Received 14 April 2024; revised 26 May 2024; accepted 26 June 2024. Date of publication 3 July 2024; date of current version 25 July 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3422872

ML-Powered KQI Estimation for XR Services: A Case Study on 360-Video

O. S. PEÑAHERRERA-PULLA (Member, IEEE), CARLOS BAENA, SERGIO FORTES (Senior Member, IEEE), AND RAQUEL BARCO

Telecommunication Research Institute (TELMA), E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga, 29010 Málaga, Spain

CORRESPONDING AUTHOR: S. FORTES (e-mail: sfr@ic.uma.es)

This work was supported in part by the Ministerio de Asuntos Económicos y Transformación Digital and European Union—NextGenerationEU within the framework "Recuperación, Transformación y Resiliencia y el Mecanismo de Recuperación y Resiliencia" under Project MAORI; in part by the Universidad de Málaga through the "II Plan Propio de Investigación, Transferencia y Divulgación Científica"; in part by the Junta de Andalucía through Secretaría General de Universidades, Investigación y Tecnología under predoctoral Grant PREDOC_01712; and in part by the Ministerio de Ciencia y Tecnología under Grant FPU19/04468.

ABSTRACT The emergence of cutting-edge technologies and services such as Extended Reality (XR) promises to change how people approach everyday living. At the same time, the emergence of modern and decentralized architectural approaches has ushered in a new generation of mobile networks, such as 5G, as well as outlining the roadmap for B5G (Beyond-5G) and further advancements. These networks are expected to be the enablers for the realization of the metaverse and other futuristic services. In this context, quantifying the service performance is a key enabler for dynamic, environment-adaptive, and proactive network management. This work presents an ML-based (Machine Learning) framework that uses data from the network, such as radio measurements, statistics, and configuration parameters to infer the best ML models that fit diverse XR Key Quality Indicators (KQIs). The output models integrate feature engineering techniques that enhance model size and performance. The proposed framework comprises data preprocessing, model definition, training, tuning, and validation. Additionally, to select the best combination algorithm this work introduces a metric called PET_{score}, which evaluates algorithm candidates in terms of error performance and prediction time. These are considerations that are needed for time-sensitive services like XR's. To validate our proposal, the 360-video service has been chosen to demonstrate the potential of this ML framework with a real XR use case. In addition, the dataset generated for the use case evaluation is publicly accessible and properly referenced. Furthermore, this work serves as a foundation for future research on end-to-end (E2E) quality of experience (QoE)-based network management in conjunction with other enabling technologies, including network slicing, virtualization, and multi-access edge computing (MEC).

INDEX TERMS Machine learning, mobile communications, extended reality, virtual reality, quality of experience, multimedia, 360-video, key quality indicators, 5G, B5G.

I. INTRODUCTION

THE NEW generation of services aims to revolutionize our day-to-day activities as well as the way people interact with each other. These novel services, which involve cutting-edge multimedia technologies like Extended Reality (XR), are intended to bring different levels of virtual and enriched experiences to our lives. Although the "virtual" approach has been a topic discussed since decades ago, the enabler technologies were not as ready as they are nowadays.

In this context, the implications of XR in real life are expected to be omnipresent in all tasks and human activities. For instance, the metaverse concept, recently reinvigorated by the META company [1], looks to bring physical human interactions (e.g., meetings, entertainment, shopping, etc.) to the virtual world in a real-like manner. In concordance with the development of new radio mobile technologies, these implications introduced opportunities to integrate this kind of service into the network. This perspective will generate new exploitation ways of these features by vertical vendors and network operators in an organized and standardized manner. With this in mind, it is possible to create an E2E (End-to-End) scenario where several parties are involved.

XR is an umbrella term that involves different sub-technologies regarding the level of abstraction of reality. On one side, Augmented Reality (AR) is focused on overlying virtual elements (e.g., information, rendered objects, etc.) to interact with physical reality. To reach this, the physical reality is captured and processed to generate models that feedback on the experience. On the other side, Virtual Reality (VR) aims to generate a whole alternative experience, where every element is generated virtually, rendered, and displayed to the user. However, information from the physical information is required (i.e., user tracking). Finally, the mixture of both technologies is known as Mixed Reality (MR). In this way, MR aims to introduce physical and virtual features simultaneously, this way deploying a different degree of immersion (e.g., a real human avatar inside a 3D and fully virtualized environment) [2].

Although some top companies, such as Meta, Apple, and Samsung, among others, are working on this topic, delivering content from servers to user equipment is not trivial. This is because VR requirements are stricter than traditional services like plain video. The absence of adequate resources can result in not only a poor user experience but also physical issues such as cybersickness [3], confusion, anxiety, fatigue, and even physical injuries [4], [5], [6].

To address these challenges, the new generation of mobile networks aims to convert XR services into native ones making use of the network and computational resources based on their requirements. This approach will provide the networks with mechanisms that ensure proper levels of quality for each service based on automatic and intelligent resource policies. For this purpose, different enabler technologies and features of 5G and 6G will be used, such as Network Slicing (NS), Network Functions Virtualization (NFV), Software-Defined Networks (SDN), Software-Defined Radio (SDR), Mobile Edge Computing (MEC), and Artificial Intelligence/Machine Learning (AI/ML) [7]. Another important concept that is drawing attention from researchers, operators, and vertical vendors is the Open RAN trend [8], which is intended to deploy fully intelligent, virtualized, and interoperable mobile networks.

With this in mind, it is necessary to quantify the performance of the services, so actions can be taken if required to improve the E2E experience. Traditionally, the way to evaluate the performance of services has been to use subjective approaches, such as the Mean Opinion Score (MOS). However, the main disadvantage of using subjective metrics is they are based on subjective perceptions that can be biased by different impact factors [9].

To minimize this, the use of objective metrics, such as the KQIs, is recommended. These metrics allow quantifying the performance from a user-centric perspective as well but using only objective data that is measurable from the operator's point of view. The main issue of using KQIs is that they are service-specific. For example, a multimedia service will depend on visual, audio, and latency metrics, while a traditional one such as file transfer will rely on upload/download times, and connection speed, among other metrics.

The variety of services that XR will bring to reality will increase the complexity for service and network providers to handle them. This is the perfect scenario that highlights ML as a powerful tool to pave the way for intelligent management of the network. The state of the art shows ML supporting estimation of subjective metrics [10], [11], image quality [12], [13], leveraging alternative streaming strategies [14], or detecting failures in media visualization [15]. As far as the authors' knowledge, there is no previous research that used an ML framework to infer automatically the best ML algorithms to predict KQIs for XR services using only network-side data.

Hence, the key contribution of this work focuses on the development of a novel ML-powered framework to estimate KQIs from XR services. These metrics exploit the information contained in the network, such as radio measurements, statistics (e.g., Key Performance Indicators - KPIs), and configuration parameters to support the management of service-oriented new-generation mobile networks.

To that end, this work presents a framework that integrates the stages, such as preprocessing, training, validation, hyperparameter tuning, assessment, and model selection. Consequently, this framework outputs the best model that combines feature engineering techniques, algorithms, and hyperparameters per target KQI. To establish an objective criterion, the evaluation metrics are the prediction ability (i.e., error) and prediction time. Since selecting a model is not a trivial task, we introduce in this work a metric called PET_{score}, which evaluates the models in terms of both mentioned factors. Finally, the 360-video service has been selected as a case study to evaluate the potential of our framework. The results yield intriguing conclusions and insights that may inform future research endeavors. In addition, the dataset generated for this use case is made available to the research community as a tool to encourage and support the development of this research line [16].

This research work aims to encourage the research community to leverage ML as a means of enhancing the intelligence of mobile networks. In this sense, the framework is extensible to any XR service. Consequently, the administration of service quality can be conducted from a user-centric standpoint by employing objective metrics derived from network data. This can be achieved by utilizing the most appropriate ML-leveraged models that can accurately reflect the reality of the user. For instance, user-customized digital twins may benefit from this approach. This approach aims to provide additional tools to meet the expected 5G flexibility, reliability, and scalability, as well as the fulfillment of XR requirements in the near future.

In this way, this paper is organized as follows. First, Section II provides an outlook of the state of the art related to the use of ML for QoE and its transition to KQI approaches. Then, Section III describes the ML framework involving data preprocessing, model definition, training, tuning, and

validation and assessment. After that, Section IV provides a viewpoint on the 360-video service use case. Then, the proceeding for data generation and collection to create a dataset is described. Finally, in Section V the ML framework is assessed through the dataset generated in the last section. This evaluation explicitly shows the performance of the best ML algorithm outputted by the framework in terms of error performance, prediction time, and PET_{score}. Then, Section VI details some conclusions of the work, exposing the key points of this research as well as outlining some future work.

II. RELATED WORK

Over time, the delivery of services has evolved from a best-effort approach to methodologies that guarantee a certain level of quality. However, how to quantify the quality of a service has generated a plethora of options, some adequate for legacy networks and services, and some new approaches that promise to endow networks with additional degrees of intelligence. In this regard, ML has drawn attention to research owing to its capacity to address problems where traditional approaches cannot. One of these problems is the network and resource management for new network technologies like 5G and B5G, where it is intended to integrate dynamic methodologies to support time-sensitive services, such as XR.

This section provides an overview of QoE and why it is expected to transition to KQIs in the context of mobile networks. It then addresses the state of the art, describing the current state of research on KQIs and highlighting ideas that support why KQIs are expected to be an enabler for new generation network management, such as 5G and B5G.

A. QUALITY OF EXPERIENCE

QoE is defined by the standardization body 3GPP (Third-Generation Partnership Project) as the measurement of the *"degree of delight or annoyance of the user of an application or service"* [17]. From this point, offering a high-quality and value-added service is one of the main objectives for operators and service providers at the current times. In this scope, ML has been introduced as a useful tool for improving the quality of the services. Following these lines, different applications for ML are mentioned in the state of the art to approach this topic.

In this regard, [18] proposed the estimation of QoE metrics from in-band encrypted packet information. This data is obtained using tcpdump to compute window statistics (e.g., throughput, inter-arrival time, packet size, etc.). However, the experimental setup featured a WiFi deployment that emulates radio mobile network conditions using previously known 4G/5G traffic patterns for HTTPS (HyperText Transfer Protocol Secure) and QUIC (Quick UDP Internet Connections) encrypted content. The work in [19] presents a MOS estimation scheme for video streaming services. In their proposal, the MOS is estimated through an ANN (Artificial Neural Network) whose inputs are typical QoS metrics such as delay, jitter, and packet loss. The original MOS values were gathered by testing people using a mobile phone using an emulated LTE network.

The authors in [20] propose an ML approach to manage decision-making in the context of DASH (Dynamic Adaptive Streaming over HTTP) video streaming using SDN. This work is based on the use of ML to map the MOS from the KPIs of the network. Then, an orchestrator decides which high-level policy should be taken into account by network elements to manage the policies and strategies (e.g., routing). The data is gathered using a collector that develops traffic mirroring for processing information in an MEC. Nevertheless, mirroring traffic (traffic duplication) is becoming ineffective for network operators.

A different approach is presented by Gutterman et al. [21], where the QoE estimation for the service, particularly the YouTube video service, is done through an ML-based algorithm whose inputs are statistics extracted from IP headers.

In [22] the authors show the strong correlation of high-level view engagements with low startup times, buffering times, rebuffering number of events, and a considerably high resolution. Similarly, the authors in [23] present an ML-based mechanism to estimate the QoE through MOS. The outcome models were intended to calculate the subjective QoE using metrics such as PSNR (Peak Signal to Noise Ratio), bitrate, throughput, and VQM (Video Quality Metric), among various others. The algorithms were trained using a dataset that gathered the people's assessment of the video quality using a testbed. A different application of ML for QoE is analyzed in [24] where the authors present a strategy aiming to increase the QoE. Here, the QoE is assessed through the MOS of the video service based on an ML mechanism that manages the adaptive streaming. This approach considers the bitrate of the link to handle the buffer filling time, in this way improving the QoE. Moreover, in [25] an ML approach is developed to characterize the QoE of an HTML service through KPIs using a testbed that exploits SDN flexibility. The KPIs (e.g., bandwidth, TX (transmission), and RX (reception) load, delay, etc.) are estimated based on the network information gathered in several experiments.

Notwithstanding the wide use of the MOS, the application of subjective strategies presents some disadvantages concerning the assessment of the quality of the service. These metrics estimate the service performance from the perception of the user. This perception may be biased due to previous experiences of the user, human-related physical conditions at the moment of the evaluation (cybersickness) [26], specific preferences concerning the configuration of the service (i.e., type of content/media), expectation/reality gap, or the way it is shown (e.g., HMD or 2D screen) [9]. In addition, QoE models built on human input cannot be generalized because they depend on the user feedback for a specific service. Given this, for legacy services (e.g., voice) the assessment does not depend on a plethora of criteria like

in new-generation and immersive services like XR. This makes it difficult to evaluate the service from a similar user perspective.

Moreover, the dawn of mobile networks has updated the concept of QoE, deprecating the legacy MOS for 2nd and 3rd generations to an E2E approach. In this sense, the network performance plays a vital role that affects the overall user experience in 5G and 6G, where even the network can be considered a service (e.g., Network-as-a-Service) [27]. In the study presented in [28] it is shown that the use of QoS metrics to map QoE may not be adequate to represent real-like user's perception, thus, causing inaccurate conclusions in decision-making from operators based on false truths [28]. All these facts highlight subjective QoE as a biased [29] and inaccurate strategy to handle the concept of QoE for the wide variety of new-generation multimedia services and their deployment over mobile networks.

