Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

UNIVERSAL WISER
PUBLISHER

Article

Efficient Video QoE Prediction in Intelligent O-RAN

Aditya Padmakar Kulkarni[1,*], Navrati Saxena[1] and Abhishek Roy[2]

[1]Department of Computer Science, San Jose State University, San Jose, CA, USA
[2]Communication System Design, Advanced Communication Technology, MediaTek, San Jose, CA, USA
E-mail: adityapadmakar.kulkarni@sjsu.edu

Received: 10 October 2023; **Revised:** 14 December 2023; **Accepted:** 15 December 2023

Abstract: Open Radio Access Network (O-RAN) is a platform developed by a collaboration between wireless operators, infrastructure vendors, and service providers for deploying mobile fronthaul and midhaul networks, built entirely on cloud-native principles. The vision of O-RAN lies in the virtualization of traditional wireless infrastructure components, like Central Units (CU), Radio Units (RU), and Distributed Units (DU). O-RAN decouples the above-mentioned wireless infrastructure components into open-source elements, operating consistently with other elements of different vendors in the network. Quality of Experience (QoE) deals with a user's subjective measure of satisfaction. RAN Intelligent Controller (RIC) in O-RAN provides flexibility to intelligently program and control RAN functions using AI/ML-based models. We argue that various QoE parameters can be measured and operated by the RIC in O-RAN. We propose to improve the efficiency of O-RAN's radio resources by creating a RIC xApp that estimates the QoE measured using Video Mean Opinion of Score (MOS), and accurately optimizes the usage of radio resources across multiple network slices. We use predictive AI/ML-based models to accurately predict the QoE parameters in the network after which we can optimize the usage of network compo-nents leading to an enhanced user experience. Simulation results on 3 simulated data sets show that our proposed approach can achieve up to 95% QoE prediction accuracy.

Keywords: O-RAN, RIC, 5G, slicing, AI/ML, QoE, MoS

1. Introduction

With the advent of fifth and sixth-generation cellular networks, there are a lot of challenges in the implementation of new and advanced policies owing to the diverse demands of this technology. Some major use cases of fifth-generation networks include enhanced mobile broadband (eMBB), ultra-reliable and low-latency communications (uRLLC), and massive machine-type communications [1]. The existing RAN architecture consists of using proprietary and aggregated components that are mostly inaccessible to third-party vendors. Open Radio Access Network (O-RAN) is a new collaborative approach that involves vendors, equipment makers, and telecom industry members to include non-proprietary components in existing RAN architectures.

The O-RAN architecture is decoupled as it involves using components from different sources. A key component of the O-RAN architecture is the RAN Intelligent Controller (RIC). The RIC controls the functionality of the RAN and is a key component where we can add intelligence by implementing AI/ML algorithms. Fur-ther, this component saves infrastructural costs as it is virtual and can be used to solve business use cases as well. Adding intelligence in the RIC can help solve use cases such as predicting the QoE along with providing O-RAN resource management. The O-RAN architecture involves off-the-shelf decoupled components that can be built by third-party vendors. This is in alignment with 3GPP specifications which promote decoupled Next Generation Node Bases (gNBs).

Copyright ©2023 Aditya Padmakar Kulkarni, et al.
DOI: https://doi.org/10.37256/cnc.1220233661
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

Figure 1 shows the general architecture of O-RAN. Each gNB is split into various components including a Central Unit (CU), Radio Unit (RU), and Distributed unit (DU). These are also called O-CU, O-RU, and O-DU respectively in O-RAN terminology. The O-CU is further split into two logical divisions, the control plane (CU C-plane) and the user plane (CU U-plane). By splitting these two components, they can be individually accessed across different hardware planes and different locations as well. The intelligence aspect of O-RANs is handled by the RAN Intelligent Controller (RIC). The RIC is one of the most critical aspects of the O-RAN architecture. Figure 2 shows the architecture of the RAN Intelligent Controller. This component allows programmability in O-RANs and allows third-party vendors and applications to include their software in the architecture. This gives rise to various use cases also involving AI/ML to improve the quality of the network. The O-RAN Alliance defines a number of use cases that can be worked upon using RICs [2]. The programmable components of RIC can generally be divided into two categories. These include non-real-time (non-RT) and near-real-time (near-RT) components. The non-RT components comprise system events that have a response time of 1 second or more. Near-RT components involve events that have a response time of fewer than 10 milliseconds. Depending on this, the RIC software consists of applications in the form of microservices called xApps and rApps. The xApps and rApps can be defined as tools for the automation of the network. These microservices specialize in individual functionalities for specific tasks. xApps are created for dealing with near-RT events and rApps deal with non-RT events. xApps usually reside on the edge nodes of the network as they read data and provide functionalities in real-time. On the other hand, rApps reside on the operator's network as these functionalities are allowed to have a larger response time.

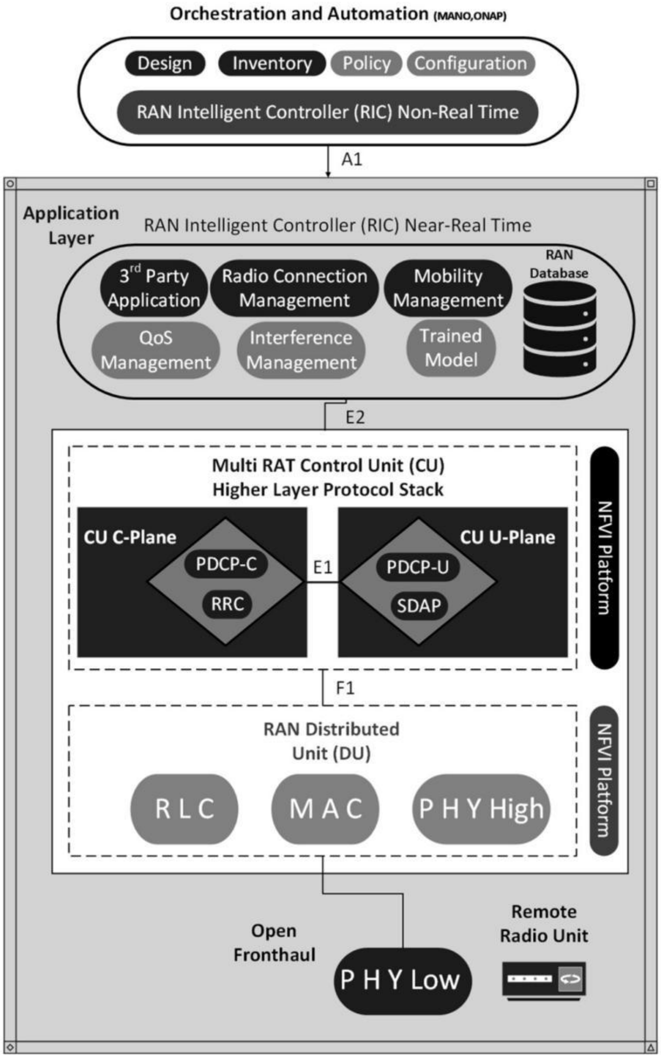

Figure 1. O-RAN Architecture.

