Accuracy vs. Cost Trade-off for Machine Learning Based QoE Estimation in 5G Networks

Susanna Schwarzmann*, Clarissa Cassales Marquezan ‡,
Riccardo Trivisonno ‡, Shinichi Nakajima*†¶, Thomas Zinner§

* Technische Universität Berlin, 10587 Berlin
‡ Huawei Technologies, Germany, 80992 München
† Berlin Big Data Center, 10587 Berlin
¶ RIKEN Center for AIP, 103-0027, Tokyo
§ NTNU, 7041 Trondheim

***Abstract*—Since their first release, 5G systems have been enhanced with Network Data Analytics Functionalities (NWDAF) as well as with the ability to interact with 3rd parties' Application Functions (AFs). Such capabilities enable a variety of potentials, unimaginable for earlier generation networks, notable examples being 5G built-in Machine Learning (ML) mechanisms for QoE estimation, subject of this paper. In this work, an ML-based mechanism for video streaming QoE estimation in 5G networks is presented and evaluated. The mechanism relies on an ML algorithm embedded in NWDAF, the collection of 5G network KPIs, and the collection of QoE information from video streaming service provider, i.e., the 3rd party AF. The mechanism has been evaluated in terms of QoE estimation accuracy against the cost in terms of required input sources and data for the estimation, and its performance has been compared to alternative methodologies not making use of ML. The evaluation, via simulation activity, clearly highlights the benefits of the proposed mechanism. Based on the derived results, the required input sources are ranked with respect to their importance.**

***Index Terms*—HAS, QoE, Machine Learning, 5G**

I. Introduction

Online video streaming has become a very popular application in today's networks and consequently contributes a large fraction of the global IP traffic. Cisco forecasts that mobile video streaming will make up nearly 80% of the overall mobile data traffic by 2022 [1]. Driven by business incentives, providing a good Quality of Experience (QoE) is of up-most importance for Mobile Network Operators (MNOs). However, while the MNO has capabilities to perform application-aware resource control, it is the content provider that is capable to obtain a reliable QoE estimation based on application data.

This discrepancy can be tackled by the 5G system architecture, which introduces extended capabilities for information exchange between MNOs and third party Application Functions (AFs). 5G Network Functions (NFs) and AFs have standardized interfaces that allow a content provider (CP) to communicate application-specific information, such as QoE, to an MNO. However, this information exchange increases the costly control plane traffic and the MNO is depending on the information provided by the CP, i.e., the MNO cannot influence the type, granularity, or frequency of the received data. To nevertheless obtain an estimation of the QoE in the system at any point in time, MNOs of 5G networks can apply different techniques to derive the QoE from network-level statistics, as the newly introduced Network Data Analytics Function (NWDAF) specifies.[1]

Still it is not clear which techniques are appropriate for deriving the QoE from network-level statistics. The applicability of a certain technique is determined by i) the accuracy that can be achieved and ii) the costs that are incurred. It is to be assumed that the more network statistics, i.e., features, are provided, the more accurate the QoE can be estimated. To the best of our knowledge, this trade-off between costs, e.g. expressed as the efforts for collecting network statistics, and the accuracy has not been studied so far.

This paper aims at filling this gap by analyzing this trade-off for QoE estimation based on a video streaming use-case. As an enhancement to our prior work [2], we rely on a mechanism which tunes both, the selected features and the QoE estimation model, for a limited number of features. This allows to evaluate the performance in terms of estimation accuracy versus costs in terms of required input data. In order to study and emphasize the possible benefits from ML, we compare the proposed mechanism with two simple alternative approaches for the sake of benchmarking. Furthermore, we reveal the mobility of a video streaming client as an influence factor on both, the estimation accuracy and the relevant features.

The remainder of this paper is structured as follows. Section II presents background and related work. Section III details on our methodology. Afterwards, Section IV presents the results and Section V concludes the paper.