B. KEY QUALITY INDICATORS

The challenge introduced by the new-generation mobile networks can be approached using objective strategies based on standardized technical criteria. For this purpose, the 3GPP has introduced the use of KQIs in the last releases of LTE (Long Term Evolution) Advanced Pro and 5G. In this sense, KQIs are defined as service-specific Figures of Merit (FoM) that provide a vision of the current status of the service objectively [17]. Unlike the *traditional* methodologies used in the past by operators to quantify the degree of satisfaction or dissatisfaction with a service such as the MOS, KQIs provide a non-biased and user-agnostic perspective through service-specific criteria. In addition, the TR-28.863 3GPP technical report indicates that the KQIs can be calculated from network-layer and service-layer metrics, and even QoE metrics.

In light of multimedia services, there is a variety of metrics used to quantify and qualify multimedia services. From the legacy plain video streaming to the interactive video in the XR approach, some commonalities are generally used to establish a degree of satisfaction (e.g., resolution and frame rate). Nonetheless, determining the quality of an XR service is different because these kinds of applications are standardized as pillar services for 5G and B5G (e.g., immersive technologies, and tactile Internet, etc.). Here, the requirements for low latency and high throughput go beyond the traditional network performance management to a user-isolated E2E QoE management [27].

For instance, the ITU-T standardized a parametric model of QoE based on specific metrics derived from the bitstream of the content in P.1203 [30]. The 3GPP has released a similar approach in the TS 26.247 [31] for video streaming in the context of LTE networks. Both models make use of specific indicators captured from metadata or bitstream. Although most objective QoE video assessment techniques employ parametric, bitstream or media layer models (e.g., Human Visual System - HVS), the behavior of the models highly depends on the data used to determine the coefficients. In addition, these models are conditioned to influence factors (IF), namely the context (i.e., sex, age, place), conditions where the data was collected (i.e., temporal validity of the data, devices employed to display content), and the technologies used to transport the content (generally fixed network because it minimizes external IF). With all this, it is not possible to generalize subjective models for an E2E approach, where different external factors play a vital role in service provision.

To approach this issue, the KQIs are suitable to identify objectively if an E2E service is performing adequately exploiting network and application level information. Concerning the network side, KQIs can be derived from KPIs, which are metrics that reflect the network performance based on data (i.e., counter, alarms, flags) collected in runtime. From the user side, service metrics can be collected from the client or the server, therefore, exhibiting an objective perception of the overall service without user bias. Following these lines, the use of KQI enables an additional dimension, where the transport networks impact the performance of the service, but also provides the network operators with an extra tool to support a Network-as-a-Service (NaaS) paradigm. This idea goes in concordance with the down of the new-generation networks such as 5G and B5G, where the key idea is to open the network to be exploited as a platform by verticals and content providers. Here, KQIs can estimate the service performance using the own network data instead of specific metrics collected on the user side. Thus, it is possible to generalize network management with an additional level of intelligence through ML/AI techniques improving legacy mathematical/parametric models or biased and costly user-related strategies.

For a summary of the terminology used in defining the performance of the network or the services and applications, see Table 1.

C. KQI ESTIMATION

To establish objective scales, the use of service-specific KQIs is being standardized for some organizations, consortia, and standardization bodies around the world. In the context of 5G and B5G networks, KQI estimation is considered a potential strategy to objectively manage networks from a user-centric perspective. In the state of the art, several services have been used as study cases such as traditional video streaming, FTP (File Transfer Protocol), Web-Browsing, and so on. This approach is suitable for managing correctly 5G and B5G networks to guarantee proper quality service levels. Moreover, it is useful for supporting the correct resource management in an automated perspective using only network information that is well-known and reachable to the operators.

In the context of KQI estimation, [32] describes their work as a methodology to meet the service performance through the use of KPIs that depict the network performance and behavior. With this approach, the network operator can

TABLE 1. Performance and quality-related relevant concepts.

Scope	Concept	Acronym	Assessment type	Description
Network	***Key Performance Indicators***	*KPI*	Objective	Defines the network performance. Can be derived from counter and network statistics.
	Quality of Service	*QoS*		Defines the overall level of the network performance. Can be derived from network KPIs.
E2E	***Key Quality Indicators***	*KQI*	Objective	Defines the service performance from a user's perspective.
	Mean Opinion Score	*MOS*	Subjective Objective	Subjectively provided by user assessment (may be affected by external factors). Objectively, it can be estimated through parametric models.
	Quality of Experience	*QoE*		Degree of delight or annoyance of the user that consumes an application or service.

estimate an objective perspective of the user's experience without the need to trespass the level of intrusiveness as other methodologies do, for instance, packet inspection. This work offered a mechanism to estimate KQIs for FTP service. Similarly, in [33] an approach is proposed to estimate KQIs in a network-slicing scenario for a video streaming service. The metrics are estimated from network information and statistics. This approach is useful in the context of new-generation networks, where the operators need to know the quality perceived by the user but also use this information to estimate possible resources required and their pricing.

Conversely, [34] describes a different approach for KQI estimations for HAS (HTTP Adaptive-video Streaming). This strategy infers stalling, resolution, and throughput based on mechanisms that use estimation and classification techniques. The key contribution of this work is the use of pure network metrics such as packet-level statistics. However, its application is limited to the protocols and data patterns for HAS. Furthermore, [35] proffers a KPI-driven KQI mapping based on qualitative levels. The authors present an Adaptive Naive Bayesian Classifier and compare it with KNN (K-Near Neighbors) and Gaussian Kernel Function, to establish the state (ranging from unacceptable to excellent) of KQIs for video, IM (Instant Messaging), and Web services. The results are assessed through accuracy and various specificity metrics.

The future trend for mobile networks is to provide a vertical-friendly NaaS to deliver services. To achieve this goal, it is necessary to establish Service Level Agreements (SLAs) to meet high-quality services under certain pre-established conditions. This is where KQI plays a fundamental role. In [36] a strategy for E2E slicing for 5G using deep learning is presented. Resource provisioning depends on the level of compliance within an SLA, where the FoMs are KPIs. A similar approach is defined in [37] with a framework intended to provide E2E vertical services. Although both cases analyze E2E from a network-centric point of view to meet SLA requirements, these strategies are not consistent with the user-centric vision of 5G and B5G to ensure not only quality of service in terms of network and service provider performance but also end-user satisfaction.

In summary, the current bibliography suggests that there is a wide range of research on QoE using MOS-based strategies. However, these strategies do not reflect E2E quality from a user-centric objective perspective but rather a network-performance one. Additionally, there have been no previous proposals or work focused on the objective estimation of XR KQIs based on network data in the context of 5G and B5G mobile networks. This statement is relevant because XR is a popular subject for the latest generation of networks. Similarly, there is a gap in research on actual implementations of these types of services using commercial mobile infrastructure.

Consequently, the present work's primary contribution is the application of ML as a technology enabler in the context of mobile networks to determine the quality of extended reality services using network data in the context of end users. Given that mobile networks are potential technologies that will meet the requirements of XR, time-sensitive approaches must be employed. In order to achieve this objective, an ML framework that infers the optimal models for estimating XR KQIs based on error and time performance is presented. To comply with time-sensitive approaches, a novel metric is introduced which allows for the objective establishment of the most adequate model that addresses both factors.

III. FRAMEWORK

Measuring or acquiring KQIs is challenging because obtaining them is not trivial. Along the same lines, user privacy arises as a concern, since access to user terminals is required. In this context, this section presents an innovative framework for estimating KQIs of services from network-accessible information and statistics.

In particular, the framework consists of a software pipeline designed to foster several stages or procedures in an organized manner. This methodology is assumed to ensure all the processes are done in the right order (e.g., transformations, training, and posterior assessment) but also to guarantee the objectiveness of the training phase by removing possible statistical leaking of the test data to the training subset. Therefore, the proposed framework aims to leverage reliable KQI prediction by inferring the best-performing algorithms. The optimal model is identified through a multi-step process that integrates feature engineering techniques, hyperparameter tuning, and performance/prediction time evaluation, utilizing the novel and proposed PET_{score}. The general architecture of this proposal is presented in Figure 1.

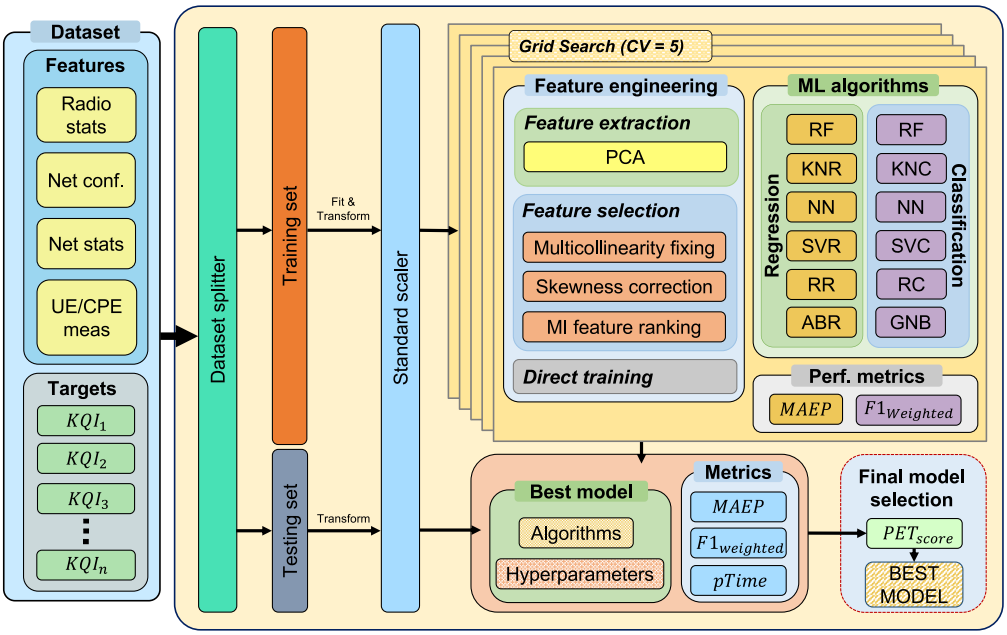

FIGURE 1. ML framework.

The following subsections provide a comprehensive description of each of the stages of the framework.

A. DATA PREPROCESSING

Data preprocessing is performed before the pipeline to ensure data consistency. The dataset undergoes a two-step preprocessing phase to prepare the training data. The first step aims to remove samples with measurement errors or experiments that experienced issues during evaluation, such as disconnection with the radio cell, electrical or processing outages, etc. The later step consists of deleting all the parameters or features whose variance is zero. This means the features whose means have no variation in every experiment executed. These variables generally feature textual information or network or client configurations that remain constant throughout the data collection.

Prior to the training phase, the data set is divided into a training set and a test set using a 70%/30% strategy. Each data set consists of the input features and the targets or KQIs. Then, the features of the training and test sets are standardized so that their scales are modified to be used appropriately in training the algorithms. The standardization consists of subtracting the mean of each metric and dividing the values by its standard deviation, as seen in Equation (1). This proceeding outputs features whose values are scaled and range from –1 to 1. For example, if the unscaled value of a feature is close to the mean, its standardized value will be close to zero. After this process, the standardized split datasets are saved in JSON format for future model evaluation and validation.

$$X_{standardized} = \frac{X - u}{\sigma}, \tag{1}$$

where $X_{standardized}$ is the standardized feature, X the original feature, u the mean, and σ the standard deviation.

B. FEATURE ENGINEERING

Once the data has been transformed by the standard scaler, the training process commences with the creation of a pipeline that integrates a feature engineering technique in conjunction with an ML algorithm. Feature engineering techniques are applied to boost the information that can be extracted from the data. To do so, it is necessary to evaluate different strategies to define which kind of feature engineering provides the best performance concerning the nature of each KQI, its variation, complexity, and how much information can be extracted from the features to predict them. To reach this goal the framework has been designed to test three scenarios: (i) estimation with no Feature Engineering techniques (No_FE), (ii) Feature Selection (FS) using a feature importance ranking, and (iii) prediction of KQIs using Feature Extraction (FE) using PCA (Principal Component Analysis).

The first case is the lowest-complexity strategy of estimation. This consists of inputting the standardized dataset into the pipeline, with no extraction or selection stages (i.e., neither PCA nor another feature engineering strategy). This is done to check if no previous data treatment is needed according to the nature of the collected data. Despite this, the next stage concerning the model training process in the pipeline is common for the three scenarios.

The second case involves a Feature Selection methodology featured by a *Mutual Information* (MI) strategy. This methodology allows the algorithm to input only the best features that impact the most in the KQI estimation. To do so, the MI between the features and target KQI is estimated using

the training set. This strategy generates a feature importance ranking based on the information contained by each feature regarding the KQI. Then, this information is passed to a *SelectKBest* object that inputs progressively each feature to the ML model.

The MI information is a metric that measures the degree of dependency of two random variables. From the information theory perspective, the MI explains the quantity of information contained in one variable by observing the other one. Mathematically, the MI is defined as follows:

$$I(x, y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \cdot log\left(\frac{p(x, y)}{p(x) \cdot p(y)}\right), \tag{2}$$

where $I(x, y)$ is the MI of variables x and y, $p(x, y)$ is the joint probability, and $p(x)$ and $p(y)$ are the marginal probabilities of both variables.

To estimate the MI between the variables in the dataset, the methods *mutual_info_regression* or *mutual_info_classification* from the scikit package are used for continuous or discrete target variables, respectively. For both cases, the output is the estimated MI in nat units ($1\ nat = 1/ln(2)\ shannons$).

Before the application of the feature selection strategy, a preprocessing stage is defined to discard the features that present high multicollinearity, and skewness. The goal of this preprocessing step is to dispose of the best features that will minimize bias, so the inferred model performance is comparable with other techniques without the influence of external factors.

