

Towards Trustworthy AI for QoE prediction in B5G/6G Networks

José Luis Corcuera Bárcena¹, Pietro Ducange¹, Francesco Marcelloni¹,
Giovanni Nardini^{1,2}, Alessandro Noferi¹, Alessandro Renda¹, Giovanni Stea¹ and
Antonio Viridis¹

¹Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy

²Center for Logistic Systems, University of Pisa, Via dei Pensieri 60, 57124 Livorno, Italy

Abstract

The ability to forecast Quality of Experience (QoE) metrics will be crucial in several applications and services offered by the future B5G/6G networks. However, QoE timeseries forecasting has not been adequately investigated so far, mainly due to the lack of available realistic datasets. In this paper, we first present a novel QoE forecasting dataset obtained from realistic 5G network simulations and characterized by Quality of Service (QoS) and QoE metrics for a video-streaming application; then, we embrace the topical challenge of trustworthiness in the adoption of AI systems for tackling the QoE prediction task. We show how an eXplainable Artificial Intelligence (XAI) model, namely Decision Tree, can be effectively leveraged for addressing the forecasting problem. Finally, we identify federated learning as a suitable paradigm for privacy-preserving collaborative model training and outline the related challenges from both an algorithmic and 6G network support perspective.

Keywords

Machine learning, B5G/6G networks, QoE forecasting, Explainable AI, Federated learning,

1. Introduction

The development of Beyond 5G (B5G) and 6G networks is currently underway, as testified by the activity of international projects such as Hexa-X¹. The advent of such new technologies is expected to pave the way to innovative services that will pose stringent quality requirements. For example, automotive applications, such as see-through or tele-operated driving [1, 2], require the transmission of high-definition videos in real time. The Quality of Experience (QoE) perceived by the end users determines whether the service can be provided or not. In computer networks - including B5G/6G ones - QoE depends on the Quality of Service (QoS) provided by the network itself, although direct mapping between QoS and QoE is not straightforward [3].

AI6G'22: First International Workshop on Artificial Intelligence in beyond 5G and 6G Wireless Networks,

July 18-23, 2022, Padua, Italy

✉ joseluis.corcuera@phd.unipi.it (J. Corcuera Bárcena); alessandro.noferi@phd.unipi.it (A. Noferi);

{name.surname}@unipi.it (all other authors)

🆔 0000-0002-9984-1904 (J. Corcuera Bárcena); 0000-0003-4510-1350 (P. Ducange); 0000-0002-5895-876X (F. Marcelloni); 0000-0001-9796-6378 (G. Nardini); 0000-0002-0482-5048 (A. Renda); 0000-0001-5310-6763 (G. Stea); 0000-0002-0629-1078 (A. Viridis)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹<https://hexa-x.eu/>, accessed March 2022

For this reason, Artificial Intelligence (AI) and in particular Machine Learning (ML) approaches are foreseen as a valid mean to predict QoE from QoS.

Some recent works on the topic [3, 4] have relied on aggregate metrics and have not addressed the problem of predicting QoE in near real-time over input data streams. The main roadblock toward this goal is the lack of available realistic datasets featuring QoS and QoE timeseries. Indeed, obtaining a reliable dataset related to mobile network applications is difficult, because network operators are hardly willing to share information about their network. Even if a subset of measurements is provided, their level of detail may not be the required one. Simulations are a viable option to overcome the above limitations: they can replicate custom (large-scale) network deployments with custom parametrization (e.g., number of users or amount of traffic). This allows AI systems to accurately model a variety of network conditions that may not be observed during a limited time frame in the real network.

However, the design of AI systems cannot be simply targeted to optimize accuracy and technical robustness but must also comply with additional requirements towards *trustworthy* AI [5], such as transparency of AI models and privacy of data owners. The former requirement is at the core of the eXplainable Artificial Intelligence (XAI) field [6], concerned with producing details and reasons regarding the functioning of a model, by either ensuring its interpretability by design, or exploiting post-hoc techniques. On the other hand, the privacy requirement poses serious challenges to the training stage of data-hungry ML models and may require to revisit the learning paradigms: Federated Learning (FL) [7] has been recently proposed as a solution to enable collaborative training of AI models while preserving the privacy of data owners.