II. Background and Related Work

Machine learning is widely applied in the context of classifying QoE influence factors, such as bitrate, resolution, or stallings, from encrypted network traffic. For example, the capabilities of different ML-based algorithms to classify such

[1]3GPP TS 23.288 V16.0.0, Architecture enhancements for 5G System (5GS) to suppor tnetworkdata analytics services, 2019-06

influence factors are studied in [3]–[8]. Instead of classifying specific QoE influence factors, the works presented in [9] and [4] aim at classifying an overall QoE score on MOS scale, using the standardized ITU-T P.1203 model [10]. QoE estimation focusing mobile networks is given in [11], which studies the performance of various classifiers and the impact of the used features retrieved from network- and application-related data. However, it is desirable to estimate the QoE solely using network-related features, as such data is typically available to an MNO. Such a solution is presented in [12], focusing on LTE networks.

Applying machine learning in 5G gains an ever-increasing interest, e.g. for predicting the number of active users [13], for traffic forecasting [14], or cognitive networking [15]. A first QoE-centric approach provides a service quality prediction model for UHD real time video streaming [16] and a network resource allocation system for autonomous QoE-aware 5G network management is discussed in [17].

In our previous work [2] we proposed a possible integration of an ML-based QoE estimation workflow into the 5G architecture, making use of the newly introduced network functions NWDAF and AF. We evaluated the performance of two basic regression methods for an increasing number of features, following a greedy approach. Based on this, we now use a more sophisticated method, which retrieves an optimal set of features for a given cost limit, i.e. the number of features. This allows to analyze the trade-off between the costs, e.g. expressed as the number of input features, and the estimation accuracy. We furthermore highlight the benefits of using more complex ML-based approaches over simple QoE estimation techniques, such as least-square fittings.

III. Methodology

In our previous work [2], we proposed a three step approach for estimating QoE in 5G networks. Firstly, ground-truth QoE is received via the AF and made available to the NWDAF. Next, the QoE values and NWDAF statistics are used to reveal relevant features, train models, and evaluate the performance of regression methods for different sets of input features. Finally, the QoE can be estimated using the trained model and NWDAF network statistics.

In this work, we detail on the second step and propose to use the more sophisticated method *least absolute shrinkage and selection operator (LASSO)* [18], which finds optimized feature sets and models given a limited number of features to be used. In the following, we describe LASSO and two benchmark approaches which require less data analytics. Afterwards, we introduce the simulative experiments, the collected data, and show how we quantify the costs in terms of monitoring efforts.

A. Feature Selection and QoE Estimation Using LASSO

Our goal is to estimate the QoE y from a limited number of features as accurate as possible under a limited total cost of making the necessary features available (in the run time). This can be expressed in a standard machine learning formulation, called the regularized linear regression. Let c_d be the computation/communication cost of the d-th feature x_d. Then, we want to solve the following problem to minimize the regression error with regard to weights:

$$\min_{\boldsymbol{w}} L \equiv (y - \textstyle\sum_{d=1}^{D} w_d x_d)^2 \qquad (1)$$
$$\text{subject to} \textstyle\sum_{d=1}^{D} c_d \delta(w_d \neq 0) \leq \lambda_0.$$

Here, $\boldsymbol{w} = (w_1, \ldots, w_D) \in \mathbb{R}^D$ is the regression parameter, $\lambda_0 > 0$ is the total cost budget, and $\delta(\cdot)$ is the indicator function giving one if the event is true and zero otherwise. Unfortunately, the problem (1) is known to be intractable (NP-hard) because of the non-convex constraint, for which a convex surrogate is often used:

$$\min_{\boldsymbol{w}} L = (y - \textstyle\sum_{d=1}^{D} w_d x_d)^2 \qquad (2)$$
$$\text{subject to} \textstyle\sum_{d=1}^{D} c_d |w_d| \leq \lambda_1,$$

where y and $\boldsymbol{x} = (x_1, \ldots, x_D)$ are assumed to be standardized, so that the means and the standard deviations of y and $\{x_d; d = 1, \ldots, D\}$ are all zero and unity, respectively. The problem (2) is now convex, and can be simplified as