On the one side, multicollinearity is a statistical circumstance where some independent variables have a high linear dependency or high correlation between them. If some correlation is present in the input data, it is difficult for the model to explain the influence or effects of a specific feature over it. This can lead to errors or misinterpretation of the MI ranking calculated in the next step.

To eliminate this issue, the Variation Inflation Factor (VIF) is calculated recursively for all the features to delete the one with the highest VIF value. VIF is defined as a measure of the multicollinearity resulting from the estimation of the determination coefficient (R^2) in a multi-variable linear regression problem. The VIF calculation is performed recursively along the features until the remaining ones reach a threshold of 5 or less. The calculation of this metric is done using the *outliers_variance_inflation_factor* function in the *statsmodels* package [38]. The VIF for the variable j is defined as follows:

$$VIF_j = \frac{1}{1 - R_j^2}. \tag{3}$$

On the other side, the resulting features from the VIF filtering process are subject to Quantile transformation to remove the skewness. The skewness is a statistical metric that explains how asymmetric is the probability distribution concerning its mean value. The existence of highly skewed variables in the input dataset can introduce additional bias due to the lack of balance in the data. To solve this situation, the *Quantile Transformation* [39] converts the skew features ($-1 < skew >= 1$), supported by the *pandas.Dataframe.skew* method, into approximated normal-distributed features using their quantiles information. Once both preprocessing techniques are applied to the data, the resulting features are used to train and evaluate the models.

All these mentioned techniques make FS a powerful tool for KQI prediction since it can provide some advantages, such as the reduction of the dimension of the input dataset, the lightening of the ML model training process, and the acceleration of target prediction time, among others. However, it may increase the processing times and complexity of the training stage due to the data preprocessing proceedings.

Concerning the third case, a PCA stage is used for feature extraction. This consists of mathematically transforming and separating the original information into key information components, which are known as Principal Components (PC). The main application of PCA is data dimension reduction, however, it can also be used for synthesizing new non-correlated features. These PCs are a linear combination of the original features that are ordered in function of how much variance they can explain from the input data. Moreover, each component is orthogonal to the other ones, thus, ensuring there is no redundant information.

The application of this methodology generates an output dataset that synthesizes the original information (patterns, statistics, correlation between variables) into new and non-correlated features that feed the models.

To train the models, the pipeline applies a similar approach to FS but not equal. In this sense, different numbers of PCs are inputted into the model. The number of PC components is progressively recalculated until reaching the number of original features minus 1. The PCA data transformation is applied to the training and test set, however, the PCA transformation coefficients are estimated only using the train set.

C. MODEL DEFINITION

To establish the best model for each algorithm an exhaustive search strategy was used to find the best hyperparameters that achieve the best performance with the validation set. For this purpose, the *Grid Search* algorithm with a 5-fold CV strategy [39] is used. The algorithm looks for the best combination of feature engineering techniques (i.e., varying the number of features for FS or PCs for FE), ML algorithms, and model-specific hyperparameters.

This approach is intended to split the whole training set into several k-folds (in this case in $k = 5$ folds) to train the model with certain conditions determined by a group of predefined hyperparameters that are passed to the algorithm. The training process is repeated k times per configuration using groups of $k-1$ folds, while the reserved one determines

TABLE 2. Grid of parameters for ML model optimization.

Algorithm	Hyperparameter	Values
RF	*n_estimators*	[10, 20, ..., 90, 100]
	max_depth	[5, 6, 7, 8, 9, 10]
RR / RC	*alpha*	$[10e^{-5}, ..., 10e^{5}]$
	fit_intercept	[false, true]
SVR / SVC	*kernel*	[poly, rbf, sigmoid]
	degree	[1, 2, 3, 4, 5, 6, 7]
	epsilon	[0.01, 0.1, 0.5, 1.0]
	C	[0.1, 1, 10, 100]
KNR / KNC	*leaf_size*	[10, 20, 30]
	n_neighbors	[2, 4, 6]
	p	[1, 2, 3]
NN	*alpha*	[0.0001, 0.0003, 0.001, 0.003, 0.01]
	hidden_layer_sizes	[(80,), (100,), (80, 80), (100, 100), (80, 80, 80), (100, 100, 100), (200, 200, 200)]
ABR	*n_estimators*	[50, 75, 100, 125, 150]
	learning_rate	[0.0, 0.333, 0.666, 1.0]
RFC	*min_samples_leaf*	[10, 20, ..., 90, 100]
	max_depth	[5, 6, 7, 8, 9, 10]
GNB	*var_smoothing*	[0, 0.5e-9, 1e-9, 1.5e-9, 2e-9]

which hyperparameter configuration performs better in terms of a metric. Therefore, the use of this technique usually leads to model overfitting avoidance, boosting the performance of the models in different scenarios. An overview of this approach is represented in Figure 1.

The algorithms considered in the framework are:

- Random Forest Regressor (RF)
- Ridge Regression (RR)
- Support Vector Machine Regression (SVR)
- K-Neighbors Regressor (KNR)
- Multilayer Neural Networks (Perceptron - NN)
- AdaBoostRegressor (ABR)

For discrete targets, the classifier version of the algorithms is used, except for ADB. In this case, the AdaBoost algorithm is replaced with the Gaussian Naives Bayes Classifier (GNB). The list of tested algorithms for classification problems is the following:

- Random Forest Classifier (RF)
- Ridge Classifier (RC)
- Support Vector Machine Classifier (SVC)
- K-Neighbors Classifier (KNC)
- Multilayer Neural Networks (Perceptron - NN)
- Gaussian Naives Bayes Classifier (GNB)

In light of the abovementioned algorithms, the grid of parameters that are evaluated using this ML framework depends on the type of ML problem, regression, or classification. To ensure algorithm convergence and affordable training times, some values have been selected by trial and error tests. These value ranges have been previously tested individually for each edge to ensure that its value is valid for each algorithm. The aforementioned values are summarized in Table 2.

D. MODEL TUNING

To evaluate the performance of the algorithms throughout the cross-validation grid search of parameters phase it is necessary to dispose of specific metrics that quantify the prediction ability of the model. For this purpose, the selected metrics reflect the degree of error of a model with a certain hyperparameter configuration. In this sense, a lower error can be translated as a better ability to predict the KQIs.

In this aspect, two metrics are considered regarding the nature of the target KQI. For the case of continuous indicators (i.e., regression problems), it is well known that R2 linear dependency hinders the assessment of the regression performance. Similarly, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) only provide information about the variation around the mean, lacking information about the overall trend. Likewise, the Mean Average Percentage Error (MAPE) may lead to erroneous performance interpretations when values are close to zero.

In this scope, the scaled version of the Mean Absolute Error (MAE) is contemplated. This metric, denoted as MAE% or MAEP, is a percentage version of the traditional MAE, computed by dividing its value by the mean of the observed target. In this context, the closer the value is to zero, the better estimation is obtained. This modification allows converting the absolute scope of the MAE into a relative scale (0 – 1), which eases the analysis and comparison between KQIs of different units or scales, for instance, resolution in pixels or latency in milliseconds. The MAEP is defined as follows:

$$MAEP = \frac{\sum_{n=1}^{N} \left(y_{pred_n} - y_{real_n}\right)}{\sum_{n=1}^{N} y_{real_n}}, \tag{4}$$

where N is the number of evaluated samples, y_{pred_n} is the n-th predicted target, and y_{real_n} is the n-th observed target.

When it comes to discrete targets, the F1 score is one of the most preferred metrics to evaluate the feasibility of classification solutions. However, this metric is only recommended for binary classification. Thus, to quantify the performance of multiclass solutions, the weighted F1 score is considered. This metric introduces weights that consider the proportion of each available class. So, the weighted F1 score is calculated as the sum of the weighted independent F1 scores calculated for each class, denoted as follows:

$$F1_{weighted} = \sum_{l=1}^{L} W_l \cdot F1_{score\_l}, \tag{5}$$

where L represents the number of classes, $F1_{score\_l}$ denotes the F1 score of class l, and W_l is the weight assigned to class l.

Consequently, the single-class $F1\_score$ can be computed as:

$$F1_{score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{6}$$

Therefore, these metrics are used in the CV grid search process, aiming to find the best hyperparameter combination for each ML technique. This leads to an exhaustive search procedure that minimizes the degree of the error by combining different model hyperparameters and feature engineering options. As a result, 18 optimized ML models per KQI are obtained, deriving from the combination of 6 ML algorithms and 3 feature engineering strategies.

E. MODEL EVALUATION

The final stage of the framework consists of the model assessment using the testing dataset. This latter allows for quantifying the real ability of the models to generalize their prediction power with new information that differs from the training knowledge. To achieve this, every model is evaluated using the *MAEP* or the $F1_{weighted}$ scores. Furthermore, the mean prediction time is measured per model using an iterative KQI prediction using randomly selected samples from the testing dataset. This is done by accounting for the elapsed time between the test data inputs the model and it predicts the KQI, being computed as below:

$$pTime = \frac{\sum_{m=1}^{M} \left(t_{output_m} - t_{input_m}\right)}{M}, \tag{7}$$

where M corresponds to the number of randomly selected samples for evaluation, $pTime$ is the average prediction time, while t_{output_m} and t_{input_m} represent, respectively, the timestamp of the moment the target m was predicted and the timestamp of the moment the model was executed for target m prediction.

Although both metrics evaluate the model performance based on the capacity of prediction of the KQI, and on how time-efficient the model is, it is still not trivial to define which ML algorithm is better for each KQI. To overcome this situation, this work introduces the *PET_{score}* (Performance Error and prediction Time) as a metric that integrates both variables. The $PET\_\{score\}$ is defined as the weighted harmonic mean of the model performance score and the prediction time. The mathematical description of $PET\_\{score\}$ is displayed in Equation (8).

$$PET\_\{score\} = \frac{\sum_{i=1}^{I} W_i}{\sum_{i=1}^{I} \frac{W_i}{X_i}}, \tag{8}$$

where W_i is the weight of the metric X_i, and I is the number of metrics accounted, in this case $I = 2$ for $pTime$ and the performance metric (P); *MAEP* for regression and $1 - F1_{weighted}$ for classification problems.

Note that the sum of the weights must be unitary:

$$\sum_{i=1}^{I} W_i = 1. \tag{9}$$

For the weighted $PET\_\{score\}$, when $W_{pTime} \neq W_P$ the definition is the following:

$$PET\_\{score\} = \frac{pTime \cdot P}{W_{pTime} \cdot (P - pTime) + pTime}, \tag{10}$$

where P is the performance metric value, W_{pTime} is the weight assigned to the prediction time, and W_P is the weight of the performance metric. Note that the latter is the complement of the unit (i.e., according to Equation (9)).

For the unweighted $PET\_\{score\}$, when $W_{pTime} = W_P$ the definition is as follows:

$$PET\_\{score\} = 2 \cdot \frac{pTime \cdot P}{P + pTime}, \tag{11}$$

According to the aforementioned definition of $PET\_\{score\}$, the best model performance corresponds to the least $PET\_\{score\}$ for all the situations. In regression problems, this metric rewards less error and prediction time. In classification ones, this metric rewards higher accuracy and less prediction time. The use of weights in this metric is a strong feature that enables the ability to select which characteristic should be highlighted in a model: error performance or time efficiency. This can be quite useful for instance in cases where a minimum error can be tolerated but time sensibility is mandatory to make decisions (e.g., KQI-based resource allocation in new-generation mobile networks).

To summarize, this section presented the proposed ML-based framework for KQI estimation in XR services. To that end, an ML pipeline for training and validation was described as a joint strategy that allows the framework to establish the best hyperparameter tuning that provides the ML model with the highest performance. Different feature engineering approaches are considered, such as FE, FS, and No_FE. Furthermore, the framework enables the selection of models for diverse 360-Video KQIs using regression or classification approaches. To select the best parameters, the MAEP and the weighted F1 score are used to quantify the performance.

In the testing stage, the error performance is estimated for each one of the best hyperparameter-tuned ML models using the mentioned metrics. The prediction time is also measured to define how time-efficient each algorithm is. However, since selecting the best model accounting performance and time efficiency is not a trivial problem, the PET_{score} was defined as an integral assessment mechanism for the selection of the ML models. This metric looks for a trade-off point that objectively establishes the best model performance/prediction time per each KQI.

IV. CASE STUDY: 360-VIDEO

Even though this work focuses on the ML framework, data generation and collection are important to ensure that the inferred ML models can capture the information and characterize the service performance. To test the validity and potential of the framework presented in Section III, the 360-video service is selected as an XR use case.

This section first provides an overview of 360-video and then describes the procedure for collecting the dataset that will be used for evaluation in Section V. This data includes information from multiple sources, such as the service KQIs, radio measurements, and statistics and configuration parameters from the network that were previously obtained using a testbed to generate multiple and iterative tests under different channel conditions. The detailed description of this previous study can be found in [40].

To get an adequate overview of the overall scheme, the following subsections provide a summary of the testbed used for the data generation and collection.

A. 360-DEGREE VIDEO

360-degree video, or 360-video in short, is an XR service that provides an immersive experience through omnidirectional multimedia content. It belongs to the VR category since all displayed content is virtually generated. The interaction with the media is controlled by intuitive human-based actions, thus, the user can feel inside the video itself. To deliver this content a Head-Mounted Device (HMD) should be used. Nonetheless, various traditional video providers like YouTube are presenting alternatives to enjoy 360-video on their platforms [41] using not only HMDs but also computers, tablets, etc.

Concerning the standards, 360-video is a service that belongs to the weak-interaction cloud VR service according to the ETSI (European Telecommunications Standards Institute) in quality evaluation standard F5G-015 [42] released in 2023. This service cannot be analyzed as a traditional service because the content presentation in a VR device differs from the well-known 2D-screen video even using 3D-ready platforms (e.g., YoutubeVR). In this setting, the requirements for 360-video VR are different from the Web-based alternative.

