ML-based QoE estimation into its network is faced with a variety of different challenges and design decisions. As there is no one-size-fits-all solution, we elaborate in the following on the most important involved questions and show how they are addressed within our feasibility study by means of Figure 2, which classifies our contribution with respect to the three major phases *data collection*, *model training and testing*, and *deployment*.

The top of the illustration denotes our proposed concept as introduced in Section III-A, exploiting new 5G NFs for an ML-driven QoE estimation, thus referring to the generic question: **How can an MNO exploit new features provided in 5G networks, so to estimate the QoE based on the available network KPIs?** The blue squares denote the network KPIs obtained during the *data collection phase* at different instances (1), e.g., at the UE or the Access Node (AN), which are statistically processed to generate features. Additionally, during this phase, true QoE scores are provided via the AF (denoted as the purple box) from a third party AP. In the course of our practical study, we gather these types of data by means of network simulations (2). Thereby, we consider two distinct applications, i.e., VoD and VoIP, and take moving as well as stationary clients into account. In this respect, the following question needs to resolved: **Which features need to be provided at the NWDAF?** Based on our generated data set, we address this question by studying the correlation between the network-related features and the QoE for both service types to elaborate on the application-specific relevance of different features (3).

During the *model training and testing phase*, the logical NWDAF function for *training* learns the relationship between the network KPIs and the QoE (4). The field of ML offers a huge amount of individual techniques, differing with respect to their training process, i.e., supervised vs. un-supervised learning models, as well as in terms of their respective prediction target, which can be clusters of given data points, a categorization of input samples into a pre-defined set of classes, or actual numbers covering a continuous range. Furthermore, the available models range from very simple approaches, such as linear regression techniques, to highly advanced and complex approaches like artificial neural networks. Which one to choose is a crucial design criterion and depends on numerous factors. Due to the vast amount of existing ML

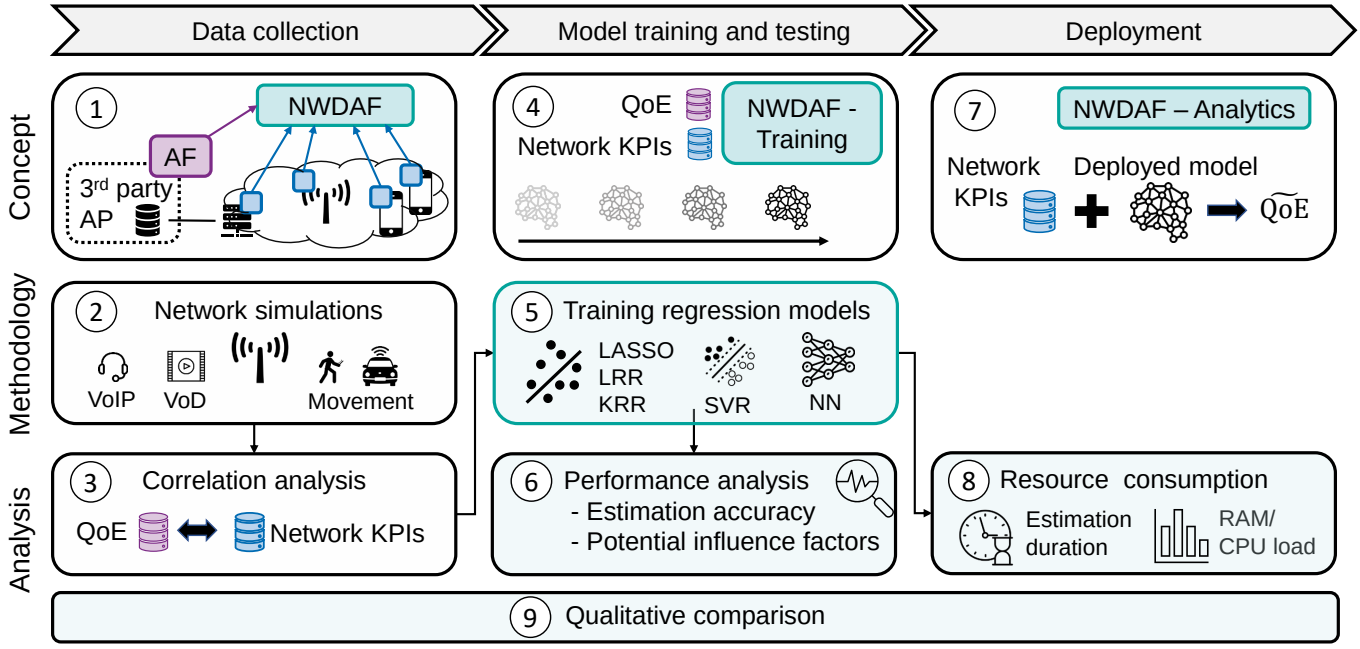


Fig. 2: Contributions of this manuscript with respect to the three phases data collection, model training and testing, and deployment. Light green boxes denote elaborations carried out on the example of the five regression techniques.

techniques, an MNO is faced with the following question: **Which ML models should be trained and considered for potential deployment?** The suitability of a certain option depends on a variety of different factors.

Firstly, the purpose of the QoE estimation reduces to a limited set of potential ML algorithms for deployment. Different use-cases have different demands, e.g., in terms of the estimation speed or the level of estimation granularity and accuracy. If only a rough estimation of the system's QoE is of interest, it can be sufficient to run simple classification algorithms using coarse-grained pre-defined classes. If the QoE estimation should be used for a fine-grained QoE-aware, real-time resource control on a per-flow level, the requirements on estimation speed, accuracy, and granularity are significantly higher. In our feasibility study, we assume that the MNO's goal is to estimate the QoE on the continuous MOS scale. Accordingly, we train regression techniques ⑤, which are capable of predicting any continuous number in the given range between 1 and 5. More specifically, we consider a representative set consisting of the five different regression techniques, as introduced in Section II: LASSO, LRR, KRR, SVR, and NN. Compared to classification, which uses pre-defined classes, e.g., low, medium, and high QoE, regression models yield a finer granularity of the predicted output and hence allow to cover a wider range of possible use-cases, including reporting, on-demand troubleshooting, or automatic corrective actions.

Secondly, the estimation accuracy is one of the key factors when determining the algorithm to deploy. Hence, it is crucial to evaluate how reliably an ML option performs. As today's networks are very heterogeneous in many terms, potential influence factors on the estimation accuracy need to be examined. This includes for example the movement patterns of users or the service type, for which the QoE should be estimated. To account for this, we conduct a general performance comparison ⑥ of the ML techniques and additionally elaborate in how far the service type, i.e., VoIP versus VoD, the movement pattern of clients, or the true QoE, i.e., low, medium, or high, potentially impact their estimation accuracy.

According to the proposed concept, during *deployment*, the logical function of the NWDAF which is dedicated to *analytics*, holds a trained ML model. Thus, based on the network KPIs, it can estimate the QoE ⑦. As a third point when deciding about an ML option, an MNO should take the computational and temporal overhead into account to ensure that the mechanisms scale and can be applied efficiently to not overload the responsible NF. For instance, the MNO needs to be aware of the resource exploitation during training and testing, as well as of the duration until an estimate is available. In the scope of our feasibility study, we address this point by analyzing the resource consumption, as well as the time it takes the regression techniques to obtain the QoE estimates ⑧.

Fourthly, there are different qualitative aspects an MNO should consider when deciding about which ML algorithm to deploy. This can be, for example, the comprehensibility of a model or the required efforts for maintaining it. The Internet ecosystem is highly dynamic, leading to the fact that a once trained and deployed model will lose its validity with ongoing changes in the system. For example, the usage of a new voice codec in a VoIP service will affect the relationship between the measured network KPIs and the resulting QoE. The associations as learned by the original model might hence be obsolete. Accordingly, it is advisable, or even mandatory, for the MNO to still collect ground-truth data for updating and re-validating a deployed model. By means of regular sanity checks and re-training, the MNO needs to ensure that

the model is still adequate, or – if necessary – deploy an updated version to account for system changes. Consequently, a deployed model can evolve over time, which potentially raises another challenge if the MNO aims to keep track of the evolution of the model. To account for those qualitative aspects, we additionally perform a comparison of the different regression techniques with respect to factors such as their comprehensibility, trackability, or whether they provide built-in features, e.g., supporting over-fitting prevention or feature selection ⑨.

IV. Methodology

The following section introduces the applied methodology. We first detail on the simulation environment used to generate the ground-truth data set and afterwards describe how the ML models are trained.

A. Simulation Scenarios and Ground-Truth Data Collection

This section first presents our simulation environment based on the discrete event simulator OMNeT++. Afterwards, we describe the considered applications and the QoE models applied to retrieve the true QoE on MOS scale. Next, the ground-truth data collected within the experiments and the evaluation scenarios are presented.