$$\min_{\boldsymbol{w}} L \equiv (y - \textstyle\sum_{d=1}^{D} w_d x_d)^2 + \alpha \|\boldsymbol{w}\|_1, \qquad (3)$$

for the regularization parameter $\alpha > 0$ (which should be appropriately set so that it matches λ_1) and the pre-processed features such that the means and the standard deviations of $\{x_d\}$ are zero and c_d^{-1}, respectively. $\|\boldsymbol{w}\|_1 \equiv \sum_{d=1}^{D} |w_d|$ is called the ℓ_1-norm, which is often used in machine learning in order to induce sparsity—many of $\{w_d\}$ get exactly zero as α grows. It was proven that the solution of the problem (3) is piece-wise linear to α, and various efficient solvers have been developed and are available online.[2]

The number $\sum_{d=1}^{D} \delta(w_d \neq 0)$ of used features is monotonically decreasing with respect to α, providing the best feature set for each number of used features. When the total cost budget λ_0 is given, we evaluate the total cost $\sum_{d=1}^{D} c_d \delta(w_d \neq 0)$ for each of those best feature sets, and choose the one giving the lowest regression error under the budget.

In later parts of this work, we refer to the ML-based LASSO approach as *ML*. Please note that adapting α allows to tune the number of input features used for the estimation model, and as a consequence, the obtained estimation accuracy. We will detail on two special feature set sizes obtained with the LASSO. The first one is the feature set which yields the highest estimation accuracy among all feature sets, which we refer to as *ML*$_{opt}$. The second one is to use only one single feature, which we refer to as *ML*$_{1}$.

B. Benchmarking and Performance Metrics

Besides LASSO, we consider two further approaches for QoE estimation, which require less data analytics and come with a significantly lower complexity. They are used as a baseline against the ML approach. We denote the first option as

[2] We used python's sklearn implementation for our studies: scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

average (AVG). The QoE estimation is in any case the average QoE of the input training data, i.e. the average of a subset of the ground-truth QoE values. Hence, it always returns the same value for the estimated QoE and does not require any input features at all. The second option is the *least-square fit (LSF)*. It estimates the QoE based on a least-square fit of the *average UE downlink throughput* and the ground-truth QoE of the training set. The average UE downlink was chosen as it turned out to be the most powerful feature for estimating QoE. From our data, 70% are used for training the models or generating the fitting functions and 30% are used to apply the models and evaluate their QoE estimation accuracy. We perform a ten-fold cross-validation. The accuracy is evaluated using the metrics mean squared error (MSE), median absolute error (MedAE), and the coefficient of determination (R^2).

C. Simulation Environment and Experiments

To simulate the video clients and the network, we use *OMNeT++* [19] and the frameworks *INET* and *SimuLTE*.[3] We have a single mobile cell which is serving 80 clients to generate a decent load in the system. We run several experiments that differ in terms of the clients' movement characteristics. Basically, we consider two different types of clients. While the *stationary* clients do not move during their video sessions, the *moving* clients are constantly moving within the cell. Thereby, the clients either move along with the speed of a pedestrian or with vehicle speed, which are assumed to be 3 kmph and 50 kmph, respectively. Our scenarios cover several distributions of the different client types. We consider having only static clients, having only moving clients, and some mixed scenarios. In order to obtain a realistic behavior of the moving clients, we use the small world in motion (SWIM) mobility model [20], to determine the paths of the clients between their initial positions and one or more points of interest (POI). We consider four different POI settings: a single POI which is located at the edge of the cell, a single POI which is very close to the access node (AN), and 10 and 100 POIs, which are randomly placed within the mobile cell. Each configuration, i.e. setting of POI and client type distribution, is simulated using several seeds, whereby the seed determines the (initial) UE placement.