In this work we approach the problem of QoE prediction in future B5G/6G networks from different perspectives. Specifically, the contributions of this paper are threefold: *i*) we use realistic simulations to make available a dataset that includes QoS and QoE metrics for a video-streaming application in a 5G network environment; *ii*) we present a preliminary analysis on the above dataset exploiting an XAI model to predict QoE; *iii*) we argue that FL is suitable to predict QoE in a privacy-preserving way, and we discuss how 6G networks should support it.

The rest of this paper is organized as follows: Section 2 formulates our problem and describes the dataset we obtained from realistic 5G network simulations. A preliminary approach for QoE prediction with an XAI model is presented in Section 3, whereas Section 4 highlights the pathway towards FL in future 6G networks. Finally, we draw conclusions in Section 5.

2. QoE prediction in B5G/6G networks

In this section, we present the problem of QoE prediction in B5G/6G networks. We also present a simulated QoE dataset and discuss the assumptions we made to generate it.

2.1. Problem statement

We consider an automotive environment where connected vehicles are User Equipments (UEs) of the mobile network, and are attached to their respective base station, or gNodeB (gNB). UEs play real-time video streams whose perceived quality is a relevant factor to determine the availability of some advanced driving assistance system. For example, in a see-through application, a vehicle receives the live video acquired by the camera of the vehicle in front of it,

which helps the driver overtake. However, such a service is only safe if one can rely on the video to be displayed continuously and with high quality for the entire duration of the maneuver, which may require several seconds. Thus, we need to predict the QoE perceived by UEs in the next future by leveraging real-time QoS and QoE data generated by the UEs themselves.

We assume that each UE collects both QoS (e.g., SINR) and QoE (e.g., percentage of frames that are correctly displayed) metrics while receiving the video stream. Context information like UE's position and speed can also be collected and used as factors to train an ML model. The above metrics are collected by each UE at discrete time periods, hence for each metric the UE generates a vector whose i -th element is the value of the metric collected at time i . Network-wise metrics like average utilization of the cell can also be leveraged to provide more meaningful predictions. All the above metrics gathered by the UE and the network will be used to predict the value of a *target QoE metric* (or a set thereof) at a time in the future.

To support the above prediction, we need to train an ML model with a dataset that includes realistic QoS and QoE metrics obtained from the mobile network. Since mobile network operators are hardly available to share data about their users and network, we resorted to network simulations to generate the relevant dataset.

2.2. Generation of the dataset

Simulations are carried out with Simu5G [8], an open-source *model library* for the OMNeT++ simulation framework² that enables the evaluation of end-to-end performance of 5G-enabled applications. Within Simu5G, we implemented a client-server video-streaming application. The server sends a video stream to the client following a trace-based approach: sending rate, size and type of video frames are read from a trace file generated from real videos via a dedicated command of the FFmpeg library³. Traces were obtained from three dash-camera videos, so as to reproduce a see-through scenario.⁴ Before sending the frames over the network, the server fragments them in packets, which are then transmitted via the Real-time Transport Protocol (*RTP*). RTP packets received by the client are then played out at their corresponding playout time. We configured the client with 100ms-playout delay: this is a tradeoff between the real-timeliness of the video streaming and the buffering time required to prevent stalls.

Figure 1 shows the simulation scenario. Seven gNBs are deployed in a regular hexagonal grid with inter-gNB distance of 500m. Fifteen UEs are deployed randomly over the floorplan and connect to the gNB they receive the highest power from. Each of them runs the client side of the video streaming application, whereas their server-side counterparts reside on a remote host connected to the 5G core network. Each client receives a different video trace, obtained by starting one of the three above-mentioned dash cam videos at different times. To create realistic load conditions, each gNB also sends $50 \frac{kB}{s}$ of downlink traffic to 30 *background UEs*. We also simulate an additional tier of *background cells*, each serving 30 background UEs, in order to generate realistic interference to the UEs attached to the seven central gNBs [9]. We run 24 independent replicas of an 120-second simulation, collecting time-tagged metrics from the 15 UEs in the seven central cells. Metrics are summarized in Table 1.

