Feature Selection for Automated QoE Prediction

Tatsuya Kikuzuki*†, Mahdi Boloursaz Mashhadi*, Yi Ma*, and Rahim Tafazolli*
* *5GIC and 6GIC, Institute for Communication Systems, The University of Surrey,* UK
† *Artificial Intelligence Laboratory, FUJITSU LIMITED,* Japan
{t.kikuzuki, m.boloursazmashhadi, y.ma, R.Tafazolli}@surrey.ac.uk

***Abstract*—With the huge number of broadband users, automated network management becomes of huge interest to service providers. A major challenge is automated monitoring of user Quality of Experience (QoE), where Artificial Intelligence (AI) and Machine Learning (ML) models provide powerful tools to predict user QoE from basic protocol indicators such as Round Trip Time (RTT), retransmission rate, etc. In this paper, we introduce an effective feature selection method along with the corresponding classification algorithms to address this challenge. The simulation results show a prediction accuracy of 78% on the benchmark ITU ML5G-PS-012 dataset, improving 11% over the state-of-the-art result whilst reducing the model complexity at the same time. Moreover, we show that the local area network round trip time (LAN RTT) value during daytime and midweek plays the most prominent factor affecting the user QoE.**

***Index Terms*—user experience, machine learning, feature extraction, histogram analysis, feature importance**

I. Introduction

Quality of service (e.g., throughput, latency) has been evaluated to observe network performance so far, but it has turned out that it cannot be transformed to user experience directly [1]. For this reason, network management based on user Quality of Experience (QoE) has attracted considerable attention of network and service operators recently [2]–[4] where QoE is predicted from deep packet inspection (DPI) or Network Functions (NFs) data. Accurate prediction of the QoE is the key component in network management, and the researchers have developed predictive machine learning (ML) models such as support vector machine [5], decision tree [6], reinforcement learning [7], random forests [8], [9], long short-term memory [10], convolutional neural networks [11], deep neural network (DNN) [12]–[14] and transformer [15] to achieve this. Moreover, quality of the extracted features plays an important role in these models, and various feature extraction methods have been studied for QoE prediction such as manually selected statistics [4]–[9], a filter method [13], wrapper methods [10]–[12], [15], and an autoencoder [14].

The ITU AI/ML in 5G Challenge 2022 in collaboration with ZTE provides the problem statement and the benchmark dataset ML5G-PS-012 [16], where the aim is to develop accurate ML models along with the corresponding feature extraction methods to classify broadband users according to their QoE. This is a challenging problem as the benchmark dataset provides a small number of data, thereby being prone to overlearning. Moreover, the QoE labels seem inaccurate, not only due to the general noise of data acquisition, but also due to the fact that various broadband users seem to have very different expectations of service quality. For these reasons, accurate feature extraction becomes critical, and participants of the ITU AI/ML in 5G Challenge 2022 ML5G-PS-012 proposed different techniques, out of which the top 3 best performing ones are based on signal statistics (mean, etc.) [17], series dynamics [18], and wavelet transforms [19]. The state-of-the-art performance achieved is a 67% [18] accuracy in classifying broadband users into two groups: users with bad experience (UBE) and users with good experience (UGE), based on 8 protocol indicators.

In this paper, we propose a feature selection method for automated QoE prediction, where we pre-process the dataset and extract features based on insights from data networking protocols as well as an accurate histogram analysis. We are inspired by the fact that a histogram study is known to be effective for feature selection as it has been applied to a variety of data, including time-series sensor data [20], performance monitoring data [21], and image data [22]. Our numerical results show a prediction accuracy of 78% on the benchmark ITU ML5G-PS-012 dataset, improving 11% over the state-of-the-art result (i.e., 67%). The remainder of this paper is organized as follows. In section II, we rigorously introduce the ITU AI/ML in 5G Challenge 2022 ML5G-PS-012 problem statement. In section III, we describe our proposed feature selection method. In section IV, the performance of the proposed method is validated through numerical results. Finally, we present our concluding remarks in section V.