D. Collected Data and Network Features

We collect video streaming KPIs, such as video quality, initial delay, and number and duration of video interruptions as application-level data. From these KPIs, we compute the QoE score ranging from 1 to 5 for each video session using the ITU-T P.1203 model [10], [21]. We refer to this data as the *QoE ground-truth*. The network-related data monitored in the simulation is summarized in Table I as so-called *monitoring types*. In total, we have eight different monitoring types, which yield time-series on a per-second scale. Furthermore, the table shows the statistics applied on each of the time series in order to generate *features*. These features are later used together with the QoE ground-truth to train models, that allow a QoE estimation only based on network-related features. In total, we obtain 96 different features.

[3]The simulation environment used in this work is available on Github. github.com/fg-inet/vagrant-omnet-simulation-mobility

TABLE I: Monitored data and statistics for feature generation.

	Monitoring Type	Description
Monitored Data	AN_Tp_Dl	Access Node downlink (Dl) throughput
	AN_Tp_Ul	Access Node uplink (Ul) throughput
	UE_Tp_Dl	UE Dl throughput
	UE_Tp_Ul	UE Ul throughput
	CQI_Dl	Dl channel quality indicator (CQI)
	CQI_Ul	Ul CQI measured at UE
	M_RTT	Measured RTT at the UE
	S_RTT	Smoothed RTT using a moving average
Statistics	average, minimum, maximum, 25th percentile, 75th percentile, median, standard deviation (std), variance, coefficient of variation (cvar), kurtosis, skewness, unbiased standard error of the mean (sem)	

E. Quantifying the Costs for QoE Estimation

In order to compare the costs for the ML-based QoE estimation approach and the benchmark approaches, we rely on two simple quantities. The first one is the number of input features. The second one is the number of different monitoring types that are needed in order to generate the respective features. For example, if a model requires as input features *mean UE_Tp_Dl*, *std UE_Tp_Dl*, and *std AN_Tp_Dl*, the number of monitoring types reduces to two, as *mean UE_Tp_Dl* and *std UE_Tp_Dl* both can be retrieved by monitoring a UE's downlink throughput.

In the context of costs, we want to emphasize that certain features are more expensive than others. For example, monitoring the throughput of an AN is cheaper than monitoring the throughput per UE, given that more than one UE is active in a cell. Thereby, the margin increases with the number of active clients in the system. Furthermore, there are costs for processing the data to generate features, as well as costs for transmitting and storing the data. These cost factors are omitted in this work.

IV. Evaluation

In the following section, we compare the cost and accuracy for LASSO against the benchmark approaches and thereby consider the user mobility as a possible influence factor. Next, we analyze the cost-accuracy trade-off for the ML-based approach in more detail and present the relevant features for exemplary feature set sizes. The section concludes with a short discussion of the results.

A. Costs and Accuracy: LASSO vs. Benchmarks

Table II shows the results obtained when estimating the QoE either as the *average (AVG)* of the training data, based on the *least-square fit (LSF)* using the average UE downlink throughput, or applying machine learning using one feature (*ML_1*) or the optimized feature set (*ML_{opt}*). In terms of estimation accuracy, there is no significant difference between applying *AVG* or *LSF*. However, *LSF* comes with increased costs, as it needs to monitor the UE downlink throughput.

When relying on the average UE downlink throughput only, but using the more complex ML approach (ML_1) instead of the

TABLE II: Accuracy of different QoE estimation techniques. #MT denotes the number of monitoring points.

	Dataset	Performance			Cost	
		MSE	MedAE	R^2	#features	#MT
AVG	all	0,59933	0,50238	-0,00014	0	0
	stationary	1,03077	0,80394	-0,00083	0	0
	moving	0,46028	0,42191	-0,00079	0	0
LSF	all	0.59576	0.50855	0.00354	1	1
	stationary	1.07177	0.83727	0.00189	1	1
	moving	0.47096	0.43967	-0.00182	1	1
ML_1	all	0.57159	0.49499	0.04616	1	1
	stationary	0.59413	0.64491	0.42312	1	1
	moving	0.29409	0.34732	0.36056	1	1
ML_{opt}	all	0.14820	0.18207	0.75269	33	8
	stationary	0.18016	0.23412	0.82507	35	8
	moving	0.12682	0.17998	0.72425	36	8

basic least-square fit, the estimation accuracy can be increased for the moving and the stationary clients. For the stationary set, the MSE can be reduced from 1.07 to 0.59 and for the mobile set from 0.47 to 0.29. However, if we compare *LSF* and *ML₁* for the whole data set, i.e. *all*, the improvement by using machine learning is not significant.