Previous studies have demonstrated that to reach a real-feel 360-video experience, the content should be provided with a minimum resolution of 60 pixels per degree at a recommended 120-Hz frame rate [43]. Moreover, the influence of the startup time of the video and the quantity of stalling events can decimate the QoE. To overcome all these barriers, mobile technologies like 5G and Beyond-5G (B5G) are being standardized and developed using different architectural concepts in comparison with LTE or other legacy networks.

Along these lines, the standardization bodies are trying to find a consensus on which indicators are appropriate to quantify service quality. For instance, the ETSI points the *initial buffering duration - IBD* and the *Average percent of frames freezing* as the relevant metrics for 360-video VR streaming. In this sense, this work will consider those metrics naming them as *Initial startup time* and *Stalling time*. In addition to the latter standard recommendation, this work will include a set of additional metrics that impact the quality of the service based on the recommendation of 3GPP TS 26.247 [31]. According to these facts, the selected 360-Video KQIs for this studio are the following:

- **Initial startup time:** The initial startup time is the period between the events when the client requests the manifest to the server, processes it to solicit the media, the media is loaded into the buffer and starts the playback on the user's screen. To estimate this value a difference in the timestamps is considered. To that end, the timestamp when the manifest is requested to the server is saved, then when the player *isPlaying* flag is set to $isPlaying = true$. This metric is only measured once per session and its unit is seconds.
- **Stalling time:** The amount of time when the client is not playing the media due to an event of rebuffering, or disconnection. This metric is calculated when the player flags switch to the states $isBuffering = true$ and $isStalled = true$ until these values are set back to false. The flags are checked for each HMDS's frame update event (i.e., $HMD_{fr} = 72Hz$ for Meta Quest HMD). The unit of measurement is seconds and the value is accumulative until the end of a video session.
- **Video resolution:** The number of pixels in a video in both dimensions: vertical and horizontal named as height and width. The resolution corresponds to the media displayed to the user and not the physical resolution of the screen. The resolution is fetched once per second from the metadata of the buffered segment and its unit is pixels. This indicator is a discrete variable since its values are fixed by the media server.
- **Video frame rate:** The number of media frames that are displayed to the user. This value is different from the screen frame rate which represents the number of frames updated per second the HMD can do. This metric is measured in frames per second or fps. The video frame rate is fetched one per second from the video player in the HMD.
- **Throughput:** The mean quantity of data transmitted in the downlink channel. This is quantified in the HMD's network interface, starting from the Android layer. The measurement is conducted on a one-second interval, with the number of bytes in the preceding measurement window being calculated. The unit of measurement utilized in this metric is kilobits per second (kbps). It should be noted that this metric considers the end-to-end throughput, which is the total amount of data transferred between the HMD and the multimedia server. This makes it a valuable service-level indicator. Furthermore, it is important to note that current HMDs are not 5G-ready. Consequently, the use of a CPE as a gateway may influence the perception of the service.
- **Latency:** The average end-to-end service-level delay is defined as the time interval between the HMD requests media to the server and the server response arriving at the client. This metric represents the latency introduced by the network that connects the client and server.

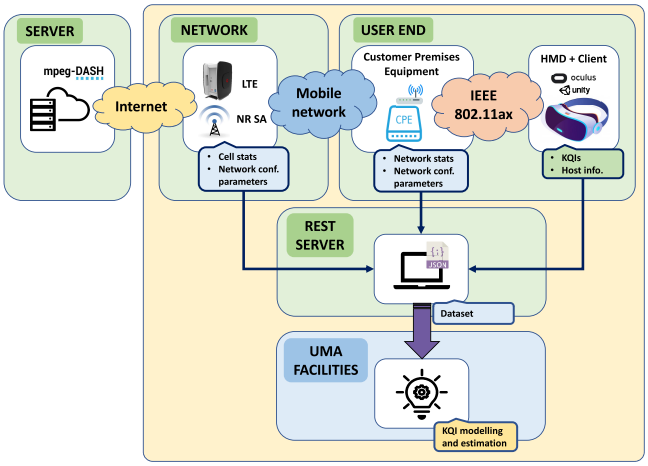

FIGURE 2. Implementation scheme.

This value does not include latencies introduced by processing and graphics tasks at the server or client. Instead, it considers only the server response time and CPE-related delays. This metric is quantified in milliseconds (ms) and is averaged over a one-second window.

- **Buffer health:** The buffer health is a measurement of the content available in the client to be displayed on the screen. This metric is estimated by subtracting the timestamp of the last buffered frame on a media segment minus the timestamp of the first available frame in the buffer. The buffer health is represented in seconds, and its value is updated every second.

B. DATA GENERATION

Considering the architecture of the service, shown in Figure 2, the client side integrates the VR HMD and a CPE (Customer Premises Equipment). The first one is intended to display the 360-video content to the user and to collect KQIs, through a dedicated application developed in Unity 3D. This application allows displaying the content while metrics are being gathered in the background. In addition, the processing and rendering tasks are executed integrally using the HMD's hardware due to the implementation of a standalone architecture. Differently, the CPE is used as a bridge between the mobile network and the WiFi HMD's network interface. Furthermore, some network performance metrics are collected in this device as well as in the transport network.

The transport network is featured by a Network-in-a-box device, which is an open-source solution that mixes SDR platforms with a softwarized network solution, hence, acting like a mobile network infrastructure [44]. In this context, the device facilitates the emulation of some radio impairments such as attenuation and noise presence due to the use of the SDR platform. Additionally from the functionalities mentioned, this solution provides some metrics related to radio performance that are included in the input dataset.

In addition to these elements, a RESTful (Representational State Transfer) server was implemented to serve as the storing point of the measurements done on the client side as well as in the network.

C. DATASET COLLECTION

To acquire the dataset for the training of the ML models, several experiments [40] were done using the testbed depicted in Figure 2. The experiments were intended to display 360-video iteratively, thus assuring that all the tests use the same multimedia content and guarantee objectiveness. The experiments were intended to capture the network influence over the service through different configurations as described in Table 3.

The dataset collection methodology consisted of 12 different configurations of the transport network. Each one is composed of 120-minute-long experiments where samples are obtained for each second of video displayed. Besides, the network-in-a-box provides cellular connectivity as well as generating different channel conditions generated by transmission power changes, channel bandwidth, and noise emulation using the SDR module. Then, the REST server gathers the metrics obtained in the HMD as well as the ones fetched by the network-in-a-box and the CPE, this way generating an integral dataset that represents the service performance from a high-level perspective as well as from a network viewpoint. The interpretation of this process can be seen in Figure 3.

On the network side, some metrics collected by the network-in-a-box and CPE are counters and KPIs of the network and configuration parameters such as channel bandwidth, carrier frequency, throughput, number of retransmissions in uplink as well in the downlink direction, and so forth.

It is important to mention that the collected dataset is composed of a total of 86400 samples. This is the result of the multiplication of the number of radio channel bandwidths (4) by the power transmission scenarios (3), the number of samples per experiment (120-second experiment with a sampling frequency of 1 sample/s), and the number of experiments for each configuration (60). The unprocessed data may be accessed via the IEEE Dataport repository [16]. Furthermore, Table 4 summarizes the features selected for this use case. It is important to note that the referenced dataset contains additional features that are not included in the results. These features do not provide further information for the explainability of the KQIs. The criterion employed is the VIF between inputs. Consequently, inputs that are very correlated are excluded.

V. EVALUATION

In this section, the results obtained through the evaluation of the framework are discussed. The outcomes here described are the performance metrics estimated using different combinations of feature engineering techniques, hyperparameter values, and ML algorithms. The analysis will be extended

TABLE 3. Testbed configuration.

Parameter	Description	Value
Experiment duration	Duration of video playback in seconds	120
Sampling frequency	Number of samples per second	1
Experiments	Number of experiments with the same configuration	60
Technologies	Transport network technologies connecting the user equipment with the Internet	5G
Crowd-BW	LTE Crowdcell channel Bandwidth	5 MHz
		10 MHz
		15 MHz
		20 MHz
MaxPT [1]	Crowdcell maximum power transmission level for *MaxPT* tests	0 dB
MinPT [1]	Crowdcell minimum power transmission level for *MinPT* tests	-10 dB
RedPT [1]	Crowdcell power transmission level for *RedPT* + *Noise* tests	-20 dB
Max-Noise [1]	Maximum noise level for *RedPT* + *Noise* tests	-20 dB
Min-Noise [1]	Minimum noise level for *MaxPT and MinPT* tests	-30 dB
Video resolution	Video available resolutions at the server	720×360
		1080×540
		1440×720
		2160×1080
		3840×1920
Average bitrate per segment	Average bitrate per each video segment (same order as resolutions)	1 Mbps
		1.5 Mbps
		3 Mbps
		5 Mbps
		9 Mbps
Frame rate	Frame rate at which video is encoded	30 FPS
Segment duration	Time period for each video segment	4 seconds
Codec	Video codec used	avc1.42c00d
Initial buffer	Filling time for initial playback	5000 ms
Min. & Max. buffer threshold	Minimum and maximum thresholds for buffer	50000 ms
Streaming protocol	Protocol used for media streaming	Standard DASH
ABR strategy	Adaptive Bitrate strategy used for buffer filling	Throughput-based
HMD Model	HMD model used in the testbed	Oculus Quest 2
Operating System	HMD operating system	Android OS 12L / API-32
Graphic engine	Graphic engine that supports the video 360 client	Unity 2022.3.9f1
Video player	Video player API integrated into the video client	ExoPlayer NonOES

[1] Device-internal configuration parameter.

for each 360-video KQI, thus, it is possible to establish the best-inferred ML model that captures the most information from the input features and outputs a precise prediction.

Concerning the data inputted to the models, a data point (or sample) represents the average value of each feature along a 120-second 360-video session. To do this, the per-second samples collected according to the methodology described in Section IV-C were grouped in experiments of 120 samples (i.e., a session) and averaged according to their nature. This means that the chosen value for a discrete variable is the mode, meanwhile for a continuous one is the average. Even though the use of fewer samples in the training set may affect the prediction accuracy, it can bring some advantages.

One benefit is the reduction in algorithm estimation times and model complexity. This can be considered an enabler for future network management. Currently, networks are not designed to continuously modify their configuration parameters within very short periods, on the order of seconds, for a specific service. This feature is expected to be available for B5G networks. For future optimization implementation, a model trained with a resolution of a few seconds can serve as a useful baseline.

The features considered for the model training and validation tasks correspond to CPE measurements and statistics (i.e., radio quality metrics, traffic metrics), and network-in-a-box [44] radio measurements, statistics, and configuration parameters. The output models are trained, tuned, evaluated, and selected using the framework described in Section III.

The KQIs selected for testing the framework are displayed video resolution, average displayed frame rate, initial startup

TABLE 4. Description of the features.

Source	Feature	Description
Network	UE_UE_dl_bitrate	User DL throughput measured at MAC layer
	UE_pusch_snr	User Signal-to-Noise Ratio in Physical Uplink Shared Channel
	ul_retx	Number of UL retransmitted transport blocks measured at MAC layer
	ul_tx	Number of UL transmitted transport blocks measured at MAC layer
	dl_tx	Number of DL transmitted transport blocks measured at MAC layer
CPE	CPE_PPucch_dBm	Transmission power of the Physical Uplink Control Channel in dBm
	CPE_PPusch_dBm	Transmission power of the Physical Uplink Shared Channel in dBm
	CPE_PPrach_dBm	Transmission power of the Physical Random Access Channel in dBm

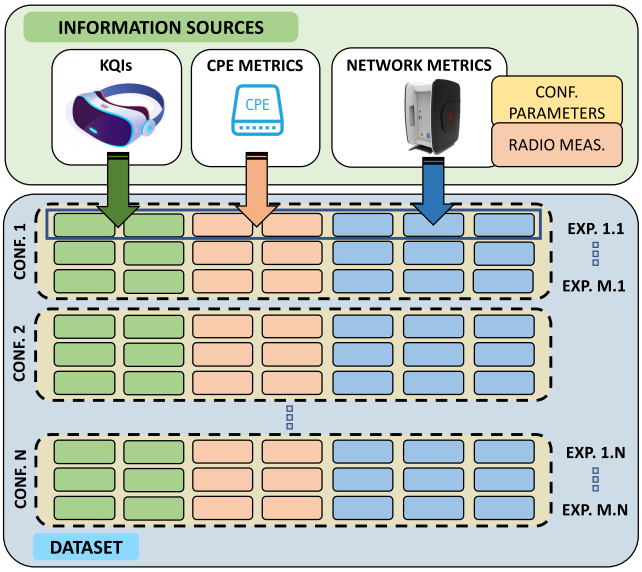

FIGURE 3. Dataset creation.

time, average stalling time, E2E latency, effective throughput at the client device, and buffer health. The models outputted by the framework are intended to introduce the minimum prediction error and, at the same time provide the maximum time efficiency. To achieve this, the PET_{score} estimation was configured to weight both metrics equally. In a time-constraining scenario, the time efficiency may be weighted higher, so little prediction error can be tolerated. The contrary situation may happen by increasing the weight for the prediction error, in scenarios where decision-making is not time-constrained.

On the one side, the hyperparameters summarized in Table 5 correspond to the best-performing model found after the CV Grid search. These values belong to the model with the least prediction error after the training stage per the ML algorithm. Note that some values display N/A (Not available) since some algorithms were used only for regression problems while others were for classification ones. For regression, *MAEP* is the metric used for error performance assessment, while $F1\_weighted$ for classification.

On the other side, the overall best model per KQI is selected through the evaluation of the PET_{score}. To accomplish this, the *pTime* and the error performance metrics (i.e., *MAEP* and $F1\_score$) are estimated using the testing set. The mean prediction time of each model is estimated by averaging the measured prediction time iteratively 100 times using randomly chosen input samples from the testing set.