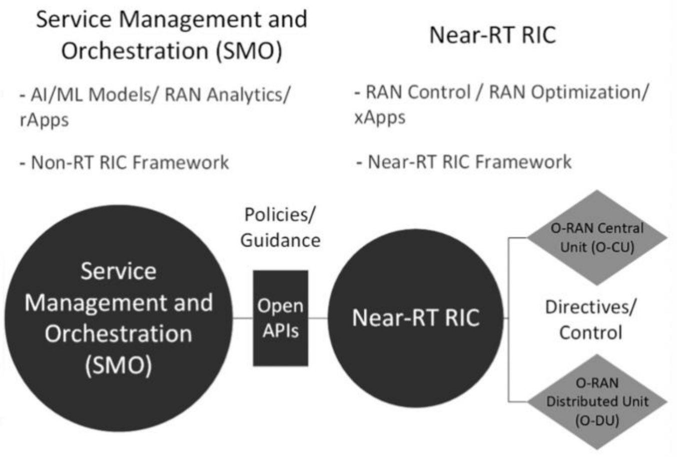
Service Management and Orchestration (SMO)
- AI/ML Models/ RAN Analytics/ rApps
- Non-RT RIC Framework
Near-RT RIC
- RAN Control / RAN Optimization/ xApps
- Near-RT RIC Framework
Service Management and Orchestration (SMO)
Policies/ Guidance
Open APIs
Near-RT RIC
O-RAN Central Unit (O-CU)
Directives/ Control
O-RAN Distributed Unit (O-DU)

Figure 2. RIC Architecture.

The technology of O-RANs is new and brings with it a lot of challenges and opportunities. Current research on O-RAN demonstrates how intelligence can be added to solve new and innovative use cases. Our aim is to demonstrate how an O-RAN can learn from multiple network slices in the same RAN. Quality of Service or QoS refers to the mechanisms, metrics, and set of technologies that are used to prioritize and manage the network traffic ensuring that certain criteria or performance levels are met. Quality of Experience or QoE represents the measure of the overall level of satisfaction that an end user perceives while using a service or an application. This metric takes into account multiple factors such as expectations of users, application quality, application responsiveness, and performance of the network. QoE is a crucial factor as it takes into consideration the opinions of an end user. Even if an application meets all QoS requirements, it could be the case that the end user might not have a good experience with it. In the context of telecommunications and networks, this plays a big role as networks are designed for users to have a smooth and pleasant experience while being connected to a network. These metrics are also critical for real-time applications like emergency services and healthcare [3]. For a network, a high QoE also enables a high rate of customer retention. In a competitive market, a high QoE rate of customers is very beneficial for network providers as it reduces churn rate. We demonstrate how the QoE can be improved for customers by predicting the Mean Opinion Score (MOS) across network slices. This shows the potential of O-RANs on how data can be collected from each network slice, and how intelligence can be applied to enhance the user experience of customers using the network.

We propose a solution to create an xApp that predicts the QoE of a 5G network across multiple network slices. The following are our contributions:

- We demonstrate that an O-RAN serving multiple network slices can be used in a way that it learns data across each network slice and we can now optimally predict the QoE across any network slice.
- We use 3 different datasets with varied features to simulate 3 network slices, each dataset consisting of streaming video parameters such as video frame rate, video bit rate, audio rate, and more.
- We use different classification-based machine learning models to learn the Mean Opinion Score (MOS) class for each dataset.
- Finally, we use this to predict the MOS of any new video. From this, we can show that the O-RAN can learn from different network slices, and can use this to predict the MOS among the network slices or new network slices.

This way, the QoE can be managed and improved with the help of the RIC in an O-RAN.

The rest of the paper is organized as follows. Section 2 reviews the existing work in O-RANs and intelligence-based use cases of O-RANs. Section 3 elaborates upon AI/ML-based techniques and how we have utilized these techniques to solve our use case. We discuss our implementation details and results in Section 4. Section 5 concludes the paper with pointers to future work.

2. Related Works

The technology of Open-Radio Access Networks is new and is increasing in popularity given its virtualized and decoupled nature. The O-RAN Alliance, a group that looks after the specifications and operating procedures of O-RANs, was first started by 5 major operators namely AT&T, China Mobile, Deutsche Telekom, NTT

Docomo, and Orange. With the growing popularity of O-RANs, it has now increased to over 300 operators worldwide [4]. This is owing to the strong professional and academic research in this domain. Garcia-Saavedra and Costa-P´erez describe how O-RANs have created major breakthroughs in next-generation virtualized radio access networks (vRANs) [5]. They describe the growing popularity of O-RANs, how this technology is now adopted across 4 continents, along with deployment scenarios of intelligent O-RANs. Polese et al. give an in-depth explanation of O-RAN specifications while elaborating the security aspects of O-RANs [6]. They have provided the first detailed tutorial on O-RANs and have explained how the design principles and interfaces of O-RAN are important for researchers and industry practitioners. Abdalla et al. explain the limitations of O-RANs and the challenges that this technology faces [7]. Some of these limitations include security aspects such as having end-to-end security over an O-RAN. Moreover, testing intelligent RANs and having real-time control over the physical layer of O-RANs is also a limitation. However, they also discuss the opportunities for researchers to overcome these challenges. Dryjanski et al. [8] propose a traffic steering use case. This is another one of the use cases from the O-RAN Alliance.

It has been shown that AI/ML techniques can be used to predict the QoE [9]. However, current research does not demonstrate how this progress can be integrated into an xApp for a RAN Intelligent Controller (RIC) in O-RANs. Kavehmadavani et al. propose a solution for a traffic steering use case in 6G O-RANs [10]. They use joint implementation of scheduling, flow-split distribution, and congestion control to add intelli-gence to O-RANs to optimize traffic steering. Qasim et al. proposed a personalized QoE model specifically for data-driven architecture (DDA) in 5G wireless networks [11]. This research was specific to a problem that many organizations face in Enterprise Architecture. Agarwal, Togou, Ruffini, and Muntean introduced a novel QoE enhancement function to provide efficient resource provisioning to users [12]. However, this research does not use AI/ML techniques and instead uses an adaptive genetic algorithm for resource provi-sioning. Schwarzmann et al. estimate video streaming QoE in 5G architecture using regression techniques on a mobile-streaming use case [13]. They proved that the subjective QoE score can accurately be predicted using linear regression techniques. Ahmed et al. used supervised machine learning techniques for QoE prediction on a video streaming dataset [14]. They demonstrated important features of the dataset and showed how Support Vector Regression, Random Forest classifiers, and neural networks performed based on the video dataset.