1) OMNeT++ Simulation Environment: For generating the network- and application-related ground-truth data, we rely on OMNeT++ [16] and integrate the frameworks INET and SimuLTE [17] into our simulation setup.[1] Although SimuLTE simulates 4G networks, the type of monitored information is the same with 5G networks. For instance, the equivalent of 4G Packet Gateway (PGW) and eNB data is information collected from UPF and gNB in 5G. At this stage of our research, the main point of using SimuLTE is to obtain monitoring information from both, access and core network. As we assume the monitoring information to be available at NWDAF and do not consider any signaling exchange for data collection yet, SimuLTE can be used for our purpose of generating user plane traffic in a mobile network. From the perspective of the radio technology, 5G has much higher data rates compared to 4G. However, the principles of system load (number of UEs in a cell) and radio quality will still play a role in 5G systems. The number of connected devices will be significantly increased within 5G systems, which is mainly due to massive IoT and the consequential enormous communication between sensors. Please note that this circumstance is no limitation to our conducted study, because the general relationship between network KPIs and the resulting QoE for our set of applications will not be affected by the increased connectivity.

Our simulations consist of a single AN which serves a varying number of active UEs, which differ with respect to their mobility characteristics. We consider *static* clients, which do not move throughout the experiments, and *moving* clients, which walk (3 kmph) or drive (50 kmph) between different points of interest (POIs). To simulate realistic movement patterns, we use the small world in motion (SWIM) movement model [18].

[1]https://github.com/fg-inet/ML-simulation-and-ground-truth

2) Considered Applications and QoE Models: For each service type, we apply existing QoE models, which allow computing the QoE based on collected network- and/or application-related KPIs. Please note that we assume in this work that the AP provides the QoE, e.g., on MOS scale, to the MNO. Having this information and the network telemetry data, the MNO can build its own ML-based model, which allows – once the model is trained so to retrieve a reliable estimation – for obtaining the QoE only based on network-related KPIs. That is, we skip the step of applying QoE models which require application-level information, such as the stalling duration, and obtain the QoE directly from the collected network statistics.

The VoD client implements HTTP Adaptive Streaming (HAS) [19], where the video content is split into short segments and made available in different qualities. This allows the client to dynamically select the next video segment's quality based on throughput measurements or the buffer state. We consider segments of five seconds each and a total video duration of 400 seconds, resembling the average session duration of popular VoD platforms, such as dailymotion or hulu[2]. The video is made available in four different qualities, comprising bitrates of 500 kbps, 1 Mbps, 1.5 Mbps, and 3 Mbps. The VoD client applies a buffer-based heuristic, determining the next segment's quality based on the video buffer's filling level. The lowest quality level is chosen if the buffer is below 10 seconds. The second, third, or fourth quality level is chosen if the buffer exceeds the threshold of 10 s, 20 s, or 30 s, respectively. To not limit the results to a specific QoE model, we apply two different existing models for obtaining the QoE from application-specific metrics. The first one is the standardized ITU-T P.1203 model [20]–[22], where we consider mode 0 (without bitstream data or fame-level details) and only the visual part, i.e., we apply the P.1203.1 model. The second one is the cumulative quality model (CQM) [23]. In the following, we will refer to these models as *VoD-P.1203* and *VoD-CQM*. Both take the per-segment video bitrate, the number of stallings, their durations, and the initial delay into account. Furthermore, the models consider the frequency and amplitude of quality changes. The QoE is computed on a per-session level, i.e., after the full video clip has been played back. CQM allows to set different parameters, such as weights for quality window metrics. In this work, we use the default values from the implementation's repository[3].

For VoIP, we only model the receiver side, i.e., the listener of the conversation. The talk spurt duration (seconds) follows a Weibull distribution with a scale of 1.4 and a shape of 0.82. Between the talk spurts, there is silence, which also follows a Weibull distribution (0.899, 1.089). We use the g726 codec and a bitrate of 40 kbps. Although better voice codecs exist, the voice processing does not have to be representative of reality, but at least allow quality variation all along the available range, so that training and testing data can be performed to compare the ML techniques under study. The QoE of VoIP clients is computed using the E-model as defined in the ITU-

[2]https://www.statista.com/statistics/910910/us-most-popular-us-video-streaming-services-session-duration/

[3]https://github.com/TranHuyen1191/CQM

TABLE I: Collected network KPIs and applied statistics for feature generation. Check marks denote that the monitoring information is collected for the specific application.

Network KPI	**Abbreviation**	**VoD**	**VoIP**
Access node throughput on uplink	AN TP UL	✓	✓
Access node throughput on downlink	AN TP DL	✓	✗
User equipment throughput on uplink	UE TP UL	✓	✗
User equipment throughput on downlink	UE TP DL	✓	✓
Channel quality indicator on downlink	CQI DL	✓	✓
Channel quality indicator on uplink	CQI UL	✓	✗
TCP round trip time	RTT	✓	✗
End to end delay	E2E delay	✗	✓
Radio link control delay	RLC delay	✗	✓
Hybrid automatic repeat request error rate	HARQ ER DL	✗	✓
Applied Statistics	**Abbreviation(s)**		
Average / minimum / maximum / median	avg / min / max / median		
Standard deviation/ variance	std / var		
25th percentile / 75th percentile	25perc / 75perc		
Coefficient of variation / kurtosis / skewness	cov / kurt / skew		
Unbiased standard error of the mean	sem		

T G.107 standard [24]. Thereby, a MOS value is computed after each talk spurt. As a client's overall QoE score for the whole VoIP session, we consider the average MOS of all its talk spurts during the session. Newer models, such as the ITU-T P.863 (POLQA) are capable of better reflecting the user's perceived voice quality. POLQA is a full reference metric, i.e., it needs a reference audio sample to compare it against the received audio signal to obtain the MOS. However, due to the simulative nature of our approach, such full reference metrics cannot be applied in our case. Nevertheless, the general relationship between the MOS and its impacting factors, i.e., delay and packet loss rate, is the same for both models [25], [26]. Hence, we can expect a similar performance of the applied ML models, independent of the specific QoE model applied.

3) Collected Ground-Truth Data and Features: With the term *ground-truth* data, we refer to those data points, where both, network-related information corresponding the VoIP or VoD session and its associated true QoE score are available. While the estimated QoE is the output of the trained ML model, the true QoE is reliably retrieved from the applied QoE models. Our monitored network information for the different application types are summarized in Table I. All of the monitoring data is collected as time series on a per-second scale granularity. To generate the features, which are the input for the ML models to estimate the QoE, we apply the twelve different statistics shown in the table to the per-session time series. Hence, for one complete video stream or one VoIP call, there is exactly one value for the respective combination of monitored data and statistic, i.e., feature. For example, in the case of VoD, where the twelve statistics are applied to the seven collected network KPI time-series, we obtain in total 84 feature values used for estimating the QoE of that specific video session.

Please note that in the case of VoIP, where we only consider the transmission from sender to receiver, there is no up-link related monitoring information. The only exception is the AN throughput, whose UL throughput reflects the sum of DL throughput of all active UEs. We furthermore collect two different types of delays in the case of VoIP: the end-to-end delay and the Radio Link Control (RLC) delay. The end-to-end delay as obtained in our simulation is an in-app measurement and possibly hard to obtain by the MNO. However, the MNO is in any case capable of collecting the RLC delay in a straightforward manner from the AN on a per-UE basis.

The 3GPP specification TS 28.552 for 5G performance measurements [27] denotes how and where the different monitoring data can be obtained within the 5G system. We want to emphasize that not only are all of the KPI measurements used throughout our study available, but that they can be obtained on an even more detailed and comprehensive level. For example, the smallest scale for monitoring the UE downlink throughput is defined by one HARQ transmission and besides the RLC delay, an MNO could consider the delay occurring along the different entities, e.g., between the RAN and the UE or between the RAN and UPF.

4) Evaluation Scenarios: In order to obtain QoE values exploiting the MOS range as good as possible, we run simulations with a slight overload of the cell's capacity. We consider 80 and 160 active clients for VoD and 400 for VoIP, as VoIP clients consume significantly less bandwidth compared to VoD clients. The clients are either moving or static and we set the following distribution with regard to the UEs' mobility patterns: 100% static, 100% moving, or half moving half non-moving. To vary the impact of the clients' movements on the network KPIs, we configure the following POI settings: A single POI, either located at the edge of the cell or very close to the AN, as well as 10 or 100 existing POIs, which are randomly placed within the cell. Finally, we simulate each configuration with different seeds, which determine the initial placement of the UEs.

B. Training of Machine Learning Models

We split the set of ground truth-data samples into a *training set*, which contains 80% of the data points, and a *test set* for the remaining 20%. The training set is used for training the ML models and tuning their hyper-parameters, i.e., those parameters which are not set upfront, but optimized during the training phase. For tuning the hyper-parameters with LASSO, LRR, KRR, and SVR, we apply a grid search. When tuning the NN, we use the optimizer Adam [28]. During training, we apply a 5-fold cross-validation, i.e., the training is repeated 5 times. In each of the five rounds, a *validation error* and a *training error* are obtained. A high validation error in combination with a low training score indicates an over-fit of the model, while a high validation error in combination with a high training error reveals an under-fitting model. The sweet spot is located where the validation error is low, with a moderate training error. The hyper-parameter set which maximizes the validated performance (averaged over the 5 folds) is chosen for the final model, which is then applied to the *test set*, which only contains samples the ML model has not seen so far, i.e., none of the samples of the test set were used to train the model. By applying the model to the test set, we obtain the *test error*, which is used for reporting the final performance. Consequently, when evaluating a model's accuracy in the following parts of this work, we are referring to the *test error*.