This section discusses the results focusing on the model performance, prediction time, and the associated PET_{score}. In addition, an analysis of the loss of information each feature provides to the model is presented. This study compares the actual MI between any of the input features with the target KQIs, with respect to the MI between input features and predicted KQIs. This information loss plays a role in identifying when a model cannot capture the information and properly characterize a KQI.

As can be seen in the next subsections, the baseline MI, (i.e., features with measured KQIs) will be represented with a wide dot bar per each feature. Inside it, the MI captured from each ML model (i.e., features with predicted KQIs) is depicted. Sharing these figures, the performance metric shows the progression of the error as a function of the number of features used for training. Additionally, the Figures' terminology used in this section, will be *No_FE* for Non-feature-engineering technique, *FS* for Feature Selection, and *FE* for Feature Extraction. Regarding the metrics, *MAE%* will be employed to depict the *MAEP* on a scale from 0 to 100%.

In the following subsections will be carried out a focused analysis on each 360-video KQI. To ease the understanding of the results, a lower *MAE%* means a lower error on the prediction. Conversely, a higher $F1\_weighted$ implies higher model classification ability and, thus, lower error. To select the best overall model, a lower $PET\_\{score\}$ corresponds to a better model. In this sense, a PET_{score} of 0 describes a perfect prediction ability. For the discussion of the error performance, MAE% values lower than 10% will be considered adequate estimations. Values between 10% and 20% will be established as suitable estimations. Likewise, higher MAE% values until 50% are acceptable, meanwhile, the ones higher than that threshold will be labeled as inappropriate.

A. INITIAL STARTUP TIME

In Figure 4(a) the MAE% reached by each ML algorithm is depicted. As it can be seen, the algorithms' performance

TABLE 5. Algorithm hyperparameters after the training and tuning phase.

Algorithm	Hyperparameter	Key Quality Indicators (KQIs)						
		Initial startup time	*Resolution*	*Frame Rate*	*Stalling time*	*Throughput*	*Latency*	*Buffer health*
RF	*max_depth*	8	N/A	9	8	5	6	9
	n_estimators	30	N/A	50	10	10	30	70
RR / RC	*alpha*	46.41588	3.59381	0.02154	100000.0	9.9999e-06	3.59381	9.9999e-06
	fit_intercept	true	true	true	false	true	true	true
SVR / SVC	*C*	10.0	100.0	100.0	10.0	100.0	100.0	10.0
	degree	4	1	1	1	6	1	1
	epsilon	0.01	N/A	0.01	0.01	0.01	0.01	0.001
	kernel	rbf	rbf	rbf	rbf	poly	rbf	rbf
KNR / KNC	*leaf_size*	20	20	10	10	10	10	10
	n_neighbors	2	7	2	2	4	2	2
	p	2	1	1	1	1	1	1
NN	*alpha*	0.01	0.0001	0.0003	0.003	0.0003	0.01	0.0003
	hidden_layer_sizes	[80, 80, 80]	[80, 80, 80]	[100, 100, 100]	[200, 200, 200]	[100, 100, 100]	[100]	[80, 80, 80]
ABR	*learning_rate*	0.666	N/A	0.333	0.333	0.333	0.333	0.333
	n_estimators	50	N/A	50	50	50	50	50
RFC	*max_depth*	N/A	5	N/A	N/A	N/A	N/A	N/A
	min_samples_leaf	N/A	10	N/A	N/A	N/A	N/A	N/A
GNB	*var_smoothing*	N/A	5e-10	N/A	N/A	N/A	N/A	N/A

is suitable with a special mention for RF with no feature engineering techniques with a mean error of 18.03%. This means that if the average value of the initial startup is 1 second, its prediction generates an output of $\pm 0.18s$.

In terms of the time the model takes to estimate a KQI, depicted in Figure 4(b), the least value belongs to the Ridge Regression model with 0.84 ms. In this context, this is a remarkable value that can leverage the use of ML as a means for network management in real-time decision-making. An analog performance is shown by most of the algorithms except for ABR.

Nevertheless, both metrics were analyzed independently, it is not objective to define which algorithm performs the best. For instance, RF with *No_FE* can characterize this KQI with a tolerable degree of error, however, the time it takes to predict is approximately double that of the second best-performing algorithm, SVR. This is not a trivial issue in mobile networks, or in time-sensitive applications like XR services where decisions should be taken in the scale of milliseconds. This situation demonstrates the robustness of this framework leveraged by the use of the PET_{score} in finding the adequate trade-off between prediction ability and time efficiency.

As illustrated in Figure 4(c), the algorithm that most closely aligns with this balanced approach is SVR using FS, with a PET_{score} of 0.13. To complement this scrutiny, Figure 4(d) shows the feature that contributes the most information to the model is the bitrate measured at network-in-a-box level (*UE_UE_dl_bitrate*). For the selected model, SVR, the error decreases while increasing the number of features used for training. It is remarkable that the framework not only indicates that the best model is possible with SVR with an FS approach but also indicates that the best number of features that introduce the least error is 6. In these lines, the proposed framework searches the best-performing algorithm in terms of error and time efficiency as well as model complexity.

B. VIDEO RESOLUTION

One of the most important 360-video service quality indicators is the video resolution. This metric provides salient insights into the experience of the user. A poor resolution can severely affect the sensation of immersion and/or introduce unnecessary uncomfortable feelings. To predict the visualized resolution from the client's perspective the framework infers this variable as categorical, since only defined resolutions by the server are available for delivering.

For the abovementioned approach, the performance in this case is measured in terms of the *F1_weighted*. As depicted in Figure 5(a), all the algorithms perform remarkably well in terms of KQI prediction. The best results are obtained with the RF with No_FE approach, the other models in this category perform similarly though. Note that in most cases the classification ability of the models is flawless, except for the Ridge Classifier (i.e., RC) algorithm with FS.

Likewise, Figure 5(b) shows the prediction time accounted for every algorithm. The results demonstrate a similar pattern concerning the prediction time with values in the scale of the millisecond, disregarding RF and KNC.

In terms of the PET_{score}, the framework suggests in Figure 6(c) that the best model is RC using the No_FE strategy with a value of approximately 0. This indicates a very high classification capacity. In this context, as a

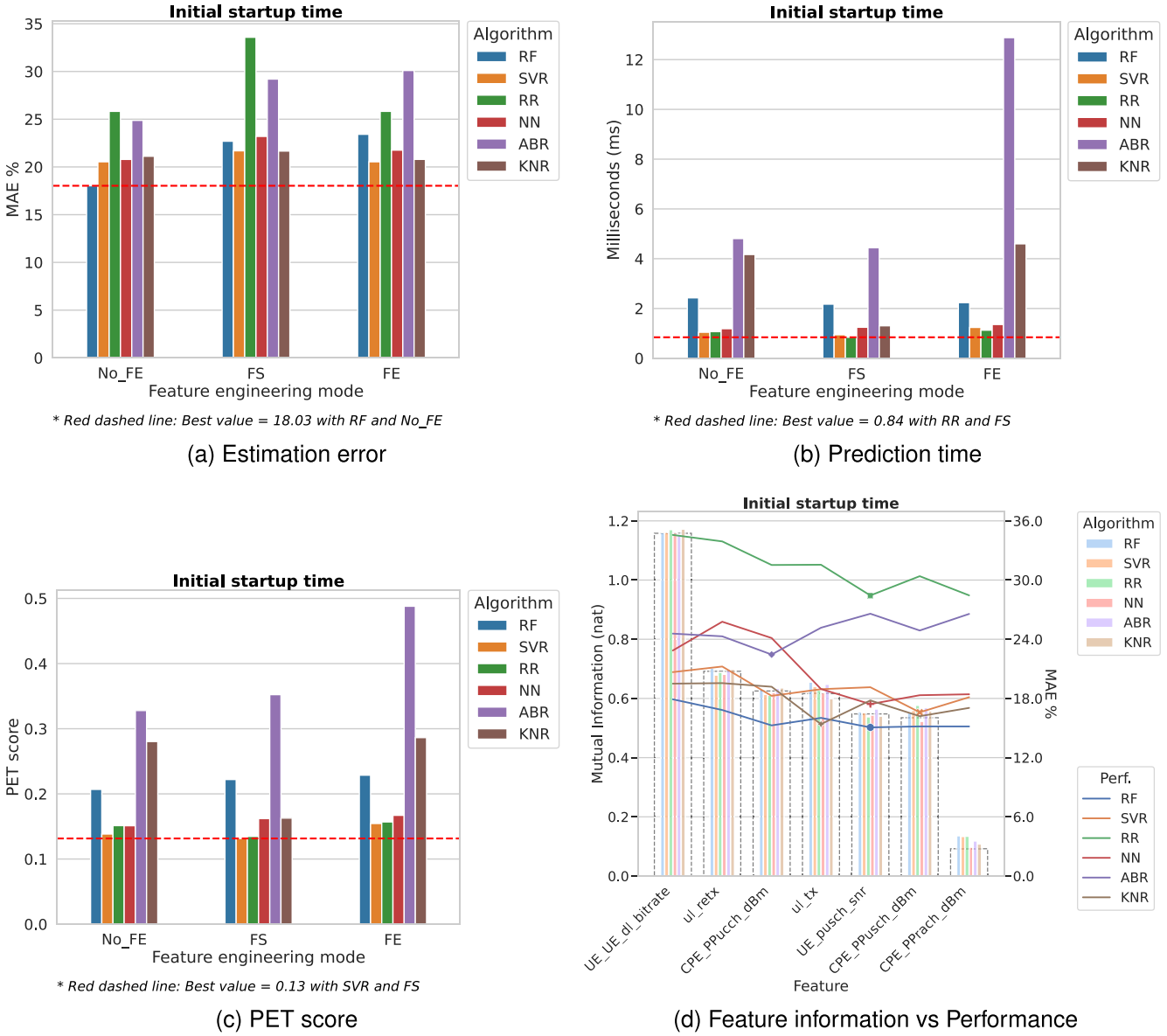

FIGURE 4. ML model performance for Initial startup time.

side-analysis, the estimation of the video resolution can develop an improvement in the estimation of other metrics correlated with its implicit information. For instance, a higher resolution implies more transport resources, which may lead to an increment in the probability of suffering from stalls in the playback or a rise in the initial playback time values.

As well as with the prior KQI, the MI loss is negligible, which refers to the outstanding performance of the ML models with this target. It is important to observe that an exceptional F1 score is obtained with the information of only the first feature. The addition of new variables increases progressively the accuracy but on a minor scale, however, the framework identifies that adding the entire set of features obtains the best results.

C. VIDEO FRAME RATE

Another KQI analyzed is the video frame rate displayed on the user side through the HMD. This parameter depends on the number of downloaded frames rather than on the hardware capacity of the device. As expected, the assessment has turned out in low MAE for most of the ML techniques and approaches used (i.e., No_FE, FS, FE). The best model shows an average error of 1.42% with RF and No_FE. This implies, that for a 60 FPS 360-video service, the effective frame rate perceived by the user (considering frame losses and stalls) and estimated by the model can fail in about less than a frame.

Now, according to the prediction time, the best results are obtained with SVR and FS but with similar performance to RF in terms of error. To find the best combination,

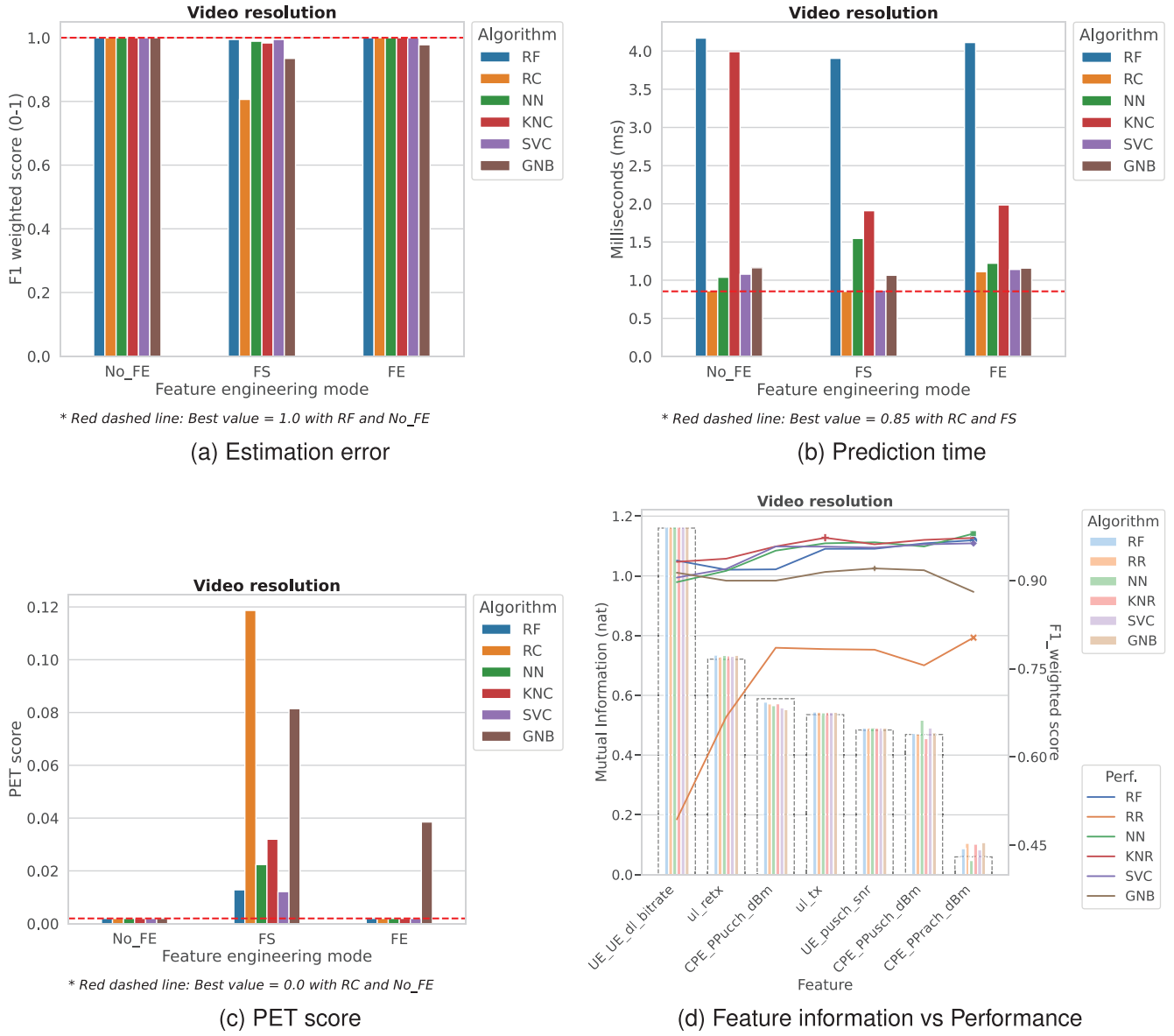

FIGURE 5. ML model performance for Video resolution.

the PET_{score} grades RF as the preferred choice since it rewards the best estimation, among other very low error options. To improve time efficiency, the PET_{score} weights should prioritize prediction time.