3. Machine Learning Algorithms

The RAN Intelligent Controller (RIC) can be programmed to add intelligence to the O-RAN. This can be done by creating AI/ML-based models that learn from the data accumulated by the RAN. Model training is generally done at the Service Management and Orchestration (SMO) level. We treat the QoE prediction problem as a classification problem. We have based our models on the techniques explained in the following subsections:

3.1 *Logistic Regression*

Logistic regression [15] is a supervised machine-learning technique that aims to classify the dependent variable with one or more independent variables. This algorithm estimates the probability of a data point falling into a particular class depending on the given input parameters. We use the multi-class logistic regression model, which is an extension of logistic regression over multiple features. Logistic regression uses the Softmax function [15] to create a probability distribution to predict the target class. The equation of a logistic regression model is defined in Equation 1:

$$P\left(y = j, \middle| x\right) = \frac{e^{w_j \cdot x + b_j}}{\sum_{i=1}^{k} e^{w_i \cdot x + b_i}} \tag{1}$$

where
P = probability of MOS to fall under particular class
k = total classes for MOS values between 1 to 5
w = weight vector
b = bias term, an adjustable parameter to tune the model
x = vector of predictor variables; this includes parameters such as audio rate, video buffer rate, frame rate, etc.

Figure 3 shows an example of how logistic regression is used to classify a target class into multiple categories. For our implementation, the response variable is the Mean Opinion Score (MOS) which can fall under 5 categories ranging from 1 to 5. Our input parameters consist of multiple features including frame rate, buffering time, audio rate, etc.

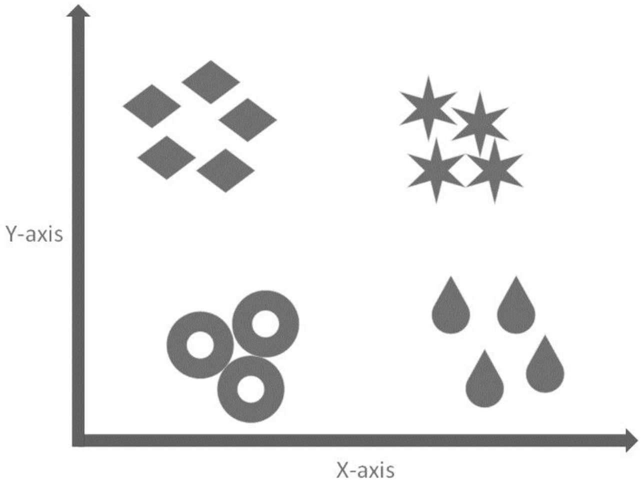

Figure 3. Logistic Regression.

3.2 *Support Vector Machine*

For a Support Vector Machine-based classification model [15], we aim to find a hyperplane that best separates the MOS target score into different classes. This hyperplane will create a decision boundary between the target classes. Once we train the SVM model, the RIC xApp will be able to predict the MOS for new data points. This model training will be done on a dataset that provides historical information of video quality. During model training, the hyperplane will be adjusted to get a clear distinction between the 5 categories of MOS. The mathematical function is described by Equation 2:

$$wx + b = 0 \tag{2}$$

where
x = the input vector of QoE factors
w = weight vector
c = bias

The weights and biases are terms that are learned during the model training phase as they aim to optimize a cost function that gives a penalty on incorrect classifications and tries to increase the margin between target classes. For the target class of MOS, the values are divided into 5 categories. These categories are explained further in Section IV. The SVM classifier will aim to find a hyperplane that divides the data for each class to show separate groups. Any new data can be classified into the target classes by applying the SVM function and the positive or negative sign of the value can determine the class of the new data point. Figure 4 shows how SVM determines the hyperplane and maximizes the margins between different target classes.

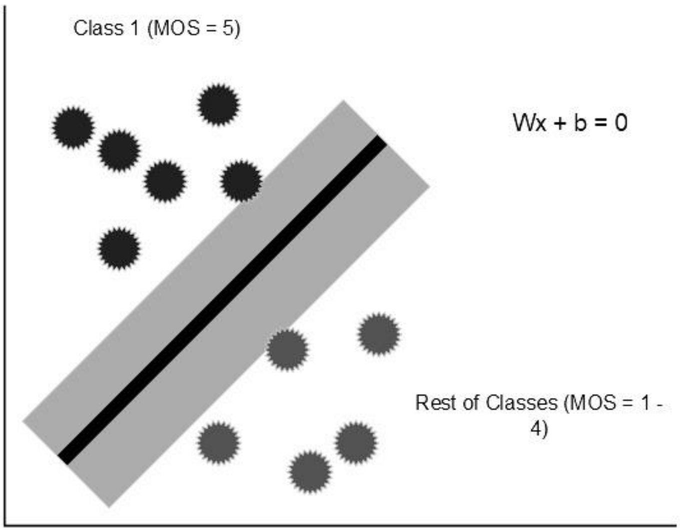

Figure 4. SVM Classifier

3.3 *Decision Tree*

The Decision Tree [15] is a machine learning algorithm that is used for both regression and classification problems. Our decision tree-based machine learning model maintains a tree-like structure with features such as audio rate, buffering time, playing length, etc. as nodes of the tree. Each branch of this tree denotes the possible range of values of these features. For example, a node of the feature ’playing length’ can have two branches, one for videos with a playing time of less than 4 minutes, and the other for videos with a playing length over 4 minutes. While training the model, this tree recursively creates branches by keeping features at the nodes. Hence, the job of the intelligent RIC will be to check the values of a new data point and find the leaf nodes that determine the category of the MOS. Figure 5 describes the working of a decision tree model.

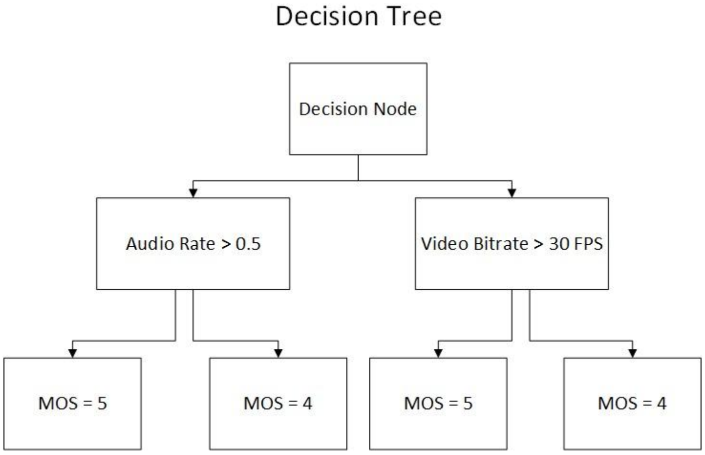

Figure 5. Decision Tree Classification.