TABLE II: Tested parameters for the different ML algorithms. The selected parameters are those which yield the highest estimation accuracy upon the tested parameters and are used in the evaluation chapter.

			VoD			**VoIP**		
	Parameter	*Description*	*Tested*	*# comb.*	*Selected*	*Tested*	*# comb.*	*Selected*
LASSO	λ	Regularization	$[0:0.1]$ ($\lvert\lambda\rvert = 21$)	21	0.0005	[0:0.1] ($\lvert\lambda\rvert = 21$)	21	0
LRR	λ	Regularization	$[0:500]$ ($\lvert\lambda\rvert = 28$)	28	10^{-9}	$[0.1:600]$ ($\lvert\lambda\rvert = 28$)	28	0.1
KRR	λ	Regularization	$[0:10]$ ($\lvert\lambda\rvert = 23$)	23	0.07	$[0:1]$ ($\lvert\lambda\rvert = 23$)	23	0.01
SVR	C	Regularization	$[0.1:6]$ ($\lvert C\rvert = 16$)	32	4.92	$[7:12]$ ($\lvert C\rvert = 10$)	60	12
	ϵ	Acceptable error	$[0.01, 0.1]$		0.01	$[0.01:0.1]$ ($\lvert\epsilon\rvert = 6$)		0.03
NN	LR	Learning rate	$[0.001, 0.01, 0.1]$	504	0.001	$[0.001, 0.01, 0.1]$	504	0.01
	$\#epochs$	Number of epochs	$[100, 200, 500, 1000]$		1000	$[100, 200, 500, 1000]$		1000
	$\#neurons$	Number of neurons	$[10, 50, 200]$		50	$[10, 50, 200]$		50
	$batchS$	Batch size	$[10:1000]$ ($\lvert batchS\rvert = 7$)		200	$[10:1000]$ ($\lvert batchS\rvert = 7$)		200
	AF	Activation function	tanh, relu		relu	tanh, relu		relu

TABLE III: Number of samples and average MOS $\pm$ standard deviation for the different service types and data sets.

	VoD-P.1203		*VoD-CQM*		*VoIP*	
Dataset	**Size**	**MOS**	**Size**	**MOS**	**Size**	**MOS**
all	13952	2.18±0.60	13952	1.65±0.79	27592	4.17±0.49
stationary	3210	2.41±0.82	3210	1.97±0.98	8392	4.04±0.67
moving	10742	2.12±0.49	10742	1.55±0.70	19200	4.23±0.37
low	8095	1.83±0.07	10139	1.89±0.13	291	1.21±0.24
medium	5067	2.46±0.39	3295	2.68±0.35	2136	2.97±0.40
high	790	4.02±0.36	518	3.74±0.16	25165	4.31±0.18

Table II summarizes the various settings tested for each respective parameter of an ML option. For the NN, we consider a single hidden layer. The table further denotes those parameter settings, that have been chosen for the QoE estimation for the different service types. The chosen parameters are those which yield the highest estimation accuracy, i.e., the lowest validation error expressed as MSE, and are used during the performance comparison in the evaluation chapter.

V. Ground-truth Data Set Analysis

This section presents the characteristics of the ground-truth data set. We first detail on the number of collected samples and show the distributions of the ground-truth QoE for VoIP and VoD. Next, we study the correlations of the network-related features with QoE and exemplarily show the relation between the most powerful features' values and the respective MOS scores.

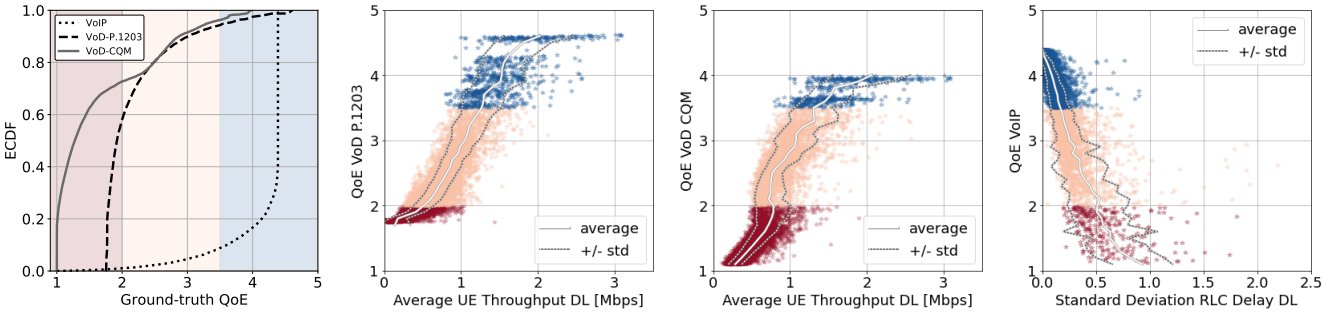
A. Ground-truth QoE Distribution

In order to evaluate the ML performance for different true QoE scores and movement patterns, we divide our ground-truth data set into subsets. We distinguish between *stationary* and *moving* clients, as well as between *low* ($MOS < 2.0$), *medium* ($2.0 \leq MOS \leq 3.5$), and *high* QoE ($MOS > 3.5$). Table III denotes the number of ground-truth samples and the average MOS values along with their standard deviations for each service type in the different subsets.The average QoE of the VoD clients is higher when using the ITU-T P.1203 model (2.18), compared to using CQM (1.65). With both models, *stationary* clients obtain a higher QoE than the *moving* clients. In the moving clients data set, we observe a lower standard deviation compared to that for stationary clients. This can also be observed for VoIP, where moving clients have a standard deviation of 0.37, while for the stationary ones, the standard deviation increases to 0.67. Contrary to VoD, moving VoIP clients obtain a slightly higher QoE than stationary ones. This is counter-intuitive, since mobility is a well known degradation factor in telecommunications. This issue nevertheless does not pose a problem in terms of training and testing the ML models, as the global distributions of MOS scores, and hence the training of the models, are not biased.

The QoE distributions for the different services are denoted in Figure 3a. The x-axis shows the ground-truth QoE scores, the y-axis the ECDF. While for the VoD clients, about 60% (P.1203) or 70% (CQM) suffer from low QoE, about 90% of the VoIP clients achieve a high MOS score. We want to add here that the data sets are not equally distributed in terms of the true QoE. The low QoE scores for the VoD clients can be explained with their high bandwidth requirements, which often cannot be fulfilled due to the relatively high cell load during simulation. Although the ground-truth QoE distribution could easily be harmonized by removing samples of too frequent MOS values, we use the data set as it is. An MNO who actually deploys ML in its system will face similar issues. Additionally, it allows us to study the estimation accuracy in a more natural and realistic setting.

B. Relationship Between Most Expressive Feature and QoE

Next, we study the relationship between the QoE for VoD and VoIP clients and their respective features with the highest correlation according to Pearson's correlation coefficient. For VoD, independent of the applied model, the feature having the highest correlation with QoE is the average downlink throughput. We first detail on this feature's impact on the VoD QoE when computed according to the P.1203 model. Figure 3b denotes the average UE downlink throughput on the x-axis and the QoE scores on the y-axis. Different colors represent the QoE ranges, i.e., *low*, *medium*, and *high*. The white line shows the average values of the UE downlink throughput obtained within discrete MOS intervals with steps of 0.1. For example, it denotes the average UE downlink throughput of all clients achieving a MOS between 3.5 and 3.6. Additionally, the dotted lines denote the average value +/-



(a) Distribution of true QoE scores (b) VoD-P.1203 (c) VoD-CQM (d) VoIP

Fig. 3: Ground-truth QoE for the different services and its relation with the respective most expressive network-related feature.

the standard deviation. We observe a linear behavior between the average throughput and QoE. As expected, the higher the average throughput, the higher the MOS score. The plot further shows that with increasing MOS scores, the standard deviations increase and there are more throughput values that lie far away from the average of the 0.1-sized QoE interval. Figure 3c accordingly shows this relationship for the VoD QoE as computed by CQM. As expected, the relationship between the average downlink throughput and QoE is similar as with VoD-P.1203.

For VoIP, the feature with the highest correlation to QoE is the standard deviation of the downlink RLC delay. The impact of this feature's values on the MOS scores is denoted in Figure 3d. The higher the RLC downlink delay, the lower the MOS. Contrary to the VoD clients, the standard deviation of this feature's value increases with decreasing MOS values. Furthermore, we can see several outliers, especially in the region of medium and low QoE.