Moving to the MI analysis in Figure 6(d) depicts a similar behavior to the past two KQIs. Random forest captures a good amount of information from the first feature, yet the incorporation of new features still improves the overall performance.

D. STALLING TIME

Contrary to the previous cases, the average stalling time is a difficult indicator to predict. Its values depend on direct factors such as the video resolution, frame rate, and buffer health as well as external ones, such as the current network conditions, and the radio ones. Besides, current streaming protocols add functionalities like Adaptive Bitrate (not the ABR term used for ML) that prevent the playback from stalling by changing the resolution of the delivered video segments.

The effects of this dependency are reflected in the prediction error shown in Figure 7(a). As displayed, the MAE% exhibits elevated inaccuracy of certain algorithms with special mention to RR and ABR, which have been outperformed by all the other algorithms. In this setting, the best model uses ABR with the No_FE approach showing an average error d 41.92%. For instance, taking account into the mean value of stall time in a session is 17 ms, the prediction will fail at about 7 ms. Nonetheless, it is valuable to get an insight into this metric which is helpful for future network optimization.

Following this discussion, the average stall time's prediction time provides a comparable outlook with the previous KQIs. The least pTime is achieved with RR jointly

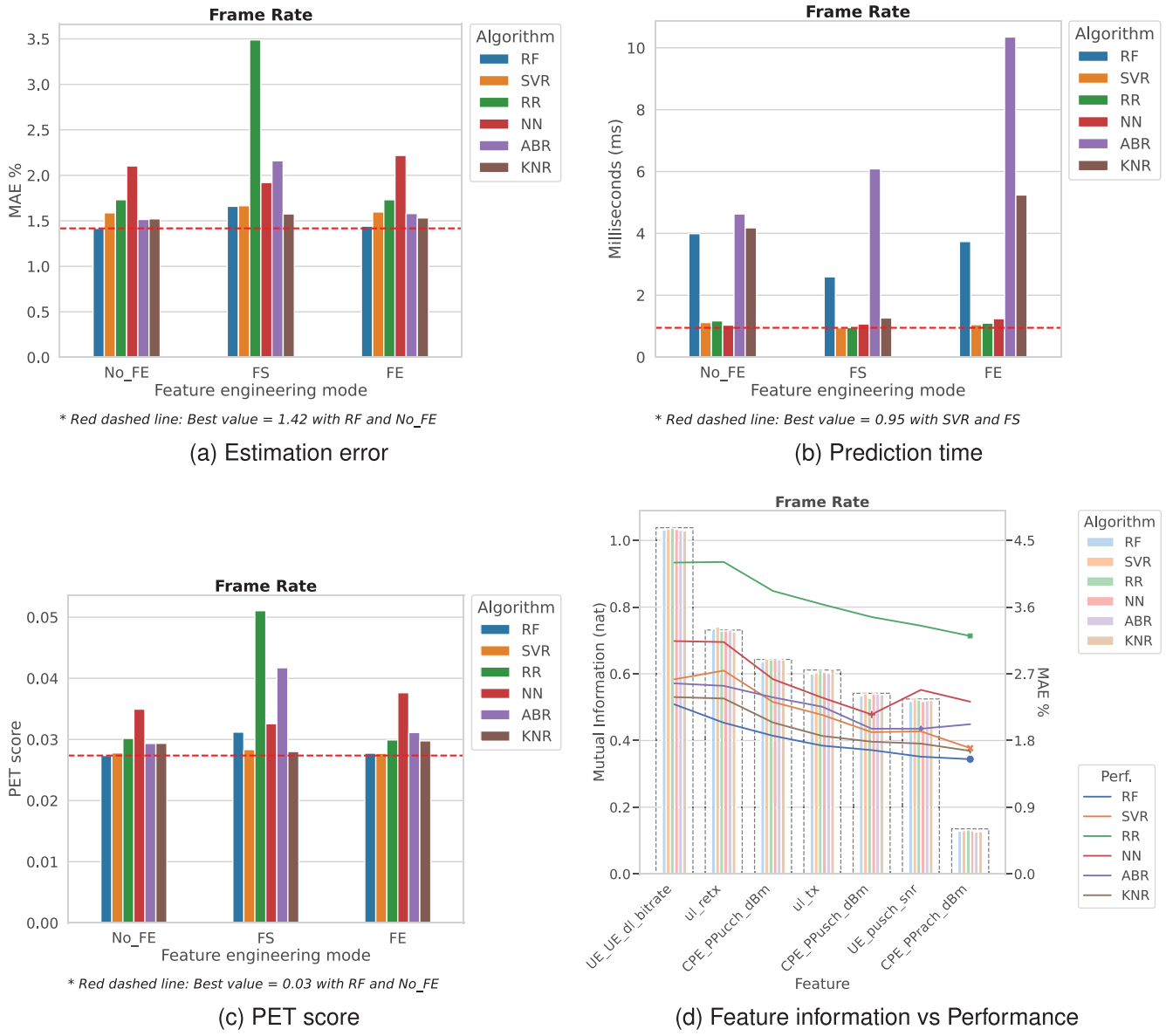

FIGURE 6. ML model performance for Video frame rate.

with FS in about 1 ms. Disregarding these specific metrics, the framework outputs SVR with FS as the most balanced model. This outcome can be considered an error from the framework, yet all the models are not able to properly capture the information from the feature to infer this KQI.

Figure 7(d) shows that the model's loss of MI is relatively high compared to previous KQIs. This lack of information causes a significant output error, despite adequate prediction time. To address this issue, the PET_{score} should be adjusted by reducing the weight of prediction time, allowing the framework to prioritize error performance.

E. THROUGHPUT

When it comes to the throughput, the estimation of this metric on the client side is an important indicator that describes a quality-of-the-service viewport based on the approximate quantity of information that arrives at the device. In this context, this parameter may directly affect the other parameters involved in this work. A constrained throughput can carry to a low-resolution video service, or lead to the increase of stalling events or the startup time of this service.

Beyond this fact, this metric is first measured and now predicted from the user's point of view. This means that even for the network side, the DL throughput metric considers packet sent, retransmissions, control plane information, and additional information, meanwhile from the user side, the throughput indicates the effective data arriving at the headset.

The results in Figure 8(a) report that the KNR algorithm with No_FE performs the best among the others with an MAE% of 5.5%. However, it is remarkable that all the other algorithms, except for SVR, perform very acceptable (less

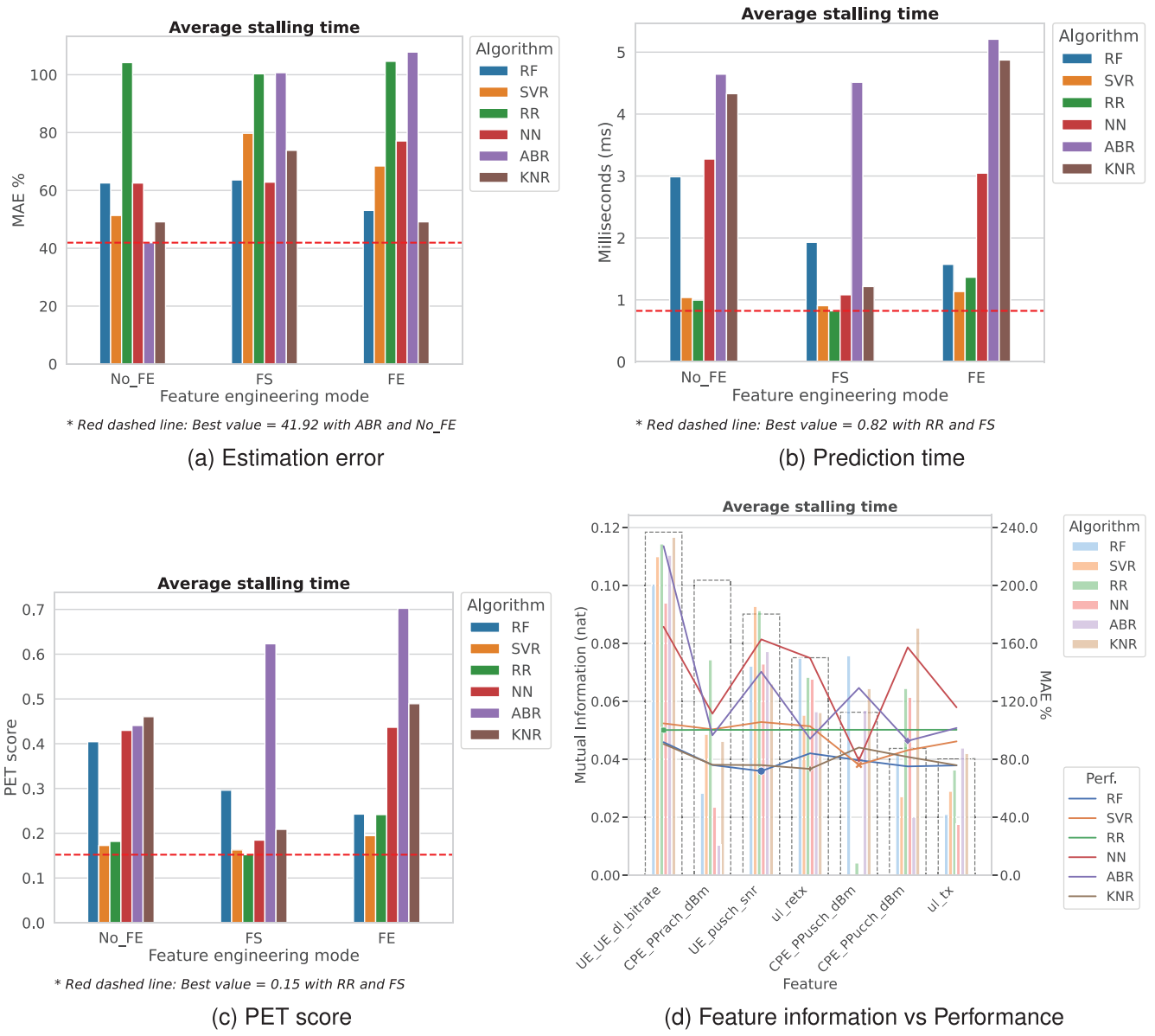

FIGURE 7. ML model performance for Stalling time.

than 10%). When it comes to the estimation time, the results show the same pattern exhibited for the other estimated metrics.

Concerning the PET_{score}, the selected model uses KNR with FS finding an adequate balance between the error performance and prediction time. Note that in cases where the model captures well the information from the features, an equally balanced PET_{score} is enough to choose a suitable model. This can be supported by the MI vs performance comparison displayed in Figure 8(d).

F. LATENCY

The E2E latency considers the bidirectional delay between the HMD and the video server. As shown in Figure 9(a), the results obtained show good performance for most of the algorithms. RF with No_FE displays the best approximation (error of 6.5%) compared with the ground-truth values in the dataset. This is a remarkable value due to the difficulty of estimating a real E2E latency from a service perspective.

Regarding the estimation time, the best-performing algorithm is RR using FS, however, this algorithm does not present a good estimation of the indicator. To get the better of this situation, the PET_{score} (see Figure 9(c)) establishes that the best combination algorithm/feature engineering technique is SVR with FE, which outputs an adequate degree of error. Along the same lines, it is possible to observe in Figure 9(d) that a good level of error translates to a good capacity of the model to capture the information from the input features alike in prior cases.