The leaf nodes of the decision tree determine the output or the final predicted values from the given input parameters. Decision trees are generally good at dealing with both categorical and continuous data. Since we use 3 datasets to simulate our network slices, we create 3 models using decision trees. Each model will have a different number of decision nodes depending on the dataset’s features. For example, one of the features in the dataset is Round Trip Time (RTT). The decision tree will have a decision node that checks if the given data point’s RTT value is within some given threshold. If so, it will then proceed to the next decision node which checks the value for another feature. The branch terminates into the leaf node which determines the MOS value for the given data point. In this way, our model will determine the MOS for new data points by checking its values at each decision node and leading to the leaf node.

3.4 *Random Forest*

The Random Forest classifier [15] follows the method of ensemble learning as it uses decision trees to build its model. In this case, our RIC xApp will be trained with a model that merges multiple decision trees to create a robust and accurate model for MOS classification. For this, the model starts with taking a random subset of features from our MOS dataset, along with a random sample of data to first create a decision tree. This process is repeated over multiple iterations, each time taking a different sample of data and a different subset of features. The leaf nodes of these decision trees will be the MOS categories. These decision trees are combined to form a random forest. This combination is done by finding the majority of trees that choose a predicted target value. This leads to creating a more robust model and tends to not overfit the model, unlike a single decision tree. Hence, this will cause the RIC to have a strong and robust model to classify the MOS based on new input data. Figure 6 describes a random forest classifier.

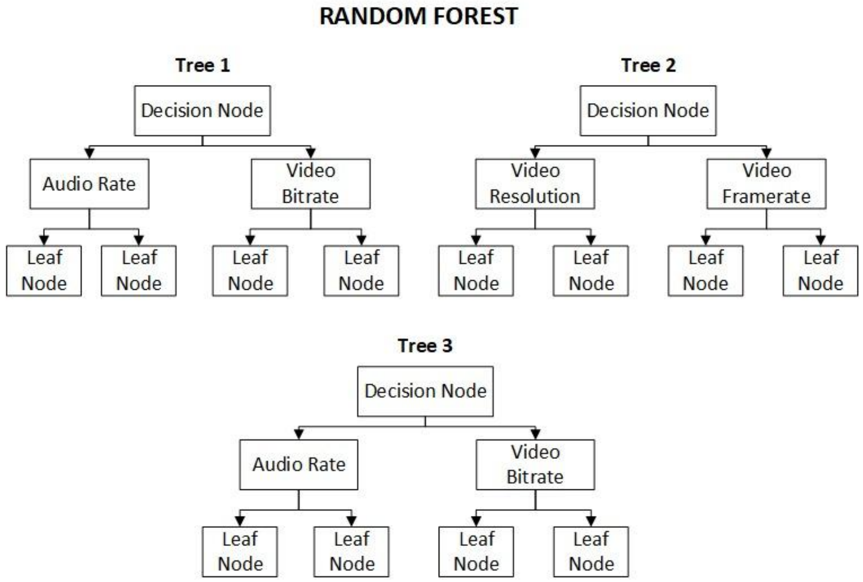

Figure 6. Random Forest Classifier.

3.5 *Naive Bayes*

A Naive Bayes-based [15] classifier model in the RIC will use the Bayes theorem to determine the probability of a new data point falling into a particular category of the MOS. This is also a supervised learning algorithm. The mathematical equation for the Bayes' theorem is as Equation 3:

$$P\left(\text{class}\middle|\text{features}\right)=\frac{P\left(\text{class}\right)\prod_{i=1}^{n}P\left(\text{feature}_i\middle|\text{class}\right)}{\sum cP\left(c\right)\prod_{i=1}^{n}P\left(\text{feature}_i\middle|c\right)} \tag{3}$$

where
n = the number of features in the QoE dataset
i = index of the feature
c = index of the MOS class.

Naive Bayes makes the assumption that the features and the target class are conditionally independent. For our 3 datasets, we create 3 probabilistic models to predict the MOS value. Each dataset consists of different features and hence, each model will calculate different MOS values according to Equation 3.

3.6 *K-Nearest Neighbors*

The K-nearest neighbors (KNN) [15] is a supervised machine learning algorithm that can be used in an RIC xApp. For this algorithm, we choose a value for k. Given a new QoE measurement, the model finds the k-closest points from this given data point. From these k-neighbors, it finds the majority of the MOS value. The most common value of the MOS from these neighbors is predicted to be the target for the new data point. This way, the RIC will be able to predict the MOS of a new video using KNN. Figure 7 shows an example of K-Nearest Neighbors.

This algorithm does not learn any particular type of model from the given data. It is termed as a lazy learning algorithm that predicts values from the training data stored in its memory store. Our KNN model uses a k-value of 5. This means that for every data point whose MOS needs to be predicted, we check the 5 nearest points to it when plotted on a graph. These points will be plotted based on the values of their features. We use the Euclidean distance metric to find the 5 nearest points and then check the target class values for these 5 nearest neighbors. The majority of these will be chosen as the MOS value for our new data point.

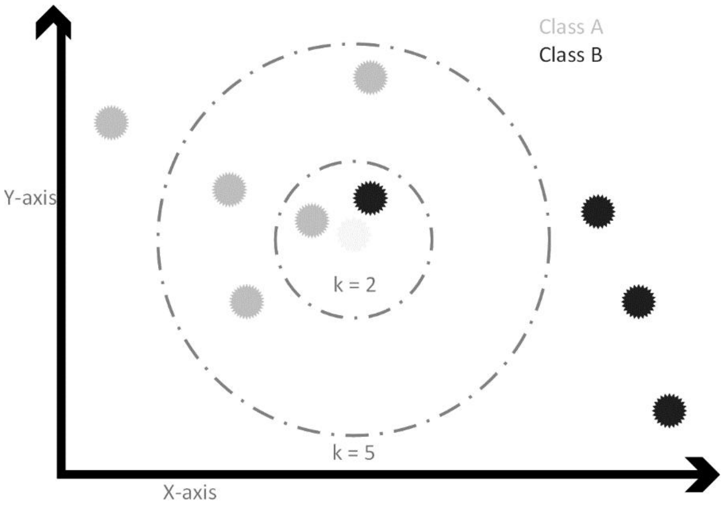

Figure 7. KNN classification.

4. Simulation Experiments and Results

We consider a use case where our O-RAN has 3 network slices. Each network slice serves a different user group. We consider streaming video data as the service that is offered to the users.