C. Feature Correlation Analysis

Figure 4 denotes the 40 features having the highest correlation with QoE for VoIP and VoD. For the VoD QoE according to P.1203, the highest correlation observed is the average UE downlink throughput (0.90), followed by the average UE uplink throughput with a value of 0.86. The subsequent 16 features are all related either to the UE uplink or downlink throughput, showing the high relevance of this monitoring statistic for the QoE estimation. The first feature which is related to a different monitoring metric than UE throughput is the 25th percentile of the UE uplink CQI, with a correlation coefficient of 0.38, followed by the minimum and average uplink CQI (0.37). The correlations when using CQM differ only slightly. Same as with P.1203, the two highest correlations are obtained for the average UE throughput on downlink (0.90) and uplink (0.86). Again, the first feature not related to UE throughput is the 25th percentile of UL CQI, which has a slightly higher correlation of 0.41 when using CQM instead of P.1203. The most expressive feature in case of VoIP is the standard deviation of the RLC downlink delay with a correlation of -0.86. In general, features generated from the RLC delay are highly correlated with the VoIP QoE. The first feature which is not related to any delay metric is ranked on the 20th place. It is the maximum UE downlink throughput which has a comparably low influence on the QoE with a correlation coefficient of only -0.17.

VI. Performance Evaluation of the Different Regression Techniques

This section evaluates and compares the different regression techniques applied in this work. We first perform a quantitative assessment and study the QoE estimation accuracy. Next, we investigate the different mechanisms with respect to their resource requirements and the duration for training and testing the models. Finally, we compare the different mechanisms in a qualitative manner.

A. Quantitative Assessment

For assessing the different mechanisms quantitatively, we first take a detailed look on the estimated versus true QoE. In a next step, we investigate how the estimation accuracy differs for the different data sets, i.e., depending on a client's movement characteristic or true QoE. The quantitative assessment concludes with an investigation of meta KPIs, such as the duration or CPU load for running the mechanisms.

1) Estimated vs. True QoE: This section studies the deviation of the estimated MOS score from the true score. Thereby, we also evaluate whether a mechanism tends to over- or under-estimate the QoE. We assume an estimation to be accurate, if it deviates less than 0.1 from the true score. For instance, if the true QoE is 2.7, any estimation between 2.6 and 2.8 is seen as accurate. Values below 2.6 represent an under-estimation and above 2.8 an over-estimation, respectively.

Figure 5 illustrates the true QoE and its estimation from the five regression techniques for VoD when using the ITU-T P.1203 model. The angle bisector represents the cases where the estimation equals the true QoE, i.e., the optimal case. Values above this line are under-estimations (shown in blue), values below this line are over-estimations (shown in red). All of the tested mechanisms have a similar fraction of under-estimates. The lowest rate is obtained for SVR (14.5%) and the highest one for LASSO, with about 17%. More significant differences can be observed when it comes to the QoE over-estimates. Thereby, KRR outperforms the other mechanisms with a fraction of about 14% over-estimation. NN performs

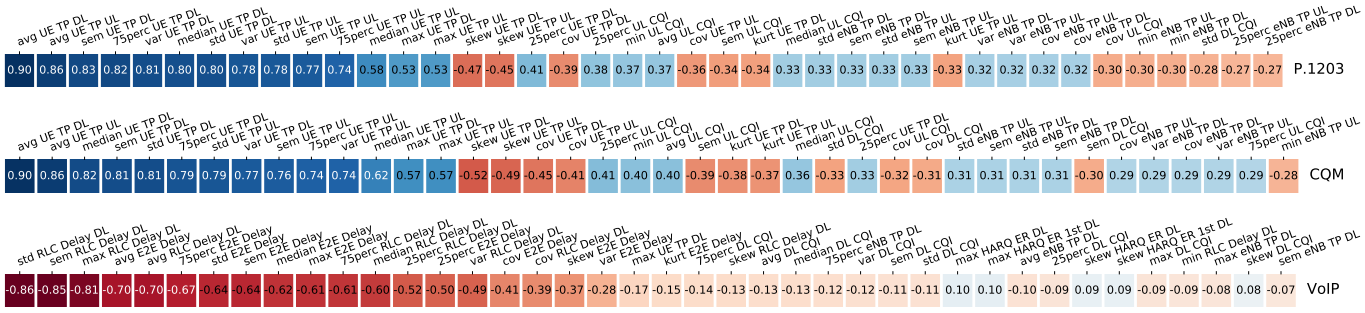


Fig. 4: Correlation between network-related features and QoE for the different service types. Only the 40 features with the highest Pearson correlation are shown. Values are sorted from left to right according their absolute correlation values.

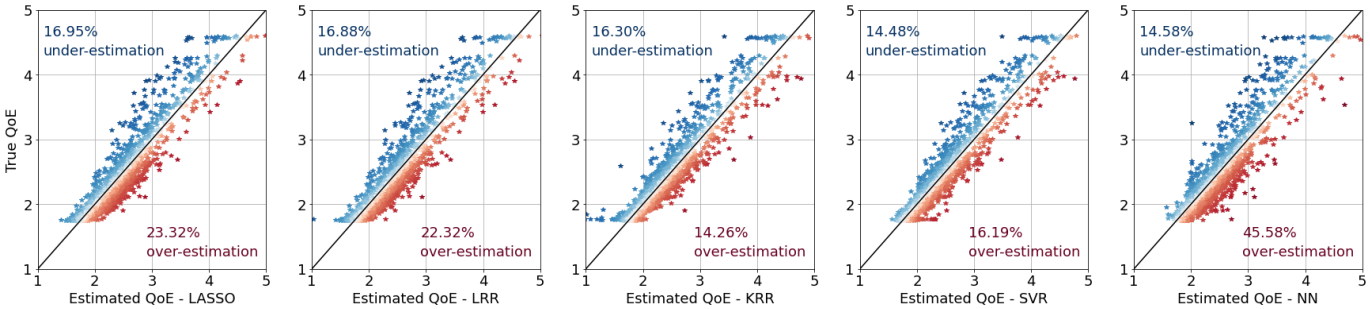


Fig. 5: VoD - ITU P.1203. Deviations of estimated QoE larger than +0.1 are counted as over-estimation, deviations larger than -0.1 are counted as under-estimation. Estimated values within the +/- 0.1 boundary are assumed as accurate estimations.

the worst and over-estimates the QoE in nearly half of the cases.

When using CQM instead of P.1203, the accuracy decreases for all of the five regression techniques, as shown in Figure 6. Similar as with P.1203, SVR and KRR yield the lowest fraction of under-estimations. However, the fractions are increased by roughly 7% compared to P.1203, resulting in an under-estimation rate of about 22% for SVR and 23% for KRR. The over-estimation rate increases as well when using CQM instead of P.1203. The only exception is NN, which over-estimates less for CQM than for P.1203. Another observation that can be drawn compared to P.1203 is the increased magnitude of deviations from the true QoE score. The data points lie farther away from the angle bisector, indicating that inaccurate estimates are of higher magnitude with CQM.

For VoIP, the QoE estimates tend to be more accurate, as shown in Figure 7. With LASSO, LRR, KRR, and SVR, the fractions of both, over- and under-estimation, are below 10%. SVR outperforms the other mechanisms and can achieve an accurate estimation within the +/-0.1-boundary for about 86.7% of the test samples. NN has a clear tendency towards under-estimating the QoE (29%), but over-estimates with a similar rate as the other four regression techniques.

2) Estimation Accuracy (RMSE): Next, we compare the five regression techniques using the root mean square error (RMSE). Thereby, we do not only consider the used application type, but also evaluate the estimation accuracy for different subsets of our ground-truth data set. For instance, when reporting the performance metrics, we take into account whether a UE was *stationary* or *moving*, and whether the ground-truth QoE is *low*, *medium*, or *high*. Please note that the models have not been explicitly trained on the subsets. Instead, they have been trained and optimized on the data set *all* and we only evaluate their performance within these subsets. The RMSE scores are illustrated in Figure 8. For the VoD-P.1203 (Figure 8a), there are only slight differences between the five mechanisms in terms of RMSE in the complete data set (*all*). LASSO and LRR both obtain an RMSE of 0.18 and KRR yields 0.19. SVR is most accurate with an RMSE of 0.16, the NN has the highest error with 0.22. If the RMSE is considered separately for stationary and for moving clients, it shows that for VoD - P.1203, in any case, the estimation is more accurate for moving clients. This is most significant for SVR and NN, which yield an RMSE of 0.26 and 0.29 for stationary clients, but can achieve 0.16 and 0.19 for moving ones.

Next, we consider the RMSE separately depending on the true ground-truth QoE. For low QoE scores, all techniques achieve a high estimation accuracy. SVR scores best with an RMSE of 0.08. The highest errors are obtained for NN and KRR with an RMSE of 0.15 each. For medium ground-truth QoE scores, the estimation errors increase for all five regression techniques. As before, SVR outperforms the other approaches and achieves an RMSE of 0.19. For high ground-truth QoE scores, we see again a decrease of the estimation accuracy. Compared to the low QoE subset, the RMSE scores are about the fourth-fold.