All performance metrics can be significantly improved when using the optimized feature sets obtained by LASSO (ML_{opt}).[4] While we obtain an MSE of roughly 0.6 with *AVG*, *LSF*, and ML_1 in the complete dataset, i.e. *all*, this value can be reduced to less than 0.15 with ML_{opt}. However, the costs in terms of number of features increases to more than 30. Obtaining these features requires monitoring all of the eight monitoring types.

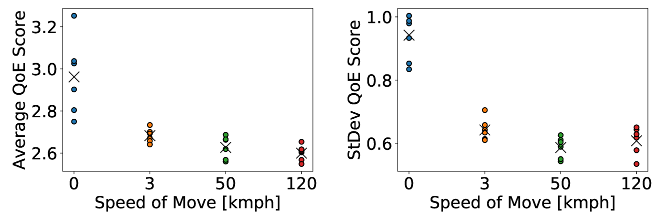
Another observation from the table is the increased estimation error for the *stationary* clients, compared to the *moving* ones. When using *LSF* or *AVG*, the QoE of the moving clients can be estimated with an MSE below 0.5. In case of the non-mobile, i.e. *stationary* clients, the MSE is above 1.0. This issue is investigated in more detail in the following paragraph.

B. Impact of Mobility on QoE Scores

As discussed above, for estimating the QoE of moving clients, we obtain significantly lower errors than for estimating the QoE of those clients who do not move. This indicates an easier QoE estimation for moving clients. To investigate this issue in more detail, we run additional experiments, whereby we consider 80 clients, all moving with a speed of either 0, 3, 50, or 120 kmph. To vary the (initial) UE placement, we use 6 different seeds.

Figure 1a shows a decreasing QoE when the clients move faster. While the non-moving UEs obtain an average QoE score of 2.96, the score decreases to 2.68 when clients move with a speed of 3 kmph and to 2.63 for a speed of 50 kmph. We can also observe that for the stationary clients, different seeds can result in significantly different QoE scores. The smallest average QoE for non-moving clients is 2.74, the highest average value is 3.25. For the moving clients, the seed has a much lower influence on the resulting average QoE.

[4]The features included in the optimized sets can be found here: github.com/fg-inet/vagrant-omnet-simulation-mobility



(a) Average QoE (b) QoE Standard Deviation

Fig. 1: Impact of speed of move on QoE. Dots denote results of specific (initial) UE placements, X denotes overall results.

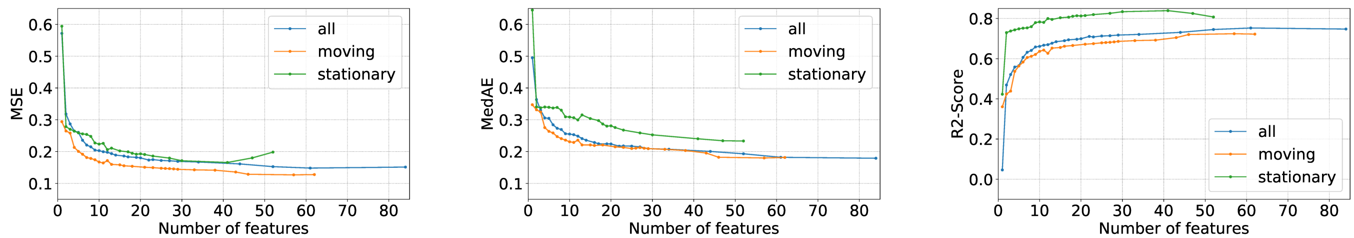
Figure 1b shows that the QoE values of stationary clients have a higher standard deviation compared to the moving ones. In the experiments with non-moving clients, the QoE standard deviation is 0.94. This value decreases to 0.64 for clients with a speed of 3 kmph.