4.1 *Data Set Preparation*

To simulate the 3 network slices, we consider 3 different datasets of video data in .csv formats. Each dataset has varied features and a different number of data points. This ensures that each network slice is offering a different type of content to its users. The 3 datasets are explained ahead.

For the first network slice, we use the PoQeMoN dataset [16] which consists of multiple QoE Influence Factors (QoE IFs) along with Mean Opinion Score (MOS) values. This dataset is based on users' opinions on several different types of videos viewed on different types of devices. These IFs are collected using a VLC media player. The 22 influence factors include video resolution, video bitrate, framerate, audio rate, audio loss, buffering rate, and more. The MOS is spread into 5 classes with values from 1 to 5. Table 1 shows the 5 classes of MOS for the given dataset. The 22 features of the dataset are described in Table 2.

Table 1. Mean Opinion Score categories.

MOS values	
1	Bad
2	Poor
3	Fair
4	Good
5	Excellent

Table 2. Dataset 1 Features

id	user id	QoA VLCresolution	QoA VLCbitrate
QoA VLCframerate	QoA VLCdropped	QoA VLCaudiorate	QoA VLCaudioloss
QoD model	QoD os-version	QoD api-level	QoU sex
QoA BUFFERINGcount	QoA BUFFERINGtime	QoS type	QoS operator
QoU age	QoU Ustudy	QoF begin	QoF shift
QoF audio	QoF video		

For the second network slice, we have used Huawei's dataset based on their SpeedVideo Global Operating Platform (SVGOP) [17]. This dataset is in Simplified Chinese but Wang et al. [18] have translated it into English. This dataset consists of 89,266 entries of video data entries with 15 features and a target class of vMOS (video MOS). These features include parameters such as the Average rate of playing phase, Initial buffering latency, Stalling rate, etc. Table 3 describes the features of this dataset. The vMOS value is a continuous floating point score between 1 to 5. We use normalization techniques for our implementation to set a fixed class value for these scores.

Table 3. Dataset 2 Features

Max DL Rate(kbps)	E2E RTT(ms)	Average Rate of Playing phase	Initial Buffering Latency(ms)
Stalling Ratio	Video DL Rate	Stalling Length	Playing Length
Total time of Playing phase (kbps)	Stalling times	Video Bitrate	Video Initial buffer download
Quality	Loading	Stalling period	-

The third network slice consists of a dataset with 9575 data points [19]. It consists of 7 features including download throughput, download loss rate, and stalling length. Table 4 shows the features of this dataset. The target class is the MOS which has a score between 1 to 5.

Table 4. Dataset 3 Features

Session	RTT	Download lossrate
Download throughput	Stalling number	Stalling length

Each dataset has important features regarding the video content that is provided to the users. These include network centric parameters such as Round Trip Time (RTT), as well as characteristics of each video. This includes Buffering time, Audio loss rate, Stalling time, Stalling length, and more. The datasets were meticulously chosen based on their ability to represent video transmission content over diverse network slices in the O-RAN architecture.

For learning the QoE of users on video streaming data in an O-RAN, each network slice must be treated as a separate entity. Each network slice serves video content in a different manner to its users. We have demonstrated this by using different datasets with varied features across each network slice. Hence, we will have a different machine learning model that goes through data preprocessing, model training, parameter tuning, feature selection, and finally prediction for new data. We explain our machine learning models and their results ahead.

4.2 *Model Implementation*

For the first dataset, we have 22 features and a target class of the MOS. We start by preprocessing the data and identifying features that have string values. For the features QoD Model and QoD OS, which represent the device model and device operating system, we use a One Hot Encoder to convert these string values into a numeric array. From the feature set, we identify 17 of the 22 features to be relevant and which help gain the best accuracy. Table 5 shows our selected features from this dataset.

Table 5. Dataset 1 Selected Features

id	audio	VLCbitrate
VLCframerate	VLCdropped	VLCaudiorate
Buffering count	Buffering time	VLC resolution
video	model	os-version
age	begin	shift

For feature selection, we use the Recursive Feature Elimination (RFE) method. This is a technique used in machine learning that eliminates the least important features in a dataset. This process helps in fine tuning the model by focusing on the most informative features. In this method, the model is trained on the entire set of features, feature importance ranking is performed for each and finally the least important feature is eliminated. This process is repeated until a pre-determined number of features is obtained.

The dataset comprises various features related to media playback and user characteristics. Audio attributes include bitrate and audiorate, while video features encompass framerate, dropped frames, and resolution. Buffering events are captured through count and time metrics. Additional information includes details about the video model, operating system version, user age, and timestamps indicating the beginning and duration of events or shifts. These features collectively offer insights into the performance of media streams, forming a comprehensive dataset for analysis and modeling.

We then apply different machine learning models to train the data. We split the data into 80% training and 20% testing. We used the train-test split function from the sci-kit learn library to divide the data into 80% training and 20% testing. This is done so that we can learn the features of the training data and then test our model on the test set. We then try out different regression and classification machine learning algorithms on this data. These algorithms are Logistic Regression, Random Forest Classifier, Naive Bayes Classifier, K-Nearest Neighbours, Decision Tree Classifier, and SVM classifier. After learning the data, we predict the MOS for the test set and check its actual accuracy. The summary of the accuracy results is displayed in Figure 8. For this dataset, we achieved the highest accuracy of 85.43%. This is using the Random Forest Classifier. The SVM model has an accuracy of 81%. On the other hand, KNN has an accuracy of 77.9%, the Naive Bayes model has 60.5% accuracy, and Logistic Regression has 58.5% accuracy. We have summarized the parameters of our best-performing model, the Random Forest-based classifier, in Table 6. Further, we plot the F1 score curve for the 5 categories of VMOS, the target class. From the F1 scores, we find that our predictions for the categories ’Bad’, ’Good’, and ’Excellent’ are over 0.80 and show that our prediction values are accurate. The F1 score plot is described in Figure 9.