For the VoD-CQM (Figure 8b), the estimation error in the overall data set is higher compared to VoD-P.1203. SVR achieves the lowest RMSE with 0.23 (0.16 for P.1203) and

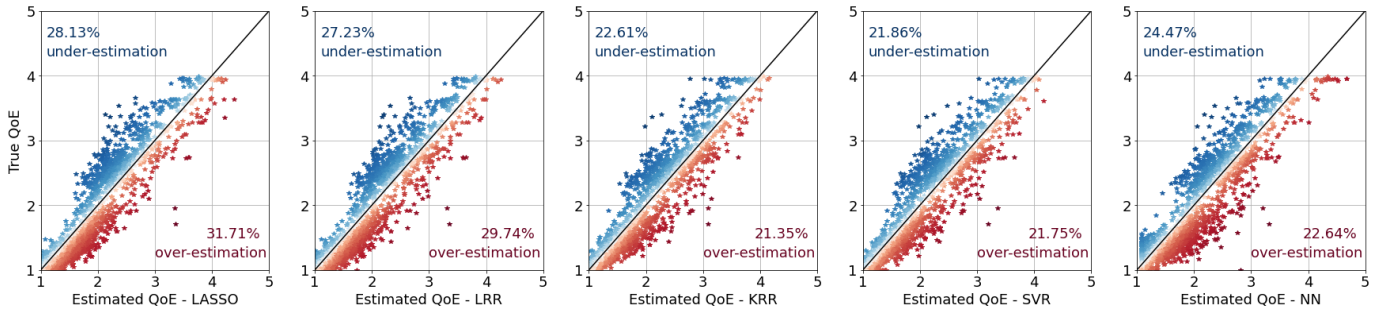


Fig. 6: VoD - CQM. Deviations of estimated QoE larger than +0.1 are counted as over-estimation, deviations larger than -0.1 are counted as under-estimation. Estimated values within the +/- 0.1 boundary are assumed as accurate estimations.

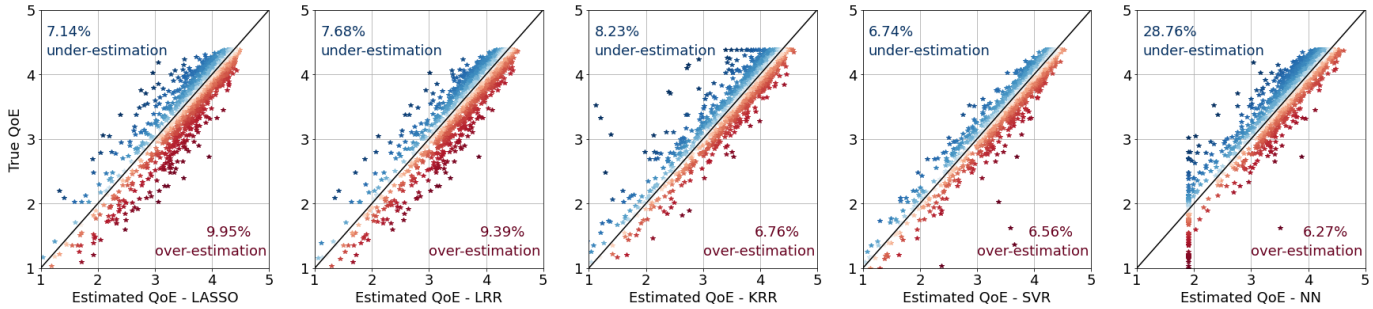
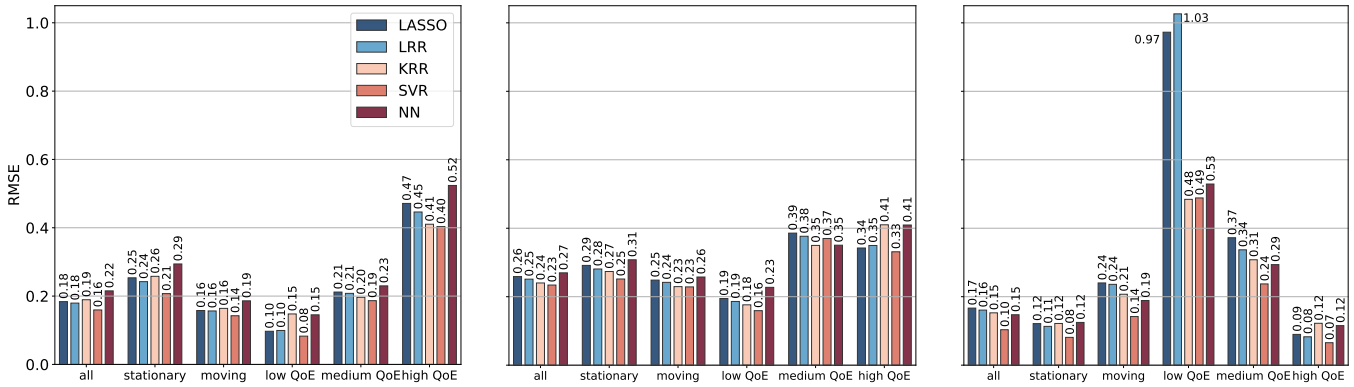


Fig. 7: VoIP. Deviations of estimated QoE larger than +0.1 are counted as over-estimation, deviations larger than -0.1 are counted as under-estimation. Estimated values within the +/- 0.1 boundary are assumed as accurate estimations.

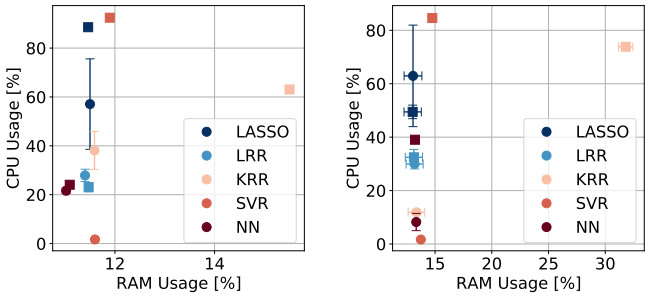


(a) VoD-P.1203 (b) VoD-CQM (c) VoIP

Fig. 8: RMSE scores obtained with the different regression techniques for the considered service types in the different subsets.

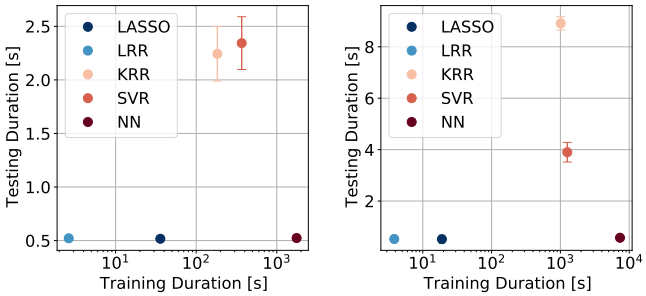
NN has the highest RMSE with 0.27. Again, the accuracy for moving clients is slightly better. However, the difference is less significant as with the P.1203 samples. Furthermore, low QoE scores can be estimated with higher accuracy than medium or high ones. The highest RMSE in the *low QoE* subset is 0.23 obtained from NN. This is still lower than the lowest RMSE in the *medium QoE* subset, which is 0.35 for KRR.

Finally, we investigate the estimation accuracy for VoIP, as shown in Figure 8c. Considering the whole data set (*all*), SVR achieves the lowest RMSE with 0.10 and LASSO the highest with 0.17. In general, the estimation is more accurate for VoIP, compared to VoD. Contrary to VoD, the QoE of stationary clients can be estimated with higher accuracy than the QoE of moving clients. Furthermore, Figure 8c shows that the VoIP QoE estimation is more accurate for higher ground-truth QoE values. While KRR achieves the best estimation accuracy in the *low QoE* subset with an RMSE of 0.48, it performs worst compared to the other mechanisms in the *high QoE* subset, but is still very accurate with an RMSE of only 0.12. This can be explained by the lower amount of ground-truth samples in the *low QoE* subset. Additionally, we showed in Figure 3d that with lower QoE scores, the values of the feature *std RLC Delay DL* are more spread along the x-axis, i.e., the relationship between the feature and QoE becomes less distinct and consequently makes the estimation of those samples more difficult. Finally, we note the low estimation



(a) VoD (b) VoIP

Fig. 9: Resource consumption of the five regression techniques for training (squares) and testing (circles). Errorbars denote the standard deviation obtained after five repetitions.



(a) VoD (b) VoIP

Fig. 10: Time needed to train the five regression techniques and duration for estimating the QoE with the trained model. Errorbars denote the standard deviation after five repetitions.

accuracy of LASSO and LRR in the *low QoE* subset. While the other three mechanism yield an RMSE around 0.5, it is 0.97 for LASSO and 1.03 for LRR. A possible explanation for the low performance of these two linear models is their limited capability to accurately capture the non-linear relationship between the delay, its variation (jitter), and the QoE. Although the non-linear models also outperform the linear ones in the data set *all*, the effect is more obvious when it comes to the low QoE scores, where the delay and its variation actually play a role and degrade the QoE.

In addition to the RMSE, we evaluated the estimation performance according to Pearson's correlation coefficient, Spearman's rank correlation coefficient, MSE, and median absolute error. Similar as for the RMSE, with respect to these metrics, the two kernel-based methods KRR and SVR outperform the other approaches in most of the cases.