II. Problem Statement

In this section, we introduce the details of ML5G-PS-012. The competitors are required to classify each user data into UBE or UGE through time-series of 8 protocol indicators obtained from real networks by the optical line terminal (OLT) and the DPI. Fig. 1 shows the network layout and the definition of 8 indicators where the DPI is located as dividing the broadband end-to-end network into two parts: Local Area Network (LAN) side including optical network unit (ONU) and Wide Area Network (WAN) side including broadband remote access server (BRAS). More formally, the ith input time-series data is given as $\left\{x^{(1:8)}_{t_{s_i}:t_{e_i}}\right\}_i$, and the ith label is given as y_i (y_i = UBE or UGE). Indicator 1,2,4 is the time interval between the syn packet and the syn + ack packet, the syn + ack packet and the ack packet, or the first payload packet and the ack packet, respectively. Indicator 3 is the response interval of the first payload after the establishment of the TCP session. Indicator 5,6 is the data transmission latency from

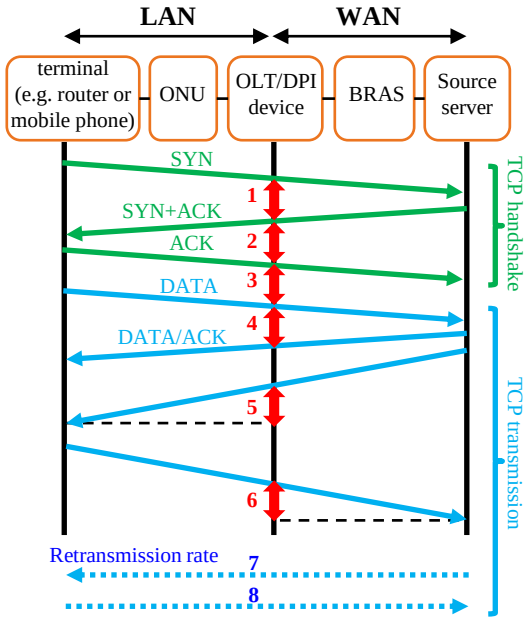

Fig. 1. Network layout and definitions of indicators.

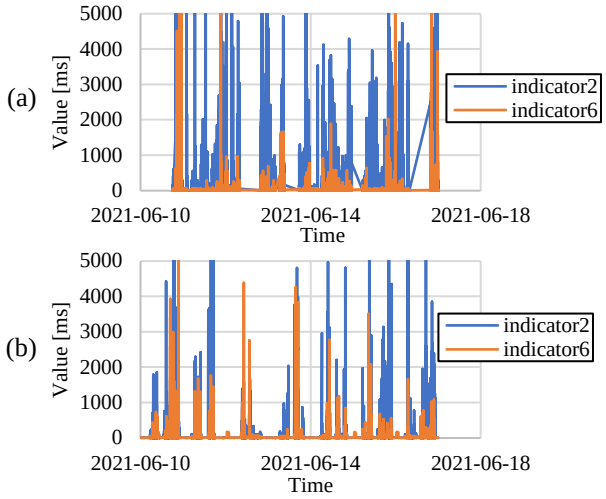

Fig. 2. Examples of indicator values of UBE (a) and UGE (b).

the DPI to the user terminal or to the server, respectively. Finally, indicator 7,8 represents the retransmission rate of downlink or uplink. Fig. 2 shows some examples of indicator values. The dataset contains a total of 500 ($1 \leq i \leq 500$) data including 300 for training, 100 for validation, and 100 for testing. Various methods are compared in terms of their classification accuracy on the test data.

The major challenge in ML5G-PS-012 is accurate feature extraction for the following two reasons. First, given the limited amount of data it is easy to overlearn. Each user data is dense as the time span ($t_{e_i} - t_{s_i}$) is generally 1 week and the number of raw time-series data is approximately 12000 though the number of train data is only 300. Furthermore, the QoE labels y_i seem inaccurate/noisy, due to the fact that various broadband users seem to have very different expectations of service quality. The other reason is that UBE and UGE seem to have very similar general statistics, e.g., mean value, standard deviation. In order to solve ML5G-PS-012, [18] proposed a solution with TSFresh library [23] focusing on dynamics of the indicator signals, achieving a classification accuracy of 67%. The feature extraction based on signal statistics (mean, etc.) and wavelet transformation explored in [17], [19] seem to yield lower accuracy values. To improve the results, a more detailed study of the indicator signals with insights from the network traffic characteristics seems necessary.