Table 6. Parameters for Random Forest classifier

Number of estimators	100
Estimator	Decision Tree
Cross Validation	4

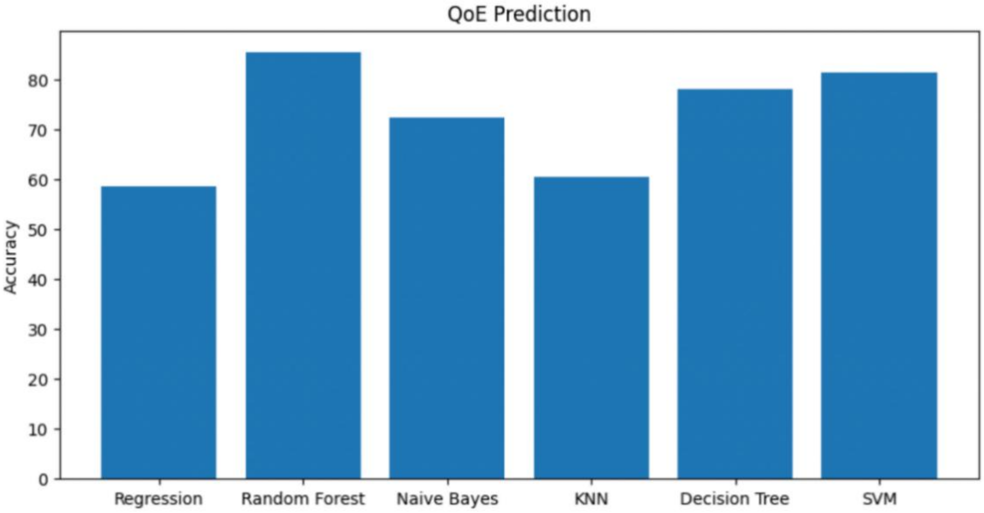

Figure 8. Results for the first dataset

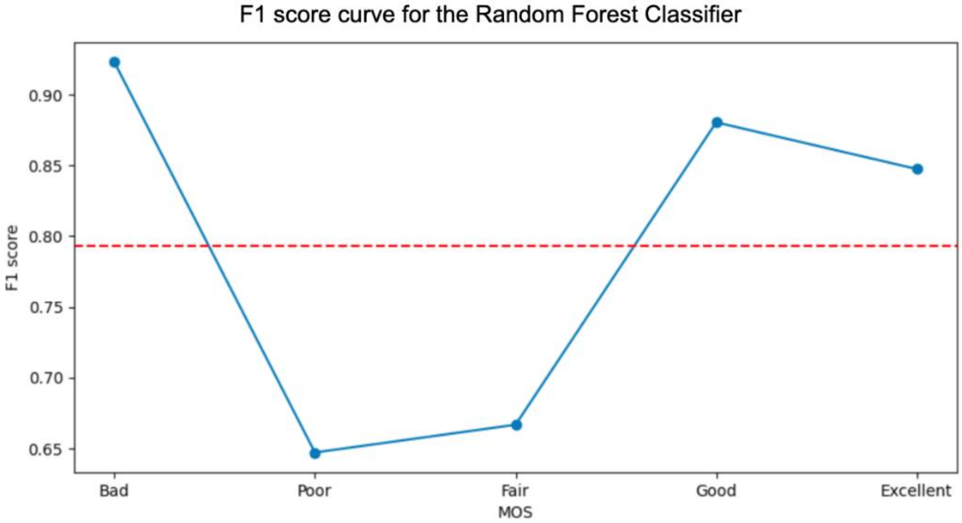

Figure 9. F1 Score Curve for the first dataset.

We follow a similar approach for the other two datasets. Dataset 2 consists of 89,266 data entries which we again split into 80% training and 20% testing data. This dataset does not consist of any String values and hence, there is no need for any such conversion. However, the target class, VMOS, has a floating point score between 1 to 5. To better train our model, we normalize this floating point score into a non-decimal value fixed score in classes 1 to 5. We describe the selected features in Table 7.

Table 7. Dataset 2 Selected Features

Initial Buffering Latency	Playing Length	S Quality
Video bitrate	S Loading	S Stalling

The results of the applied machine learning algorithms are shown in Figure 10. We gain the highest testing accuracy by using a Decision Tree Classifier. We achieved an accuracy of 97.91% for this dataset. The K-NN classifier with a k-value of 5 neighbors also performs well with an accuracy of 96.6%. Further, the Random Forest classifier also performs well with 96.2% accuracy and the Naive Bayes model has an accuracy of 71.23%. We also notice that multi-class logistic regression performs poorly with this dataset. Further, we plot the F1 score curve for the 5 categories from the Decision Tree model. From the F1 scores, we find that our predictions for all 5 categories of MOS are over 0.90 and show that these categories of MOS can be predicted accurately. The F1 score plot is described in Figure 11.

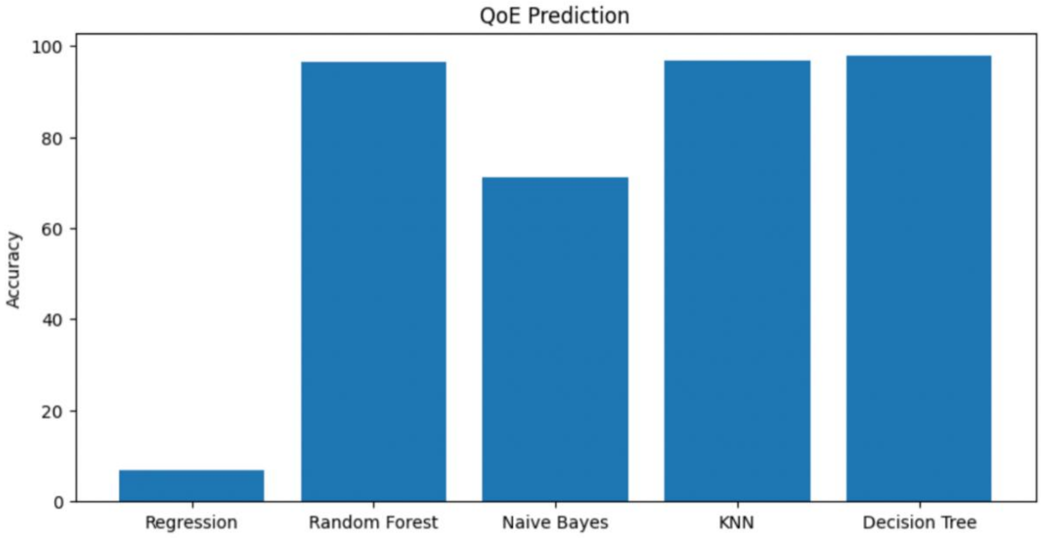

Figure 10. Results for the second dataset.