Beyond the numbers, the knowledge of this metric can provide key insights into other KQIs like stalling events, initial startup time, etc. In this context, the latency can

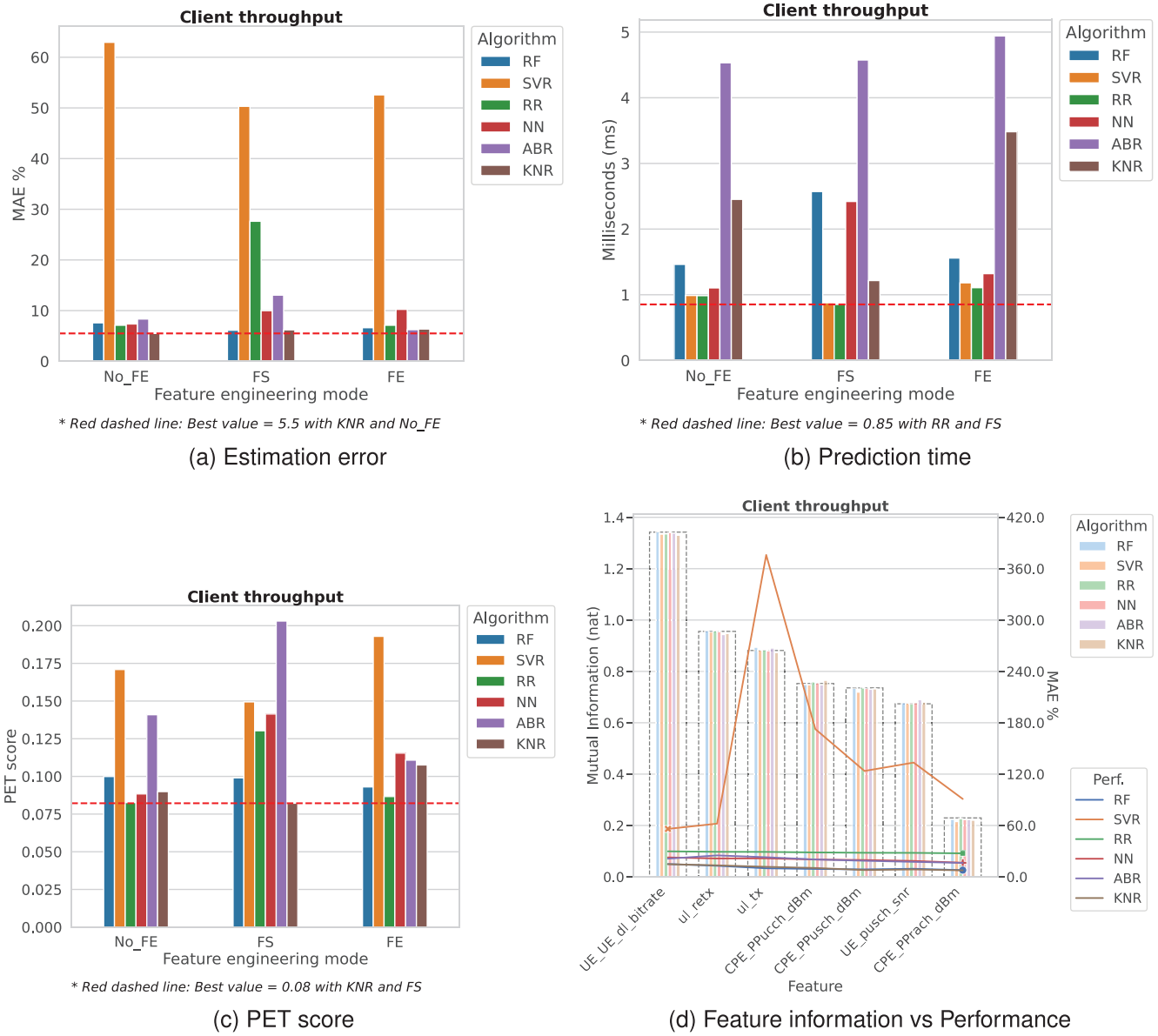

FIGURE 8. ML model performance for Throughput.

give an adequate perception of the level of the network stress which produces effects on the service performance and experience.

G. BUFFER HEALTH

To finalize the analysis of 360-video KQIs prediction, the results for buffer health PET_{score}, estimation error, and prediction time are displayed in Figures 10(c), 10(a) and 10(b), respectively. The best overall algorithm is SVR with the FE approach, although the best-performing model is RF in terms of error. Concerning prediction time, the best combination is RR with the FS technique.

Repeating the behavior of the aforementioned KQI analysis, the good performance owes to the good ability of the models to represent the buffer health using the features inputted for training.

H. INSIGHTS AND SUMMARY

The latter subsections presented the evaluation of the framework using different KQIs from the 360-video service. This assessment was featured by metrics that characterize the error performance and prediction time. To infer the best combination of ML algorithm jointly with a feature engineering technique, the balanced PET_{score} was employed. The results suggest that using equally weighted PET_{score} components is a suitable solution for models that capture enough information from the feature. For scenarios where the error is higher than expected, adjusting the weights to prioritize the error should be considered.

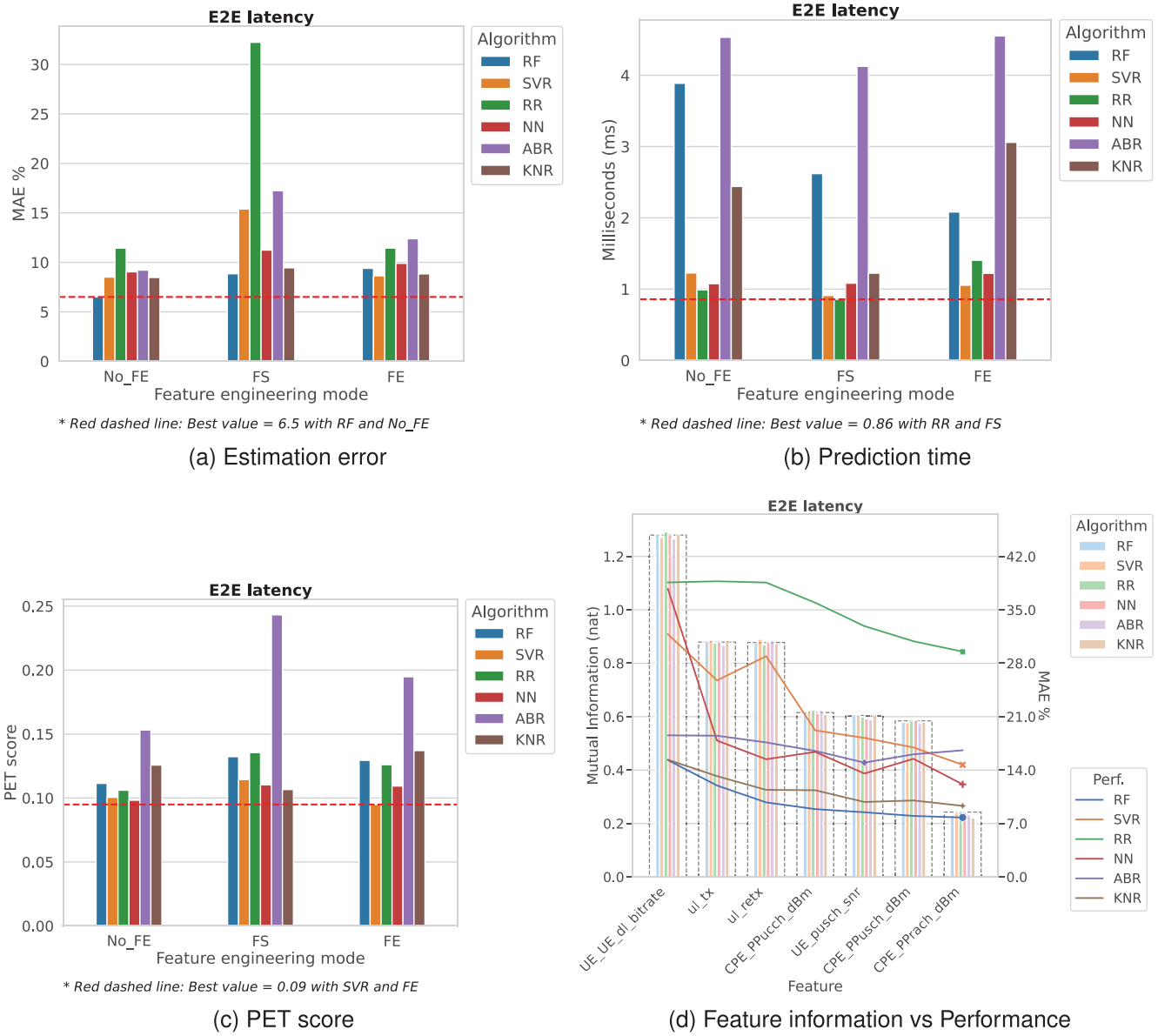

FIGURE 9. ML model performance for Latency.

Table 6 summarizes the best overall models selected by the minimum PET_{score}. For model hyperparameters refer to Table 5.

On the one hand, the results leverage the potential of this proposal to infer and determine the best ML solution for predicting KQIs. In this context, the methodology can be extended to any XR service since it only requires configuring the ML algorithms and feature engineering techniques to be assessed. From a wide point of view, the No_FE strategy reduces error by delivering more information to the models. Conversely, the FS approach improves the time efficiency, yet the limitation on the number of features may incur bias addition. FE is a good alternative since the PCA transformation captures the information from several sources in its components, nevertheless, these processes increase the complexity of the model, translating to higher prediction times.

On the other hand, to better understand why certain KQIs are more difficult to estimate concerning others, an MI matrix showing the shared information between KQIs is displayed in Figure 11. The MI calculations are performed using the approach described in the framework. The values correspond to the quantity of information each KQI has concerning others in nat units (natural log). The results show that the throughput can highly impact the initial playing time, the video resolution, the frame rate, and the latency, which is logical as previously explained in the throughput estimation analysis. Conversely, the MI between the stalling time and the other KQIs is almost negligible.

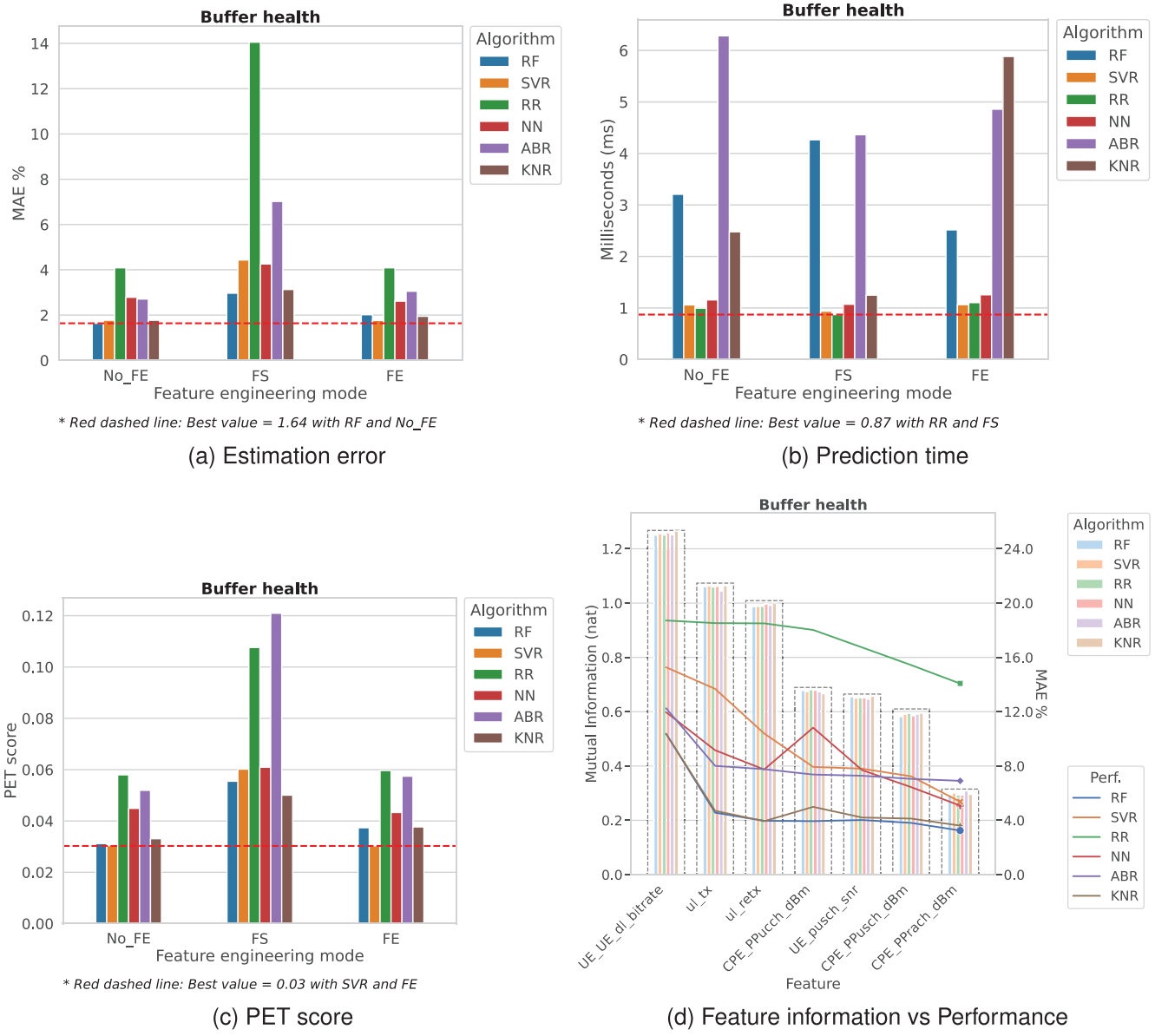

FIGURE 10. ML model performance for Buffer health.

TABLE 6. Best overall models per 360-video service KQIs ranked by PET_{score}.

KQI	Algorithm	Feature engineering	PET score
Initial startup time	SVR	FS	0.13
Video resolution	RC	No_FE	0.00 aprox.
Frame rate	RF	No_FE	0.03
Average stalling time	RR	FS	0.15
Throughput	KNR	FS	0.08
Latency	SVR	FE	0.09
Buffer health	SVR	FE	0.03

This confirms that neither the input features from the dataset nor the KQIs, which are derived from the actual features, can provide more valuable information to improve the prediction performance. To reduce the bias, it may be necessary to synthesize new features or add new information to the dataset, or alternatively, use different ML algorithms.