From the above observations, we can conclude that moving clients have a lower average QoE and a lower standard deviation of QoE. This can be explained as follows. For the stationary clients, the position in the cell is the determining factor for the quality of their video stream. While clients that are placed nearby the AN obtain high QoE scores, the clients at the edge of the cell suffer of low video performance. This effect compensates for the clients that are moving between their initial positions and the points of interest. In Table II, we showed that with *AVG* and *LSF*, the QoE estimation error for the data set of moving clients is drastically lower than for the set of stationary clients. This can now be explained with the reduced variance in the data set of moving clients.

C. Cost vs. Accuracy Trade-off for LASSO

By tuning the α-parameter, LASSO allows to tune the number of features that are used for estimating the QoE. In general, a larger feature set results in a more accurate estimation. However, from an MNO perspective, the number of necessary features should be as low as possible. In the following, we analyze the costs in terms of the number of features needed and the estimation accuracy that can be achieved with a certain number of features. Figure 2 shows the estimation accuracy in terms of MSE and MedAE as well as the resulting R^2 score that can be achieved for different feature set sizes. In case only a single feature is used, the values correspond to those shown in Table II for ML_1. In all cases, when α was tuned so to obtain a single feature, the chosen feature was the average UE downlink throughput. The estimation errors for both, MSE (Figure 2a) and MedAE (Figure 2b), is higher for the data set of static clients, compared to the moving ones. We observe a contrary behavior for the R^2 score (Figure 2c), which is in any case better for the *stationary* clients than for the *moving* clients. This is due to the fact that the variance of QoE is larger for stationary clients than for moving clients, and implies that introducing an ML model for QoE estimation is more effective for stationary clients, while a more accurate prediction is achieved for moving clients.

Figure 3 illustrates the number of monitoring types needed to generate the features leading to a certain accuracy, as shown



(a) MSE (b) MedAE (c) R^2

Fig. 2: Influence of feature set size on the QoE estimation performance for the different datasets.

in Figure 2. The total number of available monitoring types is eight (see Table I). Please note that a single monitoring type can imply more than one feature. As described above, whenever the QoE was estimated using only a single feature, it was based on the average UE downlink throughput. Accordingly, having only one monitoring type implies that this is the UE downlink throughput. In this case, the MSE values range between 0.59 (stationary) and 0.26 (moving). As soon as the CQI on the uplink is added as a second monitoring type, the MSE can be reduced by roughly 0.32 for the stationary clients' dataset. By considering as well the downlink CQI, we can achieve an MSE that is below 0.22 for any of the datasets. The order of added monitoring types is the same up to the fourth added type, which is the UE uplink throughput. For the dataset *all*, the fifth monitoring type added is the AN uplink throughput, followed by smoothed and measured RTT. In case of stationary clients, the fifth added type is the measured RTT. For the moving clients, the smoothed RTT comes fifth, followed by the AN uplink throughput and measured RTT. For all input data sets, the least relevant monitoring type is the AN downlink throughput, which can be interpreted as the sum of all UE's uplink throughput.

D. Feature Relevance

Table III shows the features which were found to be relevant for exemplary feature set sizes[5] of 5 and 10. Please note that a smaller feature set is not necessarily the subset of a larger feature set when using LASSO. The majority of features are generated from the UE downlink throughput (UE_TP DL) and CQI, which once more highlights the relevance of these monitoring types.