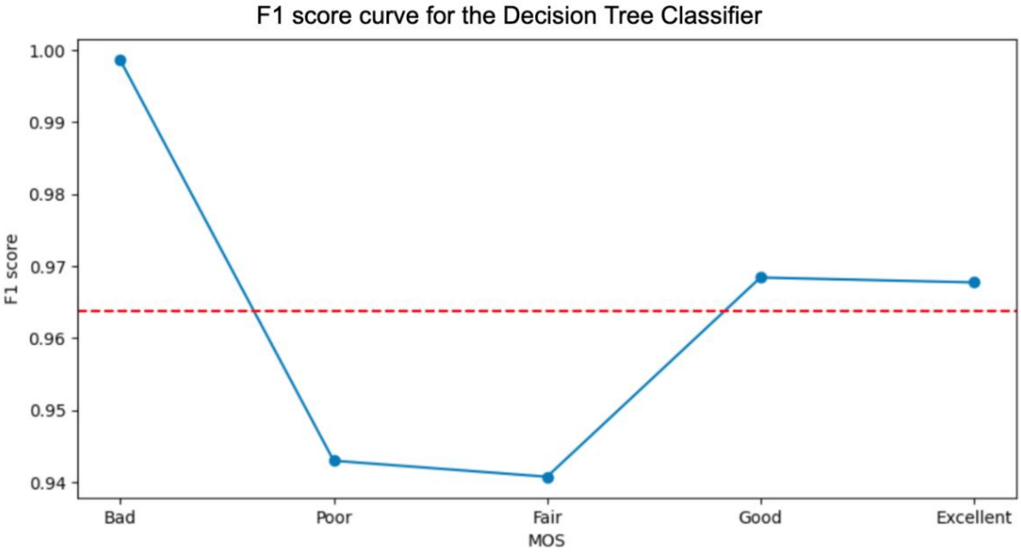

Figure 11. F1 Score Curve for the second dataset.

For the final dataset, the data is clean and does not require any preprocessing step. We choose all 7 features to train our model as shown in Table 8. Similar to the previous datasets, we split the data into 80% training and 20% testing data. Table 5 shows the chosen features from this dataset.

Table 8. Dataset 3 Selected Features

Session	Stalling Number	VLCbitrate
Join Time	Download Throughput	RTT
Stalling Length	Download Lossrate	—

The results of this algorithm compared with others are shown in Figure 12. We achieved the highest accuracy of 98.7% using the K-Nearest Neighbour classifier. This model performed the best with a k value of 16. This means that the results with the highest accuracy are obtained when a data point's MOS value is predicted from a majority of 16 neighbors. The Decision Tree and Random Forest model also perform well giving an accuracy of 97.2% and 98.1% respectively. The Naive Bayes classifier gives an accuracy score of 85.9% and the Logistic Regression model has an accuracy of 83.2% Finally, we plot the F1 score curve for the KNN model, as described in Figure 13. From the plot, we can infer that the F1 score for all 5 categories of MOS is over 0.90. This shows that our model can accurately predict the MOS of new data points with high confidence. Hence, all the 3 models, that reflect our 3 network slices, can accurately predict the MOS from their respective historical data.

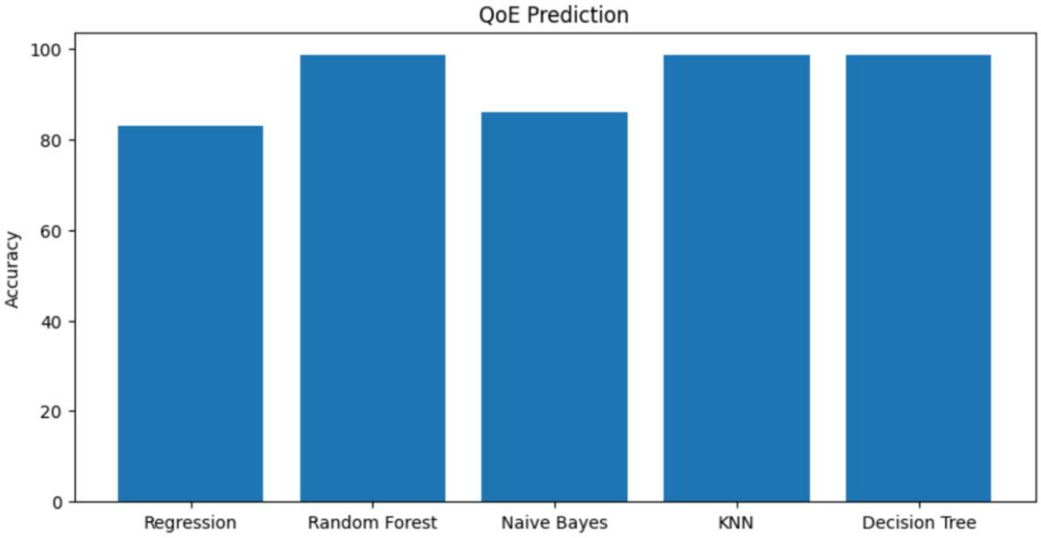

Figure 12. Results for the third dataset.

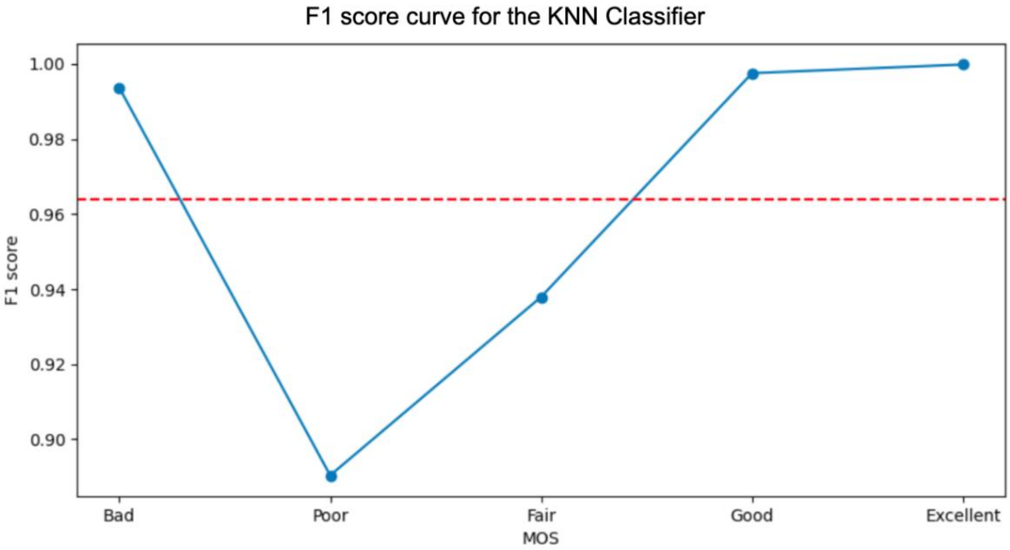

Figure 13. F1 Score Curve for the third dataset.

Across the 3 network slices, we notice that the Random Forest model performs the best for all 3 network slices. This is due to its ensemble learning capability, it provides an excellent measure to do QoE prediction. Random Forest models also handle non-linearity well and hence work well with a lot of features in the dataset. Further, the KNN classifier, which also gave a much better accuracy for datasets 2 and 3, is a much slower model. This will affect the timing requirements of certain applications. Further, other models such as Logistic Regression, Naive Bayes classifier, and decision trees did not perform that well. For the decision tree model, this could be due to the underlying data patterns of decision trees. Naive Bayes makes oversimplified assumptions of feature independence and that could be the cause for the low accuracy scores for it. Hence, Random Forest based models provide a much better estimate of predicting the Quality of Experience.