VI. CONCLUSION

This work has presented an ML framework for KQI estimation of XR services. KQIs are powerful metrics that can exploit the information self-contained in network, such as measurements, statistics, and configuration parameters that are reachable to network operators. The key advantage of this approach is the ability to speed up the process of service management from an E2E perspective, as well as introduce new features that may be used to improve the network performance in terms of the service experience.

This framework aims to automate the process of inferring the best-performing model in terms of error and time

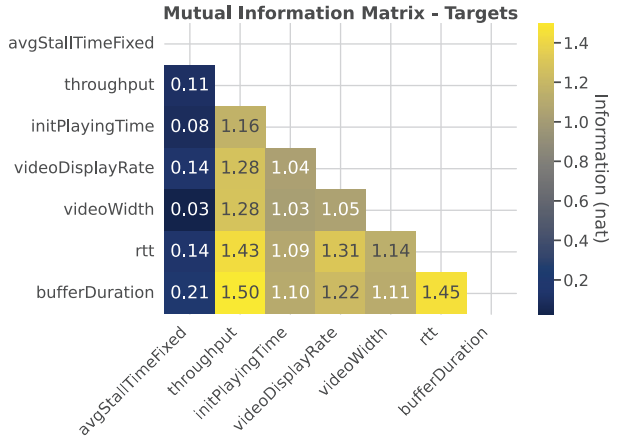

FIGURE 11. Mutual information between KQIs.

efficiency. To that end, this proposal combines an exhaustive grid search of model hyperparameters that minimize the prediction error with different feature engineering techniques, such as feature selection, and feature extraction. Since time is a crucial factor for time-sensitive XR services, it is mandatory to consider this factor in the training and validation of the models.

Taking account into the aforementioned fact, evaluating and selecting the algorithms is not a trivial task. To accomplish this, we introduced the PET_{score} as a mechanism to find a trade-off between performance error and prediction time. Thus, the chosen ML approaches fit properly the error and time performance requirements for each KQI.

To validate our framework, the 360-video service has been selected as an XR use case. In this sense, a dataset collected using an E2E testbed has been used as input for the ML framework. The results show that the selected models for the framework comply with the two objectives, minimizing error and prediction time per each target KQI. As an outcome, ML algorithms and feature engineering combinations are recommended per each 360-video service KQI. In addition, this work has also analyzed the dependency between KQIs using their MI. The results suggest that in some cases alternative ML algorithms or a nested prediction should be explored, as well as modifying the PET_{score} weight to prioritize the error.

In addition, the dataset collected for the use case evaluation has been made publicly available with the intention of promoting and encouraging research in this field.

As a future research line, the impact of nested estimation of the metrics may be performed to enhance the accuracy of the algorithms. The exploitation of previously estimated KQIs may be useful to strengthen the statistical information in the training set, according to the MI matrix, thus reducing the error in the predictions. Moreover, it is planned to work on the implementation of ML-based network configuration mechanisms to improve its performance by exploiting some 5G/B5G enabler technologies such as network slicing, virtualization, MEC, etc. Furthermore, it is possible to explore this strategy oriented to its application on novel network architectures like Open RAN (e.g., x-App and r-App design).

REFERENCES

[1] "Company info and news | Meta." Meta. 2022. [Online]. Available: https://about.facebook.com/

[2] Y. Wang et al., "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 319–352, 1st Quart., 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9880528/

[3] A. M. Gavgani, F. R. Walker, D. M. Hodgson, and E. Nalivaiko, "A comparative study of cybersickness during exposure to virtual reality and "classic" motion sickness: Are they different?" *J. Appl. Physiol.*, vol. 125, no. 6, pp. 1670–1680, Dec. 2018. [Online]. Available: https://journals.physiology.org/doi/full/10.1152/japplphysiol.00338.2018

[4] S. LaMotte, "Virtual reality has some very real health dangers—CNN." Dec. 2017. [Online]. Available: https://edition.cnn.com/2017/12/13/health/virtual-reality-vr-dangers-safety/index.html

[5] E. Chang, H. T. Kim, and B. Yoo, "Virtual reality sickness: A review of causes and measurements," *Int. J. Human Comput. Interact.*, vol. 36, no. 17, pp. 1658–1682, 2020. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/10447318.2020.1778351

[6] P. Costello, "Health and safety issues associated with virtual reality—A review of current literature," Loughborough Univ., Loughborough, U.K., Rep. 37, 1997. [Online]. Available: http://www.agocg.ac.uk/reports/virtual/37/37.pdf

[7] H. H. H. Mahmoud, A. A. Amer, and T. Ismail, "6G: A comprehensive survey on technologies, applications, challenges, and research problems," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, Apr. 2021, Art. no. e4233. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/ett.4233

[8] "O-RAN ALLIANCE | about us," O-RAN ALLIANCE e.V., 2022. [Online]. Available: https://www.o-ran.org/about

[9] T. Zhao, Q. Liu, and C. W. Chen, "QoE in video transmission: A user experience-driven strategy," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 285–302, 1st Quart., 2017.

[10] M. S. Anwar, J. Wang, A. Ullah, W. Khan, S. Ahmad, and Z. Fei, "Measuring quality of experience for 360-degree videos in virtual reality," *Sci. China Inf. Sci.*, vol. 63, no. 10, Oct. 2020, Art. no. 202301. [Online]. Available: https://link.springer.com/article/10.1007/s11432-019-2734-y

[11] S.-H. Yao, C.-L. Fan, and C.-H. Hsu,, "Towards quality-of-experience models for watching 360° videos in head-mounted virtual reality," *Proc. 11th Int. Conf. Qual. Multimedia Exp.*, Jun. 2019, pp. 1–3.

[12] P. Hanhart, Y. He, Y. Ye, J. Boyce, Z. Deng, and L. Xu, "360-degree video quality evaluation," in *Proc. Picture Coding Symp.*, Sep. 2018, pp. 328–332.

[13] H. T. Tran, N. P. Ngoc, C. M. Bui, M. H. Pham, and T. C. Thang, "An evaluation of quality metrics for 360 videos," in *Proc. Int. Conf. Ubiquitous Future Netw.*, Jul. 2017, pp. 7–11.

[14] S. Park, A. Bhattacharya, Z. Yang, M. Dasari, S. R. Das, and D. Samaras, "Advancing user quality of experience in 360-degree video streaming," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, May 2019, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8999460/

[15] R. I. T. D. C. Filho et al., "Dissecting the performance of VR video streaming through the VR-EXP experimentation platform," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 4, pp. 1–23, Nov. 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3360286

[16] S. Peñaherrera, C. Baena, S. Fortes, and R. Barco, May 2024, "E2E Dataset of 360-degree video service using mobile networks," [Online]. Available: https://ieee-dataport.org/documents/e2e-dataset-360-degree-video-service-using-mobile-networks

[17] "3GPP—TSG service and system aspects (SA): Specification # 28.863," 2024. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3960

[18] M. T. Islam, C. E. Rothenberg, and P. H. Gomes, "Predicting XR services QoE with ML: Insights from in-band encrypted QoS features in 360-VR," *Proc. IEEE 9th Int. Conf. Netw. Softwarization Boosting Future Netw. Through Adv. Softwarization*, 2023, pp. 80–88.

[19] T. Begluk, J. B. Husic, and S. Barakovic, "Machine learning-based QoE prediction for video streaming over LTE network," in *Proc. 17th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8345519/

[20] C. E. Rothenberg et al., "Intent-based control loop for DASH video service assurance using ML-based edge QoE estimation," in *Proc. IEEE Conf. Netw. Softwarization Bridging Gap Between AI Netw. Softwarization (NetSoft)*, Jun. 2020, pp. 353–355.

[21] C. Gutterman et al., "Requet: Real-time QoE detection for encrypted YouTube traffic," in *Proc. 10th ACM Multimedia Syst. Conf. (MMSys)*, Jun. 2019, pp. 48–59.

[22] F. Dobrian et al., "Understanding the impact of video quality on user engagement," *Commun. ACM*, vol. 56, no. 3, pp. 91–99, Mar. 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2428556.2428577

[23] A. Ben Letaifa, "Real time ML-based QoE adaptive approach in SDN context for HTTP video services," *Wireless Pers. Commun.*, vol. 103, no. 3, pp. 2633–2656, Sep. 2018. [Online]. Available: https://link.springer.com/article/10.1007/s11277-018-5952-6

[24] J. van der Hooft, S. Petrangeli, M. Claeys, J. Famaey, and F. De Turck, "A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 131–138. [Online]. Available: http://ieeexplore.ieee.org/document/7140285/

[25] H. Z. Jahromi, A. Hines, and D. T. Delanev, "Towards application-aware networking: ML-based end-to-end application KPI/QoE metrics characterization in SDN," in *Proc. Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Aug. 2018, pp. 126–131.

[26] *Influencing Factors on Quality of Experience for Virtual Reality Services*, ITU-T, Geneva, Switzerland, Rec. G.1035, 2021. [Online]. Available: https://www.itu.int/rec/T-REC-G.1035-202111-I

[27] A.-E. M. Taha, "Quality of experience in 6G networks: Outlook and challenges," *J. Sens. Actuator Netw.*, vol. 10, no. 1, p. 11, Feb. 2021. [Online]. Available: https://www.mdpi.com/2224-2708/10/1/11

[28] N. Ahmad, A. Wahab, J. Schormans, and A. A. Arnab, "Significance of cross-correlated QoS configurations for validating the subjective and objective QoE of cloud gaming applications," *Future Internet*, vol. 15, no. 2, p. 64, Feb. 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/2/64

[29] H. Knoche, H. G. De Meer, and D. Kirsh, "Utility curves: Mean opinion scores considered biased," in *Proc. IEEE Int. Workshop Qual. Service (IWQoS)*, 1999, pp. 12–14.

[30] *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport*, Rec. P.1203, ITU-T, Geneva, Switzerland, 2017. [Online]. Available: https://www.itu.int/rec/T-REC-P.1203

[31] "Transparent end-to-end packet-switched streaming service (PSS); progressive download and dynamic adaptive streaming over HTTP (3GP-DASH)," 3GPP, Sophia Antipolis, France, Rep. SP # 26.247, 2023. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1444

[32] A. Herrera-Garcia, S. Fortes, E. Baena, J. Mendoza, C. Baena, and R. Barco, "Modeling of key quality indicators for end-to-end network management: Preparing for 5G," *IEEE Veh. Technol. Mag.*, vol. 14, no. 4, pp. 76–84, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8868102/

[33] C. Baena, S. Fortes, E. Baena, and R. Barco, "Estimation of video streaming KQIs for radio access negotiation in network slicing scenarios," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1304–1307, Jun. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9031300/

[34] S. Wassermann, M. Seufert, P. Casas, L. Gang, and K. Li, "ViCrypt to the rescue: Real-time, machine-learning-driven video-QoE monitoring for encrypted streaming traffic," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2007–2023, Dec. 2020.

[35] K. Zhang, G. Chuai, S. Maimaiti, and Q. Liu, "User experience estimation in multi-service scenario of cellular network," *Sensors*, vol. 22, no. 1, p. 89, Dec. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/22/1/89

[36] H. Chergui and C. Verikoukis, "Offline SLA-constrained deep learning for 5G networks reliable and dynamic end-to-end slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 350–360, Feb. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8931583/

[37] X. Li et al., "5Growth: An end-to-end service platform for automated deployment and management of vertical services over 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 84–90, Mar. 2021.

[38] J. Perktold, S. Seabold, and J. Taylor, "statsmodels.stats.outliers_influence.variance_inflation_factor-statsmodels 0.14.1," 2023. [Online]. Available: https://www.statsmodels.org/stable/generated/statsmodels.stats.outliers_influence.variance_inflation_factor.html

[39] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://scikit-learn.sourceforge.net

[40] O. S. Penaherrera-Pulla, C. Baena, S. Fortes, E. Baena, and R. Barco, "KQI assessment of VR services: A case study on 360-video over 4G and 5G," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 5366–5382, Dec. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9833911/

[41] *YouTube VR—Watch—YouTube VR*. (2022). [Online Video]. Available: https://vr.youtube.com/watch/

[42] "Fifth generation fixed network (F5G); F5G residential services quality evaluation and classification release 2," ETSI, Sophia Antipolis, France, Rep. ETSI GS F5G 015, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/F5G/001_099/015/01.01.01_60/gs_F5G015v010101p.pdf

[43] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.

[44] C. Baena, S. Fortes, O. Penaherrera, and R. Barco, "A framework to boost the potential of network-in-a-box solutions," in *Proc. 12th Int. Conf. Netw. Future (NoF)*, 2021, pp. 1–3.

O. S. PEÑAHERRERA-PULLA (Member, IEEE) received the engineering degree in electronics with a mention in telecommunications from Universidad Politecnica Salesiana, Ecuador, in 2017, and the master's degree in telematics and telecommunication networks from the University of Malaga, Spain, in 2020, where he is currently pursuing the Ph.D. degree focused on the assessment and optimization of mobile networks based on the quality of experience for XR services.

CARLOS BAENA received the M.Sc. degree in telematic engineering from the University of Malaga, Spain, where he is currently pursuing the Ph.D. degree focused on E2E network optimization for video and gaming services. Since 2018, he has been working as a Research Assistant with the Department of Communications Engineering, University of Malaga.

SERGIO FORTES (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Malaga, Spain. He began his career in the field of satellite communications, holding positions in European space agencies, where he participated in various research and consultant activities on broadband and aeronautical satellite communications. In 2012, he joined the University of Malaga, where his research is focused on the application of machine learning techniques for the advanced management of cellular, satellite, and IoT.

RAQUEL BARCO received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Malaga, Spain. In 2000, she joined the University of Malaga, where she is currently a Full Professor. She has worked on projects with major mobile communications operators and vendors and is an author of more than 100 high-impact papers.