III. The Proposed Feature Selection

Fig. 3 shows an overall block diagram of the proposed method[1], in which the number of data N ($N = 300$ for training, or $N = 100$ otherwise), the feature dimensionality D ($D = 8$ at input), and the number of time-series data n_i ($n_i \approx 12000$ at input) are described above arrows. Firstly, we clean the input data and replace outlier values with the outlier threshold to remove invalid values and errors in the dataset. Secondly, we group the input data based on the meaning of each indicator and perform a histogram-based indicator splitting. Thirdly, we split each remaining sample along the measurement date (i.e., weekend or midweek) and time (3 ranges). Finally, mean values of time-series data of each dimension are selected as input features of a classifier. Note that a total number of time-series data processed at each block can be denoted as $\sum_{i=1}^{N} Dn_i$, and that N is fixed throughout as each block is processed repetitively to each user data. Algorithm 1 describes the proposed feature extraction procedure, and the following subsections explain each block in Fig. 3 with details.

A. Data Cleaning

Though the indicators 1,2,4,5,6 ($x_t^{(1,2,4,5,6)}$) represent round trip time (RTT), raw data include value zero for these indicators, which is not realistic. We remove the sample point if any of these indicators are zero because that sample point is considered not to be informative. In each user data, approximately 2000 time-series data are removed out of 12000 ones in average.

B. Outlier Replacement

In order to ensure that outliers don't distort statistical analysis, we replace outlier values with the outlier threshold. First, we tentatively set the outlier threshold as 3σ (standard deviation) for indicators 7,8 and by an interquartile range (IQR) method for the other indicators, and then tuned this threshold further in the final solution. The IQR method was adopted for indicators 1–6 because some values are very large as shown in Fig. 2 making the σ value meaninglessly large.

[1]The source code to reproduce all numerical results in this paper is available at https://github.com/University-of-Surrey-Mahdi/Histogram-Based-Feature-Selection-Method

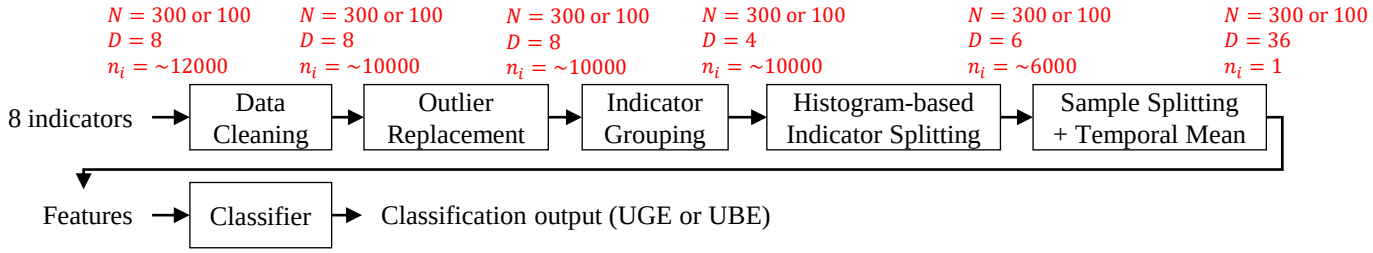

Fig. 3. Block diagram of the proposed method.

Algorithm 1: Feature Extraction in the ith input data

Input: $x_{t_{s_i}:t_{e_i}}^{(1:8)}$;
$X^{(R_1:R_6)(T_1:T_3)(D_1:D_2)} \leftarrow [\,]$;
for $t = t_{s_i} \cdots t_{e_i}$ **do**
  if $x_t^{(1)}$ *or* $x_t^{(2)}$ *or* $x_t^{(4)}$ *or* $x_t^{(5)}$ *or* $x_t^{(6)} = 0$ **then**
    delete $x_t^{(1:8)}$;
  else
    if $x_t^{(i_{indicator})} > Outlier^{(i_{indicator})}$ **then**
      $x_t^{(i_{\text{indicator}})} \leftarrow Outlier^{(i_{\text{indicator}})}$;
    end
    $x_t'^{(\text{WAN\_RTT})} \leftarrow \text{mean}\left(x_t{}^{(1,4,6)}\right)$;
    $x_t'^{(\text{LAN\_RTT})} \leftarrow \text{mean}\left(x_t{}^{(2,5)}\right)$;
    $x_t'^{(\text{LAN\_RES})} \leftarrow x_t{}^{(3)}$;
    $x_t'^{(\text{RET})} \leftarrow \text{mean}\left(x_t{}^{(7,8)}\right)$;
    if $x_t'^{(i_{group})}$ *is in* R_k **then**
      if *TIME* (t) *is in* T_k **then**
        if *DATA* (t) *is in* D_k **then**
          add $x_t'^{(i_{\text{group}})}$ to $X^{(R_k)(T_k)(D_k)}$;
        end
      end
    end
  end
end
Return mean $\left(X^{(R_1:R_6)(T_1:T_3)(D_1:D_2)}\right)$;

C. Indicator Grouping

We can categorize the input indicators into following 4 groups based on the definitions of indicators.