5. Conclusion and Future Work

Video applications and services are expected to dominate next-generation 6G wireless networks. Numerous emerging applications, like XR, connected cars, surveillance, and control depend on efficient video transmission and reception. O-RAN and its aspects of open network interfaces are also expected to penetrate more with the evolution of future cellular networks. In this paper, we have explored different ML models to make intelligent QoE predictions of different video applications across different network slices of O-RAN. Simulation results on the actual video data set give an accuracy score of 85.43%, 97.91%, and 98.7% across the 3 different 5G network slices. This demonstrates that intelligent O-RANs can be used to efficiently predict the QoE across network slices. The transfer learning aspect of this where the O-RAN can learn features from each network slice is something that we are currently working on.

The future scope for this project would include studying data-driven approaches in conjunction with fuzzy logic inference algorithms to present a robust methodology for modeling QoE in network scenarios. Research has shown that fuzzy sets and intuitionistic fuzzy sets can be used as a tool for QoE modeling in networks [20]. The future scope for this can include working on more use cases of intelligent O-RANs.

Conflict of Interest

There is no conflict of interest for this study.

References

[1] Dogra, A.; Jha, R.K.; Jain, S. A Survey on Beyond 5G Network With the Advent of 6G: Architecture and Emerging Technologies. *IEEE Access* **2020**, *9*, 67512–67547, https://doi.org/10.1109/access.2020.3031234.

[2] Alliance, O. O-RAN Use Cases, and Deployment Scenarios. Available online: https://assets-global.website-files.com/60b4ffd4ca081979751b5ed2/60e5aff9fc5c8d496515d7fe_O-RAN%2BUse%2BCases%2Band%2BDeployment%2BScenarios%2BWhitepaper%2BFebruary%2B2020.pdf (accessed on 15 February 2023).

[3] GSTR-5GQoE - Quality of experience (QoE) requirements for real-time multimedia services over 5G networks. Available online: https://www.itu.int/pub/T-TUT-QOS-2022-1 (accessed on 19 November 2023).

[4] Open and Transparent Way towards Open RAN by the O-RAN ALLIANCE. Available online: https://www.o-ran.org/blog/open-and-transparent-way-towards-open-ran-by-the-o-ran-alliance (accessed on 19 February 2023).

[5] Garcia-Saavedra, A.; Costa-Perez, X. O-RAN: Disrupting the virtualized RAN ecosystem. *IEEE Commun. Stand. Mag.* **2021**, *5*, 96–103., https://doi.org/10.1109/MCOMSTD.101.2000014.

[6] Polese, M.; Bonati, L.; D'oro, S.; Basagni, S.; Melodia, T. Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1376–1411, https://doi.org/10.1109/COMST.2023.3239220.

[7] Abdalla, A.S.; Upadhyaya, P.S.; Shah, V.K.; Marojevic, V. Toward next generation open radio access netw orks: What O-RAN can and cannot do!. *IEEE Netw.* **2022**, *36*, 206–213, https://doi.org/10.1109/MNET.108.2100659.

[8] Dryjański, M.; Kułacz, Ł.; Kliks, A. Toward Modular and Flexible Open RAN Implementations in 6G Net works: Traffic Steering Use Case and O-RAN xApps. *Sensors* **2021**, *21*, 8173, https://doi.org/10.3390/s21248173.

[9] Casas, P.; Seufert, M.; Wehner, N.; Schwind, A.; Wamser, F. Enhancing machine learning based qoe predic tion by ensemble models. In Proceedings of 2018 IEEE 38th International Conference on Distributed Comp uting Systems (ICDCS), Vienna, Austria, 2–6 July 2018, https://doi.org/10.1109/ICDCS.2018.00186.

[10] Kavehmadavani, F.; Nguyen, V.-D.; Vu, T.X.; Chatzinotas, S. Intelligent Traffic Steering in Beyond 5G Open RAN Based on LSTM Traffic Prediction. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 7727–7742, https://doi.org/10.1109/twc.2023.3254903.

[11] qasim EL-ezzi, Z.; Al-Dulaimi, A.M.; Ibrahim, A.A. Personalized quality of experience (QOE) management using data driven architecture in 5G wireless networks. In Proceedings of 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 22–24 October 2020, https://doi.org/10.1109/ISMSIT50672.2020.9254863.

[12] Agarwal, B.; Togou, M.A.; Ruffini, M.; Muntean, G.M. Joint Performance-Resource Optimization for Improved Video Quality in Fairness Enhanced HetNets. In Proceedings of ICC 2022-IEEE International Conference on Communications, Seoul, Korea, 16–20 May 2022, https://doi.org/10.1109/ICC45855.2022.9838454.

[13] Schwarzmann, S.; Breitbach, P.; Zinner, T. Computing qoe-relevant adaptive video streaming metrics using discrete-time analysis. In Proceedings of 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, 19–21 February 2019, https://doi.org/10.1109/ICIN.2019.8685906.

[14] Ahmad, A.; Bin Mansoor, A.; Barakabitze, A.A.; Hines, A.; Atzori, L.; Walshe, R. Supervised-learning-Based QoE Prediction of Video Streaming in Future Networks: A Tutorial with Comparative Study. *IEEE Commun. Mag.* **2021**, *59*, 88–94, https://doi.org/10.1109/mcom.001.2100109.

[15] Géron, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media, Inc.: Sebastopol, CA, USA.

[16] Amour, L.; Sami, S.; Hoceini, S.; Mellouk, A. Building a large dataset for model-based QoE prediction in the mobile environment. In Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Cancun, Mexico, 2–6 November, 2015, https://doi.org/10.1145/2811587.2811631.

[17] Video Mean Opinion Score Data Analysis over Mobile Networks. Available online: https://github.com/henryRDlab/VMOS (accessed on 19 February 2023).

[18] Wang, Q.; Dai, H.N.; Wang, H.; Wu, D. Data-driven QoE analysis on video streaming in mobile networks. In Proceedings of 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, China, 12–15 December 2017, https://doi.org/10.1109/ISPA/IUCC.2017.00168.

[19] Available online: https://github.com/wellimc/federatedlearning/blob/main/data/QoE_Linear_TrainingSet.csv (accessed on 19 February 2023).

[20] Torjemen, N.; em Zhioua, G.; Tabbane, N. QoE model based on fuzzy logic system for offload decision in HetNets environment. In Proceedings of 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet, Tunisia, 18–20 December 2016, https://doi.org/10.1109/SETIT.2016.7939918.