²OMNeT++ Website: <https://omnetpp.org>, accessed March 2022

³ffprobe -show_frames: online documentation <https://ffmpeg.org/ffprobe.html>, accessed March 2022

⁴<https://bit.ly/3iN651q>, <https://bit.ly/35n9eIO>, <https://bit.ly/3IT5g24>, accessed March 2022

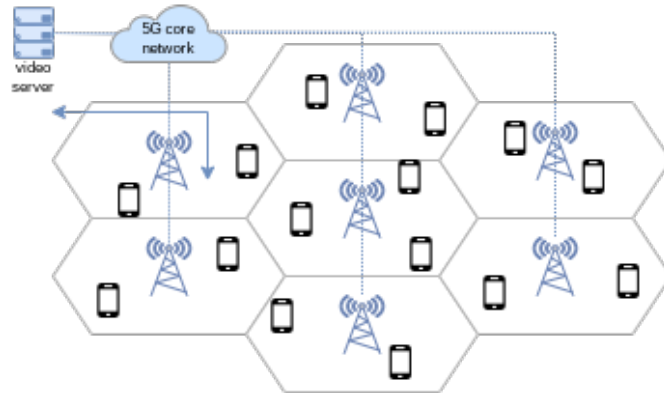


Figure 1: Simulation scenario

Table 1

Description of the metrics included in the dataset.

Name	Level	Description
Context		
UE position	Application	(x, y, z) coordinates of the UE in the floorplan
UE speed	Application	speed of the UE in $\frac{m}{s}$
QoS metrics		
avgServedBlocksDL	Network	number of Resource Blocks occupied in downlink
averageCqiDL	Network	CQI values reported in DL
rcvdSinrDL	Network	SINR value measured at packet reception
servingCell	Network	ID of the new serving cell after the handover
frameSize	Application	size of the displayed frame (Byte)
rtpPacketSize	Application	size of the RTP packet (Byte)
end2EndDelay	Application	time between transmission and reception of an RTP packet
interArrivalTimeRtp	Application	interarrival time between two RTP packets
rtpLoss	Application	RTP packets of frame lost
QoE metrics		
framesDisplayed	Application	frame percentage arrived at the time of its display
playoutBufferLength	Application	frame buffer size
firstFrameElapsedTime	Application	3 values: 1) timestamp of the UE request, 2) timestamp of the sender ACK, 3) time between the request and the first frame displayed

The resulting dataset consists of 5568 rows, each being a tuple with six fields: *run* is the ID of the replica; *network_parameters* include the variables describing the simulation configuration (e.g., the scheduling algorithm); *module* is the network entity (e.g., ue[0]) that recorded the metric; *statistic* is the name of the recorded metric; *values* is a vector including the values recorded for the above metric; *timestamp* is a vector whose elements are the timestamps of the corresponding elements of the *values* vector. The dataset and the related simulation configuration are available at *url available upon paper acceptance*.

3. QoE prediction as a regression problem

Although the forecasting literature has traditionally been dominated by statistical methods based on linear processes, ML methods are currently gaining increasing attention due to their high modelling capability especially on large datasets [10]. In this preliminary analysis we focus on the latter set of approaches and formulate our QoE prediction task as a *regression* problem. First, we describe the preprocessing steps designed to transform the original raw dataset into a regression dataset, suitable for the downstream adoption of traditional ML approaches. Then, we show the results of a first experimental analysis, using a Decision Tree as regression model. The choice of this model is dictated by the *explainability* requirement for trustworthy AI. Tree-based models are generally considered among the most inherently explainable classification and regression models; their interpretability, however, depends on several factors [4] and a thorough analysis of this aspect will be the cue for future investigations.

3.1. Data preprocessing

We observe that the time-tagged metrics in the dataset are not aligned on the same timestamps for two main reasons: *i*) only few of them have fixed sampling interval, and *ii*) some values are missing at specific times due to, e.g., connection drops. Thus, a preprocessing stage is required.

As a first step, the timeseries available to the UEs have been identified. It is worth underlining that the *avgServedBlocksDL* statistic can be retrieved by the UEs based on the available information on the *ServingCell*. In other words, for each UE, we build a new timeseries, namely *avgServedBlocksDL_UE*, by exploiting the handover information and concatenating the *avgServedBlocksDL* timeseries fragments taken from the cells serving the UE itself. For example, a UE may obtain this information through the services available in a MEC-enabled architecture.