- WAN RTT: Indicators 1,4,6 ($x_t^{(1,4,6)}$) represent RTT in WAN
- LAN RTT: Indicators 2,5 ($x_t^{(2,5)}$) represent RTT in LAN
- LAN Response (RES): Indicator 3 ($x_t^{(3)}$) represents the response interval between two packets sent by the terminal
- Retransmission (RET): Indicators 7,8 ($x_t^{(7,8)}$) represent the retransmission rates in TCP transmission

For reducing noise, we calculate the average values of each group thereby reducing the feature dimensionality to 4.

TABLE I
GROUP RANGES IN "HISTOGRAM-BASED INDICATOR SPLITTING"

Group	Group range
R_1	$0 < x_t'^{(\text{WAN\_RTT})} < 19$
R_2	$19 \leq x_t'^{(\text{WAN\_RTT})} < 39$
R_3	$0 < x_t'^{(\text{LAN\_RTT})} < 23$
R_4	$23 \leq x_t'^{(\text{LAN\_RTT})} < 60$
R_5	$0 \leq x_t'^{(\text{LAN\_RES})} < 19$
R_6	$0 \leq x_t'^{(\text{RET})} < 0.18$

TABLE II
TIME RANGES IN "TEMPORAL SAMPLE SPLITTING"

Time	Time range
T_1	0AM $\leq$ TIME $(t) <$ 7AM
T_2	7AM $\leq$ TIME $(t) <$ 7PM
T_3	7PM $\leq$ TIME $(t) <$ 0AM

TABLE III
DATE RANGES IN "TEMPORAL SAMPLE SPLITTING"

Time	Time range
D_1	DATE (t) is in weekend
D_2	DATE (t) is in midweek

D. Histogram-based Indicator Splitting

The histograms of each indicator group are shown in Fig. 4. In order to extract informative features from histograms, we split time-series data of each indicator group based on which distribution of the histograms those belong to. WAN RTT and LAN RTT consist of multiple peaks as shown in Fig. 4 (a–b), which correspond to locations of application servers in WAN RTT, and to the type of link (local Wi-Fi or mobile carrier) in LAN RTT. Since different locations of servers or links are considered to have different impacts on user experience, we split time-series data of each indicator group to represent different distributions around each peak. In the final solution where the parameters are tuned, 6 group ranges are selected which are drawn as red arrows in Fig. 4 and listed in Table I. The time-series data are split and those out of any group range are deleted simultaneously, reducing n_i to approximately 6000.

E. Temporal Sample Splitting

How network performance affect user experience is considered to depend on the measurement time or date. As the time span in each dataset is generally 1 week, we split the time-