3) Meta-KPI Analysis: Our meta-analysis for the performance of the regression techniques includes several metrics expressing the computational overhead of the ML models. Figure 9 denotes the CPU and RAM usage during training and testing, i.e., estimating the QoE for the samples in the test set. The metrics obtained for VoD are shown in Figure 9a. LRR and NN are CPU- and RAM-efficient during training. KRR has a higher RAM usage compared to the other mechanisms, while SVR and LASSO have the highest CPU-load during training. During testing, the most CPU-efficient approach is SVR and the most RAM-efficient approach is NN.

Figure 9b shows the resource consumption for VoIP. Again, KRR has the highest RAM usage during training, however, in another order of magnitude. While with VoD, it used 15.5% of the RAM, it increases to 31.8% with VoIP. The RAM usage is similar for the remaining approaches and they mainly differ in terms of their CPU usage. The highest CPU usage during training is observed for SVR, the lowest one for LRR. During testing, similar as with VoD, SVR is the most CPU-efficient approach and LASSO the least efficient one.

Finally, we denote the duration for training and testing in Figure 10. For VoD (Figure 10a), LRR, LASSO, and NN are capable of estimating the QoE of the 2790 test samples in about 0.52 seconds. KRR and SVR, the methods applying the kernel trick, are less efficient and need about 2.3 seconds. Figure 10b shows the respective results for VoIP. We point out that the VoIP testset contains 5518 test samples, roughly the double compared to the VoD testset. Again, LRR, LASSO, and NN are the fastest estimators and need about half a second. It takes 3.9 seconds for SVR, and KRR has the longest estimation duration with about 8 seconds.

These results are consistent with the theoretical complexities denoted in Table IV. During training, the kernel methods, i.e., KRR and SVR, are slower than the linear methods LASSO and LRR because $dn^2 > d^2n$, where $d = 84$ (number of features), and $n = 11160$ (training samples) for VoD and $d = 72$, and $n = 27592$ for VoIP. Training the NN with backpropagation takes more time. For testing, similarly, the kernel methods are slower than the others including NN. This is because their complexity is proportional to the number of training samples n, while the others do not depend on n.

B. Qualitative Assessment

Besides the typical performance metrics, stakeholders such as MNOs also need to take qualitative aspects into account when deciding which algorithm to deploy in their networks. In the following, we use qualitative scales and classify the algorithms applied in this work with respect to different design decisions, as shown in Table IV.

1) Complexity and Required Data Set Size: Primarily, the required size of the data set, which allows to train the model adequately, depends on the complexity of the problem. Nevertheless, the required amount of data also depends on the used model itself. In general, more complex models (i.e., with a high dimensional weight parameter space) require more data, unless unreasonably strong regularization is applied. LASSO and LRR both have a computational complexity of $O(d^3 + d^2n)$ during training, where d and n are the number of features and the number of training samples. For testing, their complexities are $O(d')$ and $O(d)$ respectively for each test sample, where $d'(\ll d)$ is the number of non-zero weights after training. Both linear models have a relatively low dimensional parameter space (usually the number of features plus one), and hence their requirements in terms of the number of samples in the training data set are comparably low.

TABLE IV: Classification of the evaluated mechanisms with respect to different qualitative aspects.

Topic	*Description*	LASSO	LRR	KRR	SVR	NN
Complexity	Computational complexity for training	$O(d^3 + d^2 n)$	$O(d^3 + d^2 n)$	$O(n^3 + dn^2)$	$O(n^3 + dn^2)$	linear to n
	Computational complexity for testing	$O(d')$	$O(d)$	$O(dn)$	$O(dn)$	linear to d
	Comprehensibility by humans	Easy	Easy	Medium	Medium	Hard
	Number of settable hyper-parameters	1 (λ)	1 (λ)	2 (λ and kernel) + kernel-specific settings	3 (C, ϵ, kernel) + kernel-specific settings	~ 10
	Number of model parameters to tune	max #features+1	#features+1	#samples	#samples	#NN connections
	Complexity to finding the (near) optimum	Easy	Easy	Medium	Medium	Hard
Data set	Requirements on the data set size	Low	Low	Medium	Medium	High
	Sensitivity towards outliers	High	High	Medium	Low	Medium
Feature selection	Detection of relevant features	Easy (Integrated)	Dedicated step needed (e.g. PCA, correlation analysis)			
	Reducing the number of used features	Easy (tuning α)	Hard/Dedicated step needed if not all features should be used			
Trackability	Feasibility to track the models evolution	Easy	Medium	Medium	Medium	Hard
	Amount of data that needs to be tracked	Low	Low	Low	Low	Medium
Over-fitting	Sensitivity towards over- or under-fitting	Low	Low	Medium	Medium	Medium
	Automatically applied prevention mechanisms	Yes	Yes	Yes	Yes	No
	Efforts to prevent over-fitting	Low	Low	Low	Low	Medium
Service coverage	Applicability to different types of problems	Low	Low	High	High	High

The two non-linear models KRR and SVR have complexities of $O(n^3 + dn^2)$ and $O(dn)$ for training and testing, respectively. Compared to the linear models, KRR and SVR with non-linear kernels typically require more data in order to learn the higher order dependencies. The complexity for training the NN is linear to n and for testing linear to d, but it highly depends on the architecture in both cases. Typically, the training time of the NN is longer than the kernel methods unless n is very large. Its benefits appear when a complex architecture is trained on a large data set. In this sense, it requires a large amount of data.

Outliers are critical in general when it comes to training ML algorithms and should be eliminated during the pre-processing step, if possible. The sensitivity towards outliers mainly depends on the loss function to be minimized — squared losses, e.g., MSE, are highly sensitive to outliers, while (piece-wise) linear loss is robust. Therefore, SVR trained with the piece-wise linear ϵ-insensitive loss tends to be more robust than the other models.

2) Training and Hyper-parameter Tuning: For the ML model for which the training is performed by solving a convex problem, reliable solvers are accessible — the problem is solved analytically or by an iterative algorithm with convergence guarantee. The training objectives for LASSO, LRR, KRR, and SVR are all convex, and therefore, we can expect that the model is stably trained when the model hyper-parameters are set appropriately. On the other hand, NN is trained by solving non-convex problems, and available state-of-the-art solvers are only guaranteed to converge to a local solution. There are many tips on choice of solvers, e.g., stochastic gradient descent or ADAM [28], initialization, and algorithm parameter setting, e.g., learning rate, momentum, the number of epochs, batch size, to likely get a "good" local solution, but good setting can depend on the model architecture and the model hyper-parameters, and therefore, cannot be fully automatic and human intervention is necessary when training fails. The model hyper-parameters have to be set appropriately. The linear methods, LASSO and LRR, have a single regularization parameter, which can be tuned by grid search on the validation error, i.e., prediction error on validation data. The kernel methods, KRR and SVR, have bandwidth parameters, which should be tuned, although the default value ($\gamma = 1$) can also work, assuming that the training data is appropriately pre-processed. For NN, the architecture corresponds to the model hyper-parameters, including the number of hidden layers, the neuron type, e.g., fully-connected, convolutional, pooling, etc., and the number of nodes in each layer. For extensive exploration, Bayesian optimization can be used.

3) Feature Selection and Interpretability: Finding the features that are relevant for estimating the QoE is important. Practically, this information can be used to reduce the costs for the estimation (e.g., collecting the data and processing it to generate the features), and allows to understand what the ML model has learned, or explain why it predicts a particular response for particular input features. The latter is important when ML models are deployed in real applications that require high reliability and security. For linear models, the learned weights can be seen as the importance of the corresponding features, and therefore are easily interpretable – the features for which the learned absolute weights are large are relevant for predicting the response. However, correlations between features can cause spuriously detected relevant features, because the contribution from the two positively correlated features with large positive and negative weights, respectively, can cancel each other out. LASSO was proposed to avoid this phenomenon. The sparsity inducing L1 regularizer suppresses the contribution (i.e., weight) from unnecessary features, and its solution for a fixed number of non-zero weights are guaranteed to correspond to the set of features that best predicts the response. The kernel methods and NN are seen as black-box predictors, and there is no straightforward way to interpret their decisions. In a recently emerging research field, called explainable artificial intelligence (XAI) [29], researchers are tackling to address this issue, and many methods have been and are being developed. However, no existing method is guaranteed to correctly explain the ML model, and furthermore,

vulnerability against adversarial attack was pointed out [30]. Accordingly, interpreting non-linear ML models is so far a relatively hard task, requiring at least some human effort and expert knowledge.