Another observation that can be drawn from the table is that in the dataset of *moving* clients, more features expressing variability can be found. While features related to co-variance or standard deviation neither occur in the dataset *all*, nor in the *stationary* one, they are present several times in the dataset of *moving* clients. On the other hand, the average occurs in the dataset of *all* clients and in the dataset of the *stationary* clients, but never in the set of *moving* clients. The fact of more variability-related metrics in the set of moving clients could indicate that MNOs need to monitor with higher granularity

[5]Supplementary data, giving the relevant features for any feature set size, is available here: github.com/fg-inet/vagrant-omnet-simulation-mobility

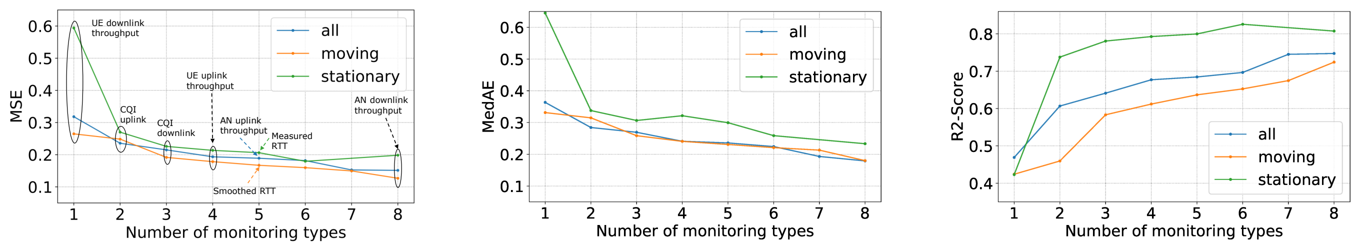
TABLE III: Features used for QoE estimation when using feature sets of size 5 and 10. Exponents denote the inclusion of a feature in the set with the respective size.

		Dataset		
Mon. Type		**all**	**stationary**	**moving**
UE_TP	DL	25th perc.^{10} median^{10} average^{5} 75th perc.^{5,10} sem^{5,10}	25th perc.^{10} median^{5,10} average^{5,10} 75th perc.^{10}	25th perc.^{5,10} median^{5,10} sem^{5,10} max^{10}
	UL	max^{10}		
CQI	DL	75th perc.^{10} average^{10}	75th perc.^{5,10} average^{5,10} max^{10} median^{10} min^{10}	std^{5,10} co-variance^{10} sem^{10}
	UL	average^{5,10} 25th perc.^{5,10} min^{10}	average^{5,10}	co-variance^{5,10}
S_RTT				max^{10}
AN_TP_UL				skewness^{10}

in the case of mobile clients. Otherwise, the variations in the time series, e.g. in CQI or downlink throughput, cannot be captured accurate enough.

E. Discussion

Our results reveal the monitoring information an MNO needs to collect, given possible limitations in the monitoring infrastructure. For instance, while it is crucial to monitor the UE's donwlink throughput and beneficial to include the clients' uplink CQI, monitoring the AN downlink throughput brings only few added value. With the mobility of clients, we studied a possible influence factor in terms of feature relevance and we could observe that features expressing variation gain more importance when clients move. This indicates that in mobile cases, a higher monitoring frequency is necessary so that the features can reliably capture the time series' variations. Finally, we highlighted the benefits of applying ML in 5G networks for QoE estimation, as it can outperform other approaches, that do not or less rely on data analytics. Although those mechanisms can also be applied in mobile networks of previous generations, only 5G mobile networks provide the relevant infrastructure for data analytics and information exchange, e.g., via a third party AF.



(a) MSE (b) MedAE (c) R^2

Fig. 3: Influence of number of monitoring types on the QoE estimation performance for the different datasets.

So far, our basic set-up does not consider further possible influence factors, such as traffic from other applications than video streaming, varying system loads, or hand-overs for clients moving from one cell to another. In multi-cell deployments, the variations in terms of QoE for the mobile clients might increase, as the QoE might negatively be affected for the fraction of clients which experiences hand-overs. Varying the number of active UEs in a cell would probably increase the importance of features related to the AN throughput, which remains relatively constant for the current experiments.

V. Conclusion

The new network functions introduced in the 5G architecture, NWDAF and 3rd party AFs, enable data-analytics driven QoE estimation for MNOs. This work highlighted the prospects of using machine learning, revealed the relevant statistics that must be available at NWDAF in order to obtain a certain QoE estimation accuracy, and examined the added value of providing certain monitoring information to the NWDAF. Additionally, we studied the mobility of clients as a factor that influences the feature relevance and hence, the accuracy vs. cost trade-off for an ML-based QoE estimation.

For future work, we plan to vary the numbers of active UEs and to consider multi-cell deployments to study the effects of varying system loads and hand-overs. We furthermore plan to study the requirements on the input data, i.e., the necessary monitoring frequency, so to still allow an accurate QoE estimation. This enables a more in-depth evaluation of the trade-off, which considers the impact of the monitoring granularity on costs and estimation accuracy.

Acknowledgement

This work is funded by the BMBF Software Campus Grant "BigQoE" (01IS17052). SN is supported by BMBF as BIFOLD - Berlin Institute for the Foundations of Learning and Data (01IS18025A/01IS18037A). The authors want to thank Huiran Liu und Marcin Bosk for their continuous support and Frank Alexander Kraemer for his valuable feedback.

References

[1] "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," White Paper, Cisco Syst., Inc., 02 2018.

[2] 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 *Internet-QoE 2019*. ACM.

[3] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "A machine learning approach to classifying youtube qoe based on encrypted network traffic," *Multimedia tools and applications*, 2017.

[4] M. Khokhar, T. Ehlinger, and C. Barakat, "From network traffic measurements to qoe for internet video," in *IFIP Networking Conference*, 2019.

[5] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, "emimic: Estimating http-based video qoe metrics from encrypted network traffic," in *TMA 2018*. IEEE, 2018.

[6] M. Seufert, P. Casas, N. Wehner, L. Gang, and K. Li, "Stream-based machine learning for real-time qoe analysis of encrypted video streaming traffic," in *ICIN 2019*. IEEE, 2019.

[7] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring video qoe from encrypted traffic," in *IMC 2016*. ACM, 2016, pp. 513–526.

[8] S. Wassermann, M. Seufert, P. Casas, L. Gang, and K. Li, "I see what you see: Real time prediction of video quality from encrypted streaming traffic," in *Internet-QoE 2019*. ACM, 2019, pp. 1–6.

[9] 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 *CNSM 2018*. IEEE, 2018.

[10] 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 *MMSys 2018*. ACM, 2018, pp. 466–471.

[11] Y. Lin, E. Oliveira, S. Jemaa, and S. Elayoubi, "Machine learning for predicting qoe of video streaming in mobile networks," in *ICC 2017*. IEEE, 2017.

[12] T. Begluk, J. B. Husić, and S. Baraković, "Machine learning-based qoe prediction for video streaming over lte network," in *INFOTEH 2018*. IEEE, 2018, pp. 1–5.

[13] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Machine learning at the edge: A data-driven architecture with applications to 5g cellular networks," *arXiv preprint arXiv:1808.07647*, 2018.

[14] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5g core network scalability: A machine learning approach," *IEEE Network*, vol. 32, no. 6, pp. 42–49, 2018.

[15] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Deepcog: Cognitive network management in sliced 5g networks with deep learning," 2019.

[16] J. Nightingale, P. Salva-Garcia, J. Calero, and Q. Wang, "5g-qoe: Qoe modelling for ultra-hd video streaming in 5g networks," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 621–634, 2018.

[17] A. Martin, J. Egaña, J. Flórez, J. Montalbán, I. Olaizola, M. Quartulli, R. Viola, and M. Zorrilla, "Network resource allocation system for qoe-aware delivery of media services in 5g networks," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, 2018.

[18] 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.

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

[20] A. Udugama, B. Khalilov, A. B. Muslim, and A. Förster, "Implementation of the swim mobility model in omnet++," *arXiv preprint arXiv:1609.05199*, 2016.

[21] 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 *QoMEX 2017*. IEEE, 2017, pp. 1–6.