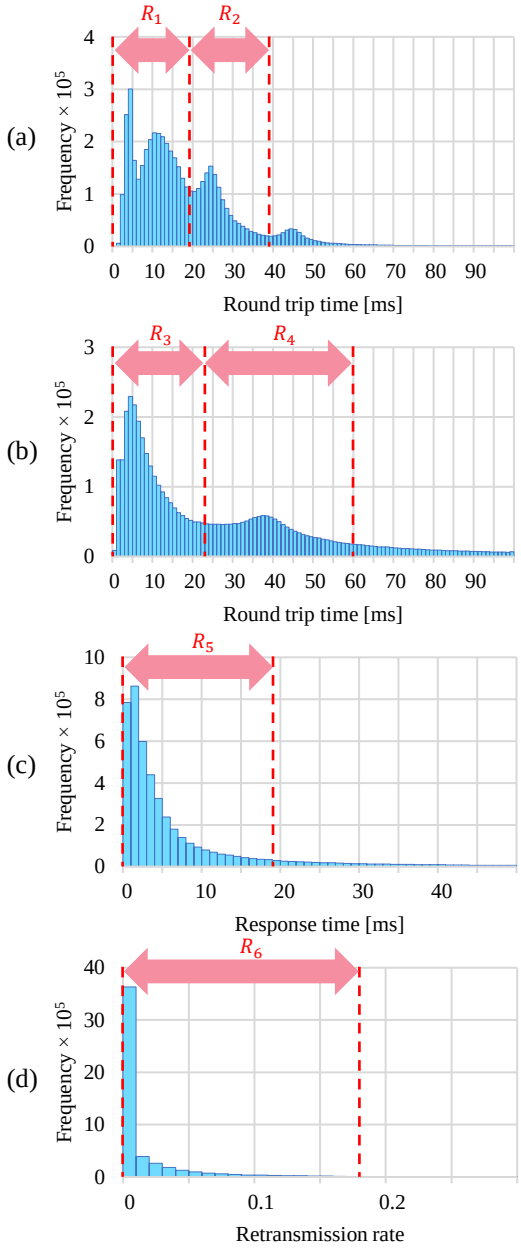

Fig. 4. Histograms of WAN RTT (a), LAN RTT (b), LAN RES (c), and RET (d).

series data into appropriate time and date ranges. In the final solution, we split into 3 time ranges (0am–7am, 7am–7pm, and 7pm–0am) and into 2 date ranges (weekend and midweek), which are listed in Table II and III respectively. Finally, the number of output dimension D is 36, which is calculated as 6 (output dimension of "D. Histogram-based Indicator Splitting" block) $\times 3 \times 2$. We then use temporal mean values of time-series data of each dimension as input features to the classifier.

F. Classifier

We select Random Forest as a classifier in the final solution. It provides the best classification accuracy among 11 ML classifier models, including XGBoost, DNN, Gaussian Naive Bayes, Gradient Boosting, Support Vector, K-Neighbors, Extra Tree, Ada Boosting, Decision Tree, Logistic Regression, and Random Forest.

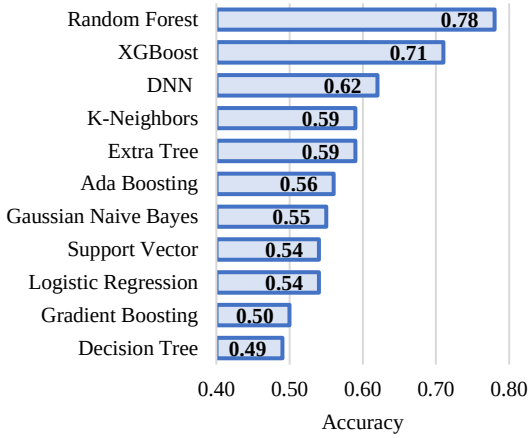

Fig. 5. Benchmarking of classifiers based on classification accuracy.

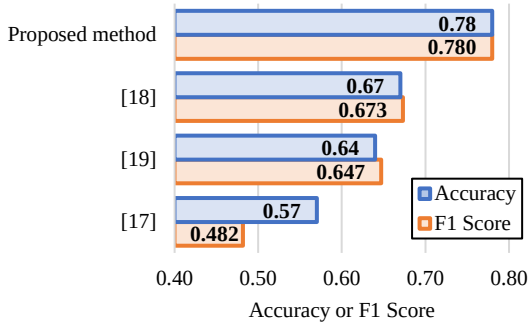

Fig. 6. Classification accuracy and F1 score.

IV. Numerical Result

In this section, we numerically evaluate the proposed method in ML5G-PS-012. In subsection IV-A, we compare the classification accuracy for the proposed method with previous studies. In subsection IV-B, we further compare the algorithm efficiency in terms of run time and a model size. Subsection IV-C provides a feature importance study to draw conclusions on the dominant factors that determine the user experience.

A. Classification Accuracy

Classification accuracy in ML5G-PS-012 is calculated by the prediction accuracy of 100 test data with a ML model trained by 300 train data. True label distribution is 50/50 between UBE/UGE.

Fig. 5 provides a comparison between 11 ML classifier models applied on the 36 extracted features. The result shows that Random Forest is the best classifier model, which is selected in our final solution. We compare the classification accuracy of the proposed method with previous studies in Fig. 6. This figure shows the advantage of our proposed method which improves the classification accuracy by 11% in comparison with the state-of-the-art (67% to 78%). Fig. 6 also

TABLE IV
RUN TIME FOR TRAINING

Method	Feature extraction [s]	Classification [s]
Proposed method	$1.8 \times 10^{+1}$	7.7×10^{-3}
[18]	$2.4 \times 10^{+4}$	4.6×10^{-1}
[19]	$2.8 \times 10^{+1}$	3.1×10^{-1}
[17]	3.2×10^{-1}	1.7×10^{-3}

TABLE V
RUN TIME FOR PREDICTING

Method	Feature extraction [s]	Classification [s]
Proposed method	6.1	7.5×10^{-4}
[18]	$8.0 \times 10^{+3}$	2.6×10^{-2}
[19]	9.2	1.2×10^{-2}
[17]	1.1×10^{-1}	4.0×10^{-4}

TABLE VI
MODEL SIZE

Method	Model size [kB]
Proposed method	33.5
[18]	119.6
[19]	495.0
[17]	0.7

shows that the F1 score of the proposed method is superior to those of the previous studies.