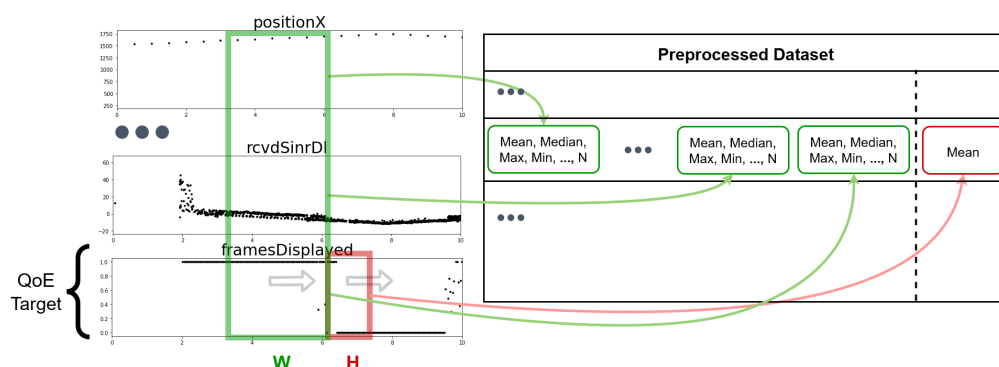


Figure 2: Preprocessing steps: the QoE prediction task as a regression problem.

Fig. 2 shows the procedure used for a preliminary analysis with ML techniques for the QoE-prediction task. As an example, the first ten seconds of the timeseries from three metrics are shown, namely *positionX*, *rcvdSinrDL* and *framesDisplayed* (QoE target metric). To obtain any record of the preprocessed dataset we compute statistics within a window W over historical data of each variable. Specifically, mean, median, max, min, variance, standard deviation, kurtosis, skewness, Q1 and Q3 are computed, and the number of samples used for the estimates is stored. The associated target value is the mean of the *frameDisplayed* variable over the time horizon

of size H (one step ahead forecasting). The subsequent record is obtained by sliding the two windows with a step H . Each instance is thus represented in \mathbb{R}^{132} (11 statistics evaluated over window of size W on 12 timeseries) and is associated with the target QoE (average value of *frameDisplayed* over window of size H). In this analysis we focus on timeseries metrics, therefore we do not include the values of *firstFrameElapsedTime* in the model.

3.2. Preliminary experimental analysis: setup and results

The dataset has been divided in training and test sets for a first experimental evaluation campaign: 20 runs are grouped to form the training set, whereas the remaining 4 runs represent the test set. A distinctive trait of the dataset is that some values are missing (e.g., because a simulated UE lost connectivity at some point). For the purpose of the present work, we simply discard such records from both the training and the test sets.

As per the choice of the XAI model, we resort to the Python implementation of the Decision Tree (DT) for regression available in scikit-learn⁵. We set the windows size $W = 10s$ and $H = 1s$ and tested with different split criteria. For the sake of brevity, we only report the results of the best configuration obtained by using MSE as split criteria, 0.01 as the fraction of samples required to split an internal node and 0.001 as the fraction of samples required to be at a leaf node. Table 2 reports the performance of the trained model measured in terms of MSE, MAE and coefficient of determination (R^2) for different values of maximum depth of DTs (5,10,15); also, we report the complexity of the models measured in terms of number of nodes, number of leaf nodes and number of features selected in the induced tree.

Table 2

Global results and model complexity for different values of maximum DT depth. Regression metrics are evaluated on whole training and test sets. Best values are highlighted in bold.

Max Depth	Regression metrics						Model complexity		
	MSE		MAE		R^2		Nodes	Leaves	Features selected
	<i>train</i>	<i>test</i>	<i>train</i>	<i>test</i>	<i>train</i>	<i>test</i>			
5	0.1038	0.1105	0.2581	0.2661	0.4080	0.3565	57	29	18
10	0.0817	0.1019	0.2134	0.2418	0.5340	0.4065	303	152	65
15	0.0791	0.1040	0.2082	0.2424	0.5489	0.3944	401	201	69

The reported results should be intended as a first baseline on our QoE forecasting dataset. It is interesting to note that the DT at depth 10 achieves the best generalization capability, with a MAE on the test set lower than 25%. An increase in depth does not bring any benefit: the complexity of the model increases leading only to a more severe overfitting. Conversely, the most compact DT exhibits a slight reduction in the regression metrics but is significantly less complex, and therefore more interpretable, compared to deeper trees.

To better assess the quality of model predictions it is worth looking at the timeseries: Fig. 3 reports the examples of QoE timeseries for two UEs, featuring both the ground-truth and the predicted values obtained by the most accurate model (depth 10). The visual analysis of

⁵<https://scikit-learn.org/>, accessed March 2022

Fig. 3 suggests that the model provides reasonable predictions in different scenarios, namely when the timeseries are plagued simply by one or by several events of QoE degradation. In the latter case, regression metrics are comparably lower, likely due to the latency of the model in identifying the transition between QoE levels.