4) Trackability: If an ML model is re-trained regularly using new ground-truth data, it will evolve over time. This happens for diverse reasons. The content provider could change the model used to compute the QoE which is communicated via the AF or it could adjust application settings. For the example of VoD, this could be a change in the quality switching thresholds, a reconfiguration of the maximum amount of buffered playtime, or changing video encoding characteristics, e.g., the segment duration or video bitrate. In case of VoIP, such a change could be the implementation of a new voice codec. Furthermore, changes regarding the network configuration, such as applying another scheduling algorithm for resource allocation at the AN, might influence the correlation between network-related features and QoE. It might be of interest for an MNO, to monitor how the model evolves over time. For instance, to track which features gained importance and which ones became less relevant. Accordingly, the trackability of a model mainly depends on three factors, which have previously been discussed: Its capability for feature relevance analysis, its comprehensibility, and its number of (hyper-)parameters. Tracking how the model evolves over time is very simple with LASSO. It returns a p-value for each feature, which can be seen as a measure of its respective importance to estimate QoE. Besides, λ is the only configurable parameter that needs to tracked. With LRR, KRR, and SVR, such a simple tracking of feature importance cannot be performed. Indeed, the weights of the input features could be seen as a rough approximation of their importance, but this requires a linear kernel and that all features have the same scale, which is seldom the case. Hence, if the feature importance should be tracked, dedicated methods need to be applied, also beyond the training process. This allows to keep track of the feature importance in general, but not the feature importance with respect to the specific model which was applied. However, only few model parameter settings need to be monitored and we classify the trackability of KRR, LRR, and SVR as medium. To obtain the feature importance with NN, more complex methods, like permutation importance, need to be applied. The values of the features are, one after another, randomly shuffled. These shuffled values are used as an input for the trained model. Analyzing how much the prediction output is distorted by modifying the input, allows to estimate the importance of a feature. As this process is very inefficient, and a huge range of model parameters need to be tracked, we classify the NN as hard in terms of trackability.

5) Service Coverage: The range of problems to which a specific ML algorithm can be applied is varying. In the context of estimating QoE, we refer to this range as the service coverage, i.e., to how many service types an algorithm can be applied without knowing the relationship between input features and QoE upfront. Due to their linearity, LASSO and LRR are limited to services where this relationship is linear. Contrary, KRR, SVR, and NN can be used for most problems, even if the relationship between input features and response is not linear. Thereby, the NN with its deep architecture might show advantages in solving highly complicated problems.

6) Scalability of ML Methods: As discussed in Section VI-A3, the kernel methods, KRR and SVR, are slow both in training and testing when the number n of training samples is large: the training and the testing (for a single test sample) complexities are $O(n^3)$ and $O(n)$, respectively. Therefore, the linear models, LASSO and KRR, as well as NN are preferable for large n, if the performance is comparable. Note that kernel methods can be significantly sped up at the expense of approximation errors: For example, the Nyström method [31] that approximates the kernel matrix with a rank r matrix reduces the complexities to $O(r^2n)$ and $O(r)$ for training and testing, respectively, and the random Fourier feature method [32] that projects the input data into a $(d <)d'(\ll n)$ dimensional space so that the inner product approximates a radial basis function (RBF) kernel reduces the complexities to $O(d'^2n)$ and $O(d')$, respectively.

For dealing with big data with millions of training samples, even the models with linear dependence, LASSO, KRR, and NN, can be prohibitive to train. In such cases, the stochastic gradient descent, a standard algorithm for NN training where the gradient descent is performed with a small batch of samples at each iteration, should be used for training.

VII. Discussion

The following section briefly summarizes the key lessons learned from our feasibility study and afterwards discusses the limitations of our study.

A. Lessons Learned

In the scope of our feasibility study, we showed that by means of regression techniques and by utilizing network statistics that are available to an MNO in 5G systems, the QoE can reliably be estimated. For both applications, VoD and VoIP, and for both QoE models used to assess the VoD QoE, i.e., P.1203 and CQM, and for any of the investigated subsets (moving vs. stationary clients and low, medium, or high true QoE), there are at least two algorithms achieving an RMSE below 0.5. This shows that despite the imbalance in the data set, independent of the user's movement pattern and the model used to assess the QoE, we can estimate the performance with a deviation of less than 0.5 on MOS scale. The lowest estimation accuracy in the data set *all* is an RMSE of 0.27, observed for the QoE of VoD as obtained using the cumulative quality model (CQM) and the estimation carried out by the NN. This can still be seen as a very accurate estimation. Besides showing the general feasibility of an ML-based QoE monitoring approach for 5G, we identified that the linear models, i.e., LASSO and LRR, fail to estimate the *low QoE* scores for VoIP.

Apart from the achievable estimation accuracy, an MNO's choice in terms of the algorithm to deploy within the NWDAF is influenced by its operational requirements. Our analysis show - despite using the same framework and physical machine - the heterogeneity of the different models in terms of their CPU-load and RAM-usage, as well as of their duration

for training and testing. We learned that KRR and SVR, which yield the highest accuracy within our study, have the highest resource demands during model training and they need much longer for estimating the QoE than any of the other tested techniques. While the resource demands during the training process, which only needs to be performed initially and then only from time to time to keep the model updated, can still be acceptable, the long testing duration will make KRR and SVR inapplicable if real-time network control actions should be triggered by the estimate.

In addition, there are several qualitative considerations which affect the applicability of a specific regression technique. To take them into account within the scope of our study, we discussed the set of five regression techniques with respect to different factors, such as their complexity or comprehensibility. For example, we found that the decisions of the best performers, i.e., KRR and LRR, are hard to trace due to their complexity, and hence, could impede a root cause analysis for the MNO.

To summarize, by exploiting three of the new key features introduced with 5G, i.e., (i) interactions with externals such as third party APs, (ii) enhanced monitoring capabilities, and (iii) performing complex computations, current QoE monitoring limitations can be overcome. More specifically, in 5G systems, MNOs are capable of obtaining an accurate QoE estimation solely based on network KPIs, once an ML model is sufficiently trained. Besides elaborating on the challenges and design criteria as faced by an MNO, we identified the relevance of different features. This can be useful for future generations of mobile networks, e.g., by guiding towards a standardized set of statistics provided at the NWDAF, so to allow the deployment of ML-based QoE estimation also across multi-vendor networks. Furthermore, by means of the exemplary set of five regression techniques, we highlighted the need of taking the scaling of the NWDAF with respect to its computational resources into account.

B. Limitations

We now summarize the limitations of the conducted study and shortly discuss their impact on the drawn conclusions.

Simulated 4G traces vs. real 5G traces: Our evaluations are based on simulated 4G data. Compared to 4G, 5G networks have higher bandwidth capacity, lower delays, and at least for the throughput, 5G shows higher variances [33]. This, however, does not affect the relationship between QoS and QoE for specific applications [34]. Consequently, the estimation accuracy of the different models itself will not be affected, but we can expect a shift towards higher QoE scores in the collected ground-truth data set.

Limited set of considered applications: With VoD and VoIP, we investigated two types of applications following different fundamental QoS/QoE relationships [35]. Our study shows that the investigated regression techniques are capable to obtain a good QoE estimation, independent of the underlying relationship. Configurational or implementation-specific settings like the used video codec or the applied adaptive bitrate algorithm have only minor influence on those fundamental QoS/QoE relationships. This indicates that the presented approach can be generalized beyond its application within the conducted study.

Unbalanced data sets: Due to the investigated network scenarios, specific QoE ranges occur less often, making it harder for the regression techniques to train an accurate model. We observed significant differences in the estimation accuracy for different QoE ranges and it is hard to assess in how far these differences are due to the unbalanced data set. However, QoE literature shows distinct relationships between QoS metrics and any QoE score, suggesting that an accurate estimation based on QoS metrics is achievable for the whole range of QoE scores [34]. In this respect, we want to emphasize the practical orientation of our study and that an MNO would face similar issues, e.g., of receiving good VoIP QoE samples with higher frequency than low VoIP QoE samples. It can overcome this issue by increasing the number of overall collected samples, thus obtaining more of the rarely occurring QoE scores, which would allow to generate a more balanced data set.

Limited settings of hyper-parameters during training: Finally, we note that further optimizations of the QoE estimation are possible by tuning the different parameter settings. However, such an analysis is beyond the scope of this work, since we aim at highlighting the general feasibility of an automated training and testing cycle for QoE estimation in the 5G architecture. This is sufficiently shown by the high accuracy of the employed models, particularly by SVR and KRR, which highlights the applicability of the proposed system. In order to allow further improvements of the estimation accuracy, e.g., by using different ML techniques or testing additional parameter settings, we publicly provide our aggregated ground-truth data sets to the research community.