Finally, the cross validation results show a classification accuracy of 57.0±4.2% for the proposed method, which is considerably higher than the benchmarks, i.e. 52.2±4.6% [18], 51.3±4.7% [19], 51.9±4.3% [17]. These results show sufficient generalization performance of our proposed method.

B. Complexity and Execution Time

Run time and a model size are evaluated for comparing the algorithm efficiency of the proposed method with previous studies. Run time is calculated as the average of 10 repeated trials of training with 300 train data or of predicting with 100 test data on Dell OptiPlex 7050 (i7-7700 Quad-Core without GPUs, 16GB RAM). A model size represents the size of classifier model.

Table IV, V, or VI shows run times for training, those for predicting, or model sizes respectively where methods are sorted by classification accuracy. Table IV and V indicate that extracting feature from dense time-series data is a bottleneck in execution time for any method and that complexity of a model is determined by that of feature extraction. Though the lightest method is the method in [17] which just extracts the mean values from input data, the classification accuracy of that is the lowest 57%. The proposed method is the lightest among those whose classification accuracy exceed 60% in terms of both run time and a model size. Thereby, the evaluation result shows that our proposed method has the advantage of algorithm efficiency.

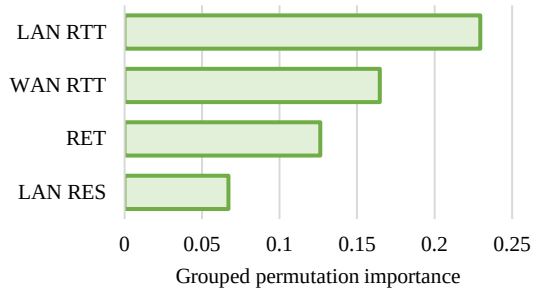

Fig. 7. Permutation importance for each group.

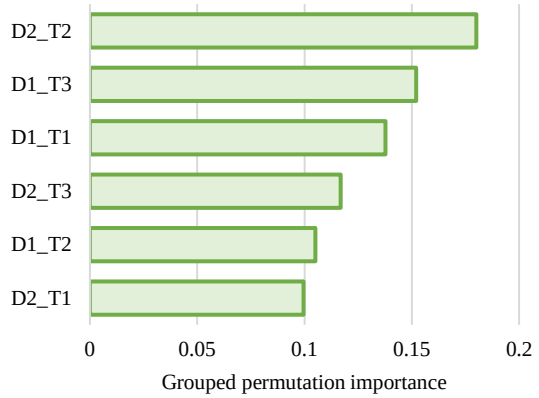

Fig. 8. Permutation importance for each time and date range.

C. Feature Importance

In addition to the evaluation of classification accuracy and algorithm efficiency, we show the key factors affecting the user experience through a feature importance study. To this end, we apply grouped permutation importance [24] to the test dataset. Grouped permutation importance is the extended version of permutation importance and can provide not only the importance of each input feature, but also that of any feature subset. Permutation importance of features $[j_1, j_2, \cdots, j_n]$ is then denoted as: $s - \frac{1}{L}\sum_{l=1}^{L} s_{l,[j_1,j_2,\cdots,j_n]}$, where l, L are the repetition index and the repetition number, s is the prediction accuracy of the test data, and $s_{l,[j_1,j_2,\cdots,j_n]}$ is that of the test data with features $[j_1, j_2, \cdots, j_n]$ being randomly shuffled. In this equation, the drop of $s_{l,[j_1,j_2,\cdots,j_n]}$ is indicative of how much the ML model depends on the feature subset.

Fig. 7 shows permutation importance of the subsets of LAN RTT, WAN RTT, RET, and LAN RES related features. The result indicates that RTT values play a more important role, while LAN RTT is the most crucial feature affecting the user QoE. Fig. 8 shows permutation importance of the subsets of each time and date range. In this figure, each subset is named as "(date)_(time)". The result shows that the network performance at daytime (7am–7pm) in midweek affects user experience more than others.

V. CONCLUSION

In this paper, we proposed the feature selection method for AI/ML-based QoE monitoring of broadband users. The

proposed approach pre-processes the dataset to extract a set of informative features from 8 basic protocol indicators exploiting domain knowledge of data networking as well as the histogram analysis results. The simulation results show a prediction accuracy of 78% on the benchmark ITU ML5G-PS-012 dataset, improving 11% over the state-of-the-art result. The algorithm efficiency was also evaluated in terms of run time and a model size, and the results show that the proposed method is the most efficient among those whose classification accuracy exceed 60%. Our grouped permutation feature importance study shows that the local area network round trip time (LAN RTT) value during daytime and midweek plays the most crucial role affecting the user QoE. In future works, feature extraction method will be improved. We manually decided the threshold value to split the data, but adjusting threshold automatically inspired by wrapper method and autoencoder improves the accuracy. In addition, more indicators (e.g., throughput, channel quality) as input data will give higher accuracy.

ACKNOWLEDGMENT

The first author would like to express sincere gratitude to M. Hamaminato, T. Ninomiya, K. Yokoo, and M. Ogawa of Fujitsu LIMITED for supporting his research visit to the 5G/6GIC Innovation Centre, The University of Surrey.

This work was partially funded by the 5G and 6G Innovation Centre, University of Surrey, and partially by the UK national TUDOR project.

REFERENCES

[1] M. Seufert, S. Wassermann, and P. Casas, "Considering user behavior in the quality of experience cycle: Towards proactive QoE-aware traffic management," IEEE Commun. Lett., vol. 23, no. 7, pp. 1145–1148, Jul. 2019.

[2] SANDVINE, "5G Service Intelligence Engine (NWDAF) for Core, Cloud, and Edge Networks," white paper, Sep. 2021.

[3] S. Kumar, N. Wang, Y. Rahulan and B. Evans, "QoE-Aware Video Streaming over Integrated Space and Terrestrial 5G Networks," in IEEE Network, vol. 35, no. 4, pp. 95–101, July/August 2021, doi: 10.1109/MNET.011.2100082.

[4] S. Schwarzmannm C. C. Marquezan, M. Bosk, H. Liu, R. Trivisonno, T. Zinner, "Estimating Video Streaming QoE in the 5G Architecture Using Machine Learning," Internet-QoE, pp.7–12, Oct. 2019.

[5] Y. B. Youssef, M. Afif, R. Ksantini, and S. Tabbane, "A novel online QoE prediction model based on multiclass incremental support vector machine," in Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl. (AINA), May 2018, pp. 334–341.

[6] F. Laiche, A. B. Letaifa, I. Elloumi, and T. Aguili, "When machine learning algorithms meet user engagement parameters to predict video QoE," Wireless Pers. Commun., vol. 116, no. 3, pp. 2723–2741, Sep. 2020.

[7] P. Athanasoulis, E. Christakis, K. Konstantoudakis, P. Drakoulis, S. Rizou, A. Weit, A. Doumanoglou, N. Zioulis, and D. Zarpalas, "Optimizing QoE and Cost in a 3D Immersive Media Platform: A Reinforcement Learning Approach," in Proc. MMEDIA, Lisbon, Portugal, Feb. 2020, pp. 1–6.

[8] M. Suznjevic, L. Skorin-Kapov, A. Cerekovic, and M. Matijasevic, "How to measure and model QoE for networked games?," Multimedia Syst., vol. 25, no. 4, pp. 395–420, May 2019.

[9] 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," in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2007–2023, Dec. 2020, doi: 10.1109/TNSM.2020.3036497.

[10] N. Eswara, S. Ashique, A. Panchbhai, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "Streaming video QoE modeling and prediction: A long short-term memory approach," IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 3, pp. 661–673, Mar. 2020.

[11] Q. Meng, K. Wang, B. Liu, T. Miyazaki and X. He, "QoE-Based Big Data Analysis with Deep Learning in Pervasive Edge Environment," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422106.

[12] L. Liu, H. Hu, Y. Luo, and Y. Wen, "When wireless video streaming meets AI: A deep learning approach," IEEE Wireless Commun., vol. 27, no. 2, pp. 127–133, Apr. 2