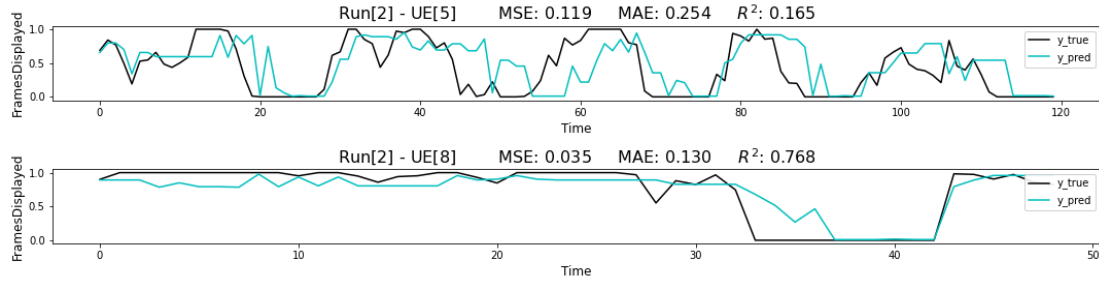


Figure 3: Real and predicted values of QoE for two example UEs of the test set.

Although the preliminary results can be considered promising, we emphasize a few aspects that deserve further investigation. First, a comparative analysis covering multiple models and parameters configuration should be carried out; in particular, the impact of the choice of windows size W and H must be assessed. Second, an appropriate strategy for handling missing values should be devised. Third, an analysis on which metrics are relevant for the QoE forecasting task is advocated. Finally, it should be noticed that the adopted approach stems from a strong assumption: we build a global training set by collecting data produced by different sources. In the following section we present a solution to tackle the QoE prediction task when this last assumption is too strict or unachievable.

4. Federated Learning for QoE prediction

Collecting peripheral data for processing and training on a centralized server is often impractical due to the resulting communication overheads and the disclosure of UEs' private data. The preservation of data owners' privacy is a crucial requirement towards the realization of trusted AI-empowered B5G/6G networks: it becomes thus essential to leverage novel paradigms, such as FL, that enable collaborative model training between UEs without any sharing of raw data with each other or with other parties. FL perfectly fits the scenario described in this paper. In the following we outline the challenges associated with the fulfillment of the trustworthiness requirement through the federated learning of inherently explainable AI models (Fed-XAI), and we define an architecture to support Fed-XAI operations in a 6G framework.

4.1. Enhancing users' trust: Federated Learning of XAI models

As anticipated in Section 1, realization of trustworthy AI entails compliance with several requirements, including privacy and transparency [5]. While the *privacy* requirement is natively satisfied by the FL paradigm, the *transparency* one strongly depends on the specific model adopted and its ability to provide explanations on any decision made. Most of the existing FL approaches leverage the federated setting for collaborative training neural networks (NNs)

and deep learning (DL) models, which are often referred to as *opaque* or *black box* models. Conversely, FL of explainable-by-design AI models, such as DTs and Rule Based Systems (RBSs), has not been adequately investigated so far. The concept of Fed-XAI (i.e., FL of XAI models) aims to fill this gap by enhancing users' trust in AI-empowered future 6G networks. Whenever the FL process is orchestrated by a central entity, a possible implementation of Fed-XAI consists in (i) local learning of XAI models by data owners, (ii) local models transmission to the central server, (iii) models aggregation by the central server, and (iv) global model transmission to the data owners for local inference. The aggregation step is the major challenge towards Fed-XAI: appropriate procedures for merging DTs and RBSs should be devised as their learning stage is not based on the optimization of a differentiable global objective function (as it is the case with NNs and DL models) and the well established federated averaging protocol (FedAvg) [7], designed for collaborative gradient-based optimization, cannot be immediately applied.

4.2. 6G network support to Federated Learning

We envision that UEs of future 6G networks will participate in FL processes following the *as-a-service* paradigm. To this aim, the 6G network must provide new protocols that handle the interactions among the entities involved in the FL framework, such as querying the list of available FL processes and joining one of them, as well as participating in the training and obtaining a model. In the following, we refer to an FL process as a collaborative learning task dedicated to a specific application (e.g., QoE prediction for automotive applications).