VIII. Conclusion

With the introduction of AF and NWDAF, the 5G networking architecture offers the necessary capabilities for data-driven QoE monitoring. Combined with automatically triggered network control mechanisms, the ML-based QoE estimation allows the deployment of self-driven QoE-aware networks, which do not require manual interference. The capability to estimate the QoE reliably in a multi-service domain with heterogeneous user-contexts is a crucial prerequisite to deploy such networks. In this work, we focused on the QoE estimation and showed that it can be efficiently and reliably implemented in 5G systems by using Machine Learning. Thereby, we first elaborated on the new 5G features, potentially eliminating the QoE monitoring limitations of earlier mobile network generations. Next, we investigated the estimation accuracy of five representative regression techniques and for two exemplary service types that differ with respect to their QoS-/QoE-relationship. All ML algorithms have been trained and tested on a large data set, generated via simulation activity. The conducted study shows that in general, a high estimation accuracy can be obtained with any of the investigated ML options and with reasonable operational overhead. However, in specific cases, the simple linear models fail to estimate the QoE accurately due to their inability to capture non-linear relationships. Despite the trade-off between a model's

complexity and its estimation performance, an MNO might consider additional factors when choosing an ML option to deploy. Hence, our quantitative evaluations are broadened by a qualitative comparison of the five regression techniques and this work can serve as a guideline and as a proof-of-concept for ML-based QoE estimation deployment in 5G networks

ACKNOWLEDGEMENT

This work is funded by the BMBF Software Campus Grant "BigQoE" (01IS17052). Shinichi Nakajima is supported by BMBF as BIFOLD - Berlin Institute for the Foundations of Learning and Data (01IS18025A/01IS18037A). Thomas Zinner is supported by the Norwegian Financial Mechanism 2014-2021 under project 2019/34/H/ST6/00599. The authors want to thank Huiran Liu und Marcin Bosk for their continuous support during the course of this work and Marija Gajic and Stanislav Lange for proofreading the article.

REFERENCES

[1] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi *et al.*, "Qualinet white paper on definitions of quality of experience," *European Network on Quality of Experience in Multimedia Systems and Services*, 2013.

[2] T. Hossfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE management for cloud applications," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 28–36, 2012.

[3] 3GPP, "Architecture enhancements for 5G System (5GS) to support network data analytics services," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.288, 12 2021, version 17.3.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579

[4] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[5] S. Schwarzmann, C. C. Marquezan, R. Trivisonno, S. Nakajima, and T. Zinner, "Accuracy vs. cost trade-off for machine learning based QoE estimation in 5G networks," in *IEEE International Conference on Communications: Next-Generation Networking and Internet Symposium*, Dublin, Ireland, Jun. 2020.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[7] H. Drucker, C. C. Burges, L. Kaufman, A. J. Smola, and V. N. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, 1997, pp. 155–161.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[9] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring video QoE from encrypted traffic," in *Proceedings of the Internet Measurement Conference (IMC)*. ACM, 2016, pp. 513–526.

[10] I. Oršolić, P. Rebernjak, M. Sužnjević, and L. Skorin-Kapov, "In-network QoE and KPI monitoring of mobile youtube traffic: Insights for encrypted ios flows," in *International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 233–239.

[11] Y.-T. Lin, E. M. R. Oliveira, S. B. Jemaa, and S. E. Elayoubi, "Machine learning for predicting QoE of video streaming in mobile networks," in *International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[12] D. Minovski, C. Åhlund, K. Mitra, and P. Johansson, "Analysis and estimation of video QoE in wireless cellular networks using machine learning," in *International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2019, pp. 1–6.

[13] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, "Intelligent network data analytics function in 5G cellular networks using machine learning," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 269–280, 2020.

[14] C. Hernández-Chulde and C. Cervelló-Pastor, "Intelligent optimization and machine learning for 5G network control and management," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2019, pp. 339–342.

[15] S. Schwarzmann, C. Cassales Marquezan, M. Bosk, H. Liu, R. Trivisonno, and T. Zinner, "Estimating video streaming QoE in the 5G architecture using machine learning," in *Workshop on QoE-based Analysis and Management of Data Communication Networks*, 2019, pp. 7–12.

[16] A. Varga, "OMNeT++," in *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[17] A. Virdis, G. Stea, and G. Nardini, "SimuLTE–a modular system-level simulator for lte/lte-a networks based on OMNeT++," in *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*. IEEE, 2014, pp. 59–70.

[18] S. Kosta, A. Mei, and J. Stefa, "Small world in motion (SWIM): Modeling communities in ad-hoc mobile networking," in *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2010, pp. 1–9.

[19] T. Stockhammer, "Dynamic adaptive streaming over HTTP – standards and design principles," in *Proceedings of the Multimedia Systems Conference (MMSys)*. ACM, 2011, pp. 133–144.

[20] ITU-T, "P.1203.1 : Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport - video quality estimation module," in *Tech. rep. International Telecommunication Union*, 2019.

[21] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M. Garcia *et al.*, "HTTP adaptive streaming QoE estimation with ITU-T Rec. P. 1203: open databases and software," in *Proceedings of the Multimedia Systems Conference (MMSys)*. ACM, 2018, pp. 466–471.

[22] A. Raake, M. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, "A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P. 1203.1," in *International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2017, pp. 1–6.

[23] H. T. Tran, N. P. Ngoc, T. Hoßfeld, and T. C. Thang, "A cumulative quality model for HTTP adaptive streaming," in *International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018, pp. 1–6.

[24] I. Rec, "ITU-T G.107 – the E-model: a computational model for use in transmission planning," *International Telecommunication Union*, vol. 8, no. 20, 2015.

[25] T. Michael, G. Mittag, and S. Möller, "Analyzing the fullband e-model and extending it for predicting bursty packet loss," in *International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[26] M. Soloducha and A. Raake, "Speech quality of voip: bursty packet loss revisited," in *Speech Communication; 11. ITG Symposium*. VDE, 2014, pp. 1–4.

[27] 3GPP, "Management and orchestration; 5G performance measurements," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.552, 12 2021, version 17.5.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3413

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

[30] A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *Advances in Neural Information Processing Systems*, 2019.

[31] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2001.

[32] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, 2008.

[33] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5G dataset with channel and context metrics," in *Proceedings of the Multimedia Systems Conference (MMSys)*. ACM, 2020, pp. 303–308.

[34] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.

[35] J. Shaikh, M. Fiedler, and D. Collange, "Quality of experience from user and network perspectives," *Annals of telecommunications-annales des telecommunications*, vol. 65, no. 1, pp. 47–57, 2010.

Susanna Schwarzmann received her Bachelor's and Master's degree in Computer Science from the University of Würzburg in 2014 and 2016, respectively. After that, she worked as a research assistant in the Internet Network Architectures group at TU Berlin, where she received her PhD in 2022. In 2021, she joined Huawei, where she holds the position of a Research Engineer at the Network Architecture group for the Advanced Wireless Technologies Lab (AWTL).

Clarissa Marquezan currently holds the position of Principle Engineer at Network Architecture group for the Advanced Wireless Technologies Lab (AWTL) at Huawei Technologies, Munich Research Center. She works on developing research and solutions for standardization bodies in the areas of Network Architecture and Core Network. She joined Huawei Technologies in 2014. She holds a PhD and MSc in Computer Science from Federal University of Rio Grande do Sul, Brazil (achieved in 2010 and 2006, respectively).

Riccardo Trivisonno (IEEE M'08, SM'19) is currently serving as Head of Network Architecture for the Advanced Wireless Technologies Lab (AWTL) at Huawei Technologies, Munich Research Center. The research department focuses on technologies development and transfer to standardization bodies, in the areas of B5G/6G Network Architecture, Core Network and Network Management. He joined Huawei Technologies in 2011 and he has been chairman of the 6G-IA Pre-standardization WG since 2020. He holds a PhD and an MSc in Telecommunications Engineering from University of Bologna (achieved in 2005 and 2000 respectively).

Shinichi Nakajima received the master's degree in physics from Kobe University, Kobe, Japan, in 1995, and the Ph.D. degree in computer science from the Tokyo Institute of Technology, Tokyo, in 2006. He worked with Nikon Corporation, Tokyo, until September 2014 on statistical analysis, image processing, and machine learning. He is currently the lead of the Independent Research Group Probabilistic Modeling and Inference, Berlin Institute for the Foundations of Learning and Data (BIFOLD), Technische Universität Berlin, Berlin, Germany. His research interest is in theory and applications of machine learning, in particular, Bayesian learning, variational inference, generative modeling, computer vision, explainable artificial intelligence (XAI), and machine learning applications in sciences.

Vincent Barriac is an R&D engineer with Orange Innovation (France). His main topic of expertise is the assessment of QoS for mobile communications. Involved since two decades in standardization activities, in particular in the development and the application of QoS prediction models, he holds a position of working party chairman at ITU-T SG12 (Performance, QoS and QoE). He is also engaged in the operational deployment and utilization of QoS monitoring solutions in European and African countries where the Orange Group is operating communications services and facing regulation concerning QoS for these services.

Thomas Zinner has been associate professor at the Department of Information Security and Communication Technology at NTNU, Norway since 2019 and currently leads the Networking Research Group. He was Visiting Professor and head of the research group INET at TU Berlin from 2018 - 2019. From 2013 - 2018 he was Head of the Research Group on Next Generation Networks at the Chair of Communication Networks, University of Würzburg. He received the Ph.D. degree in Computer Science from University of Würzburg in 2012. His research interests cover cognitive network management and network softwarization with particular focus on performance and security aspects. He is the recipient of several best paper awards, the DASH-IF "Excellence in DASH Award" (2020) and the ITC "Rising Scholar Award" (2019). Thomas is a Member of IEEE and ACM and has served as the Technical Program Chair for ITC 2018. He has been involved in the organization and technical program committees of many conferences and workshops, including ITC, Netsoft, IM/NOMS, CNMS, and ACM CoNEXT.