Our proposed logical architecture for Fed-XAI in 6G networks is depicted in Fig. 4. Each UE is supported by a *Fed-XAI local manager*, which interacts with the FL framework on behalf of the UE application. It manages both the *learning* and *inferencing* modules of the UE. When the UE wants to join (or leave) an FL process, its Fed-XAI local manager queries the *Fed-XAI service provider*, which is the module that maintains the overall view of the FL processes available in the system. The Fed-XAI service provider also takes care to orchestrate the entities that will actually execute the FL processes. In particular, each active FL process is composed of two modules, namely the *Fed-XAI controller* and the *Fed-XAI computation engine*. The former manages control-plane interactions with the Fed-XAI service provider (e.g., authorization grants) and the Fed-XAI local manager, whereas the latter acts as the FL *aggregator*. Indeed, the Fed-XAI computation engine exchanges local and global model updates with the learning submodules of UEs' Fed-XAI local manager, which in turn act as FL *collaborators*.

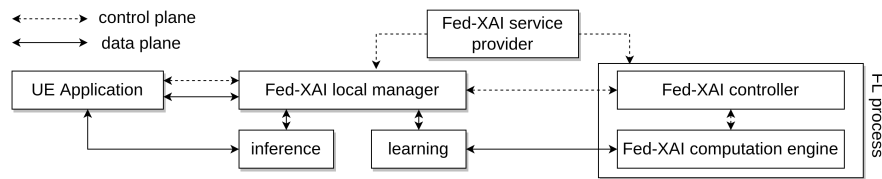


Figure 4: Fed-XAI architecture

It is worth noting that the deployment of the above entities is immaterial. The Fed-XAI service provider may reside either in the cloud or at the edge of the 6G network. Likewise, the Fed-XAI local manager may reside at either the UE device or at the edge. This last option may

be necessary with resource-constrained UEs, e.g. IoT devices.

5. Conclusions

In this work, we have presented a novel dataset obtained through realistic 5G network simulations for QoE forecasting in B5G/6G networks. Moreover, we have discussed some preliminary QoE forecasting results achieved by a Decision Tree for regression as an inherently explainable model. We have experimentally highlighted the adequacy of the adopted approach as a baseline for the QoE forecasting task. Finally, we have discussed the implications of extending the XAI model towards an FL approach, from both an algorithmic and a network perspective. Future work will include designing a Fed-XAI-based approach to tackle the prediction of QoE in B5G/6G networks, as well as evaluating the impact of network transport on the performance of XAI models, and vice versa.

Acknowledgments

We acknowledge the support of: the Italian Ministry of University and Research (MIUR), in the framework of the Cross-Lab project (Departments of Excellence) and PON 2014-2021 “Research and Innovation”, DM MUR 1062/2021, Project title: “Progettazione e sperimentazione di algoritmi di federated learning per data stream mining”; the Center for Logistic Systems of Livorno; the EU Commission through the H2020 projects Hexa-X (Grant no. 101015956).

References

- [1] C-V2X Use Cases and Service Level Requirements Vol. I, Technical Report, 5GAA, 2020.
- [2] C-V2X Use Cases and Service Level Requirements Vol. II, Technical Report, 5GAA, 2021.
- [3] V. Vasilev, J. Leguay, S. Paris, L. Maggi, M. Debbah, Predicting QoE Factors with Machine Learning, in: 2018 IEEE Int’l Conf. on Communications (ICC), 2018, pp. 1–6.
- [4] A. Renda, P. Ducange, G. Gallo, F. Marcelloni, XAI Models for Quality of Experience Prediction in Wireless Networks, in: 2021 IEEE Int’l Conf. on Fuzzy Systems (FUZZ-IEEE), IEEE, 2021, pp. 1–6.
- [5] E. Commission, C. Directorate-General for Communications Networks, Technology, Ethics guidelines for trustworthy AI, Publications Office, 2019.
- [6] A. B. Arrieta, et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information fusion* 58 (2020) 82–115.
- [7] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, *ACM Trans. on Intelligent Systems and Technology (TIST)* 10 (2019) 1–19.
- [8] G. Nardini, D. Sabella, G. Stea, P. Thakkar, A. Viridis, Simu5G—An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks, *IEEE Access* 8 (2020) 181176–181191.
- [9] G. Nardini, G. Stea, A. Viridis, Scalable Real-Time Emulation of 5G Networks With Simu5G, *IEEE Access* 9 (2021) 148504–148520.
- [10] V. Cerqueira, L. Torgo, C. Soares, Machine learning vs statistical methods for time series forecasting: Size matters, *arXiv preprint arXiv:1909.13316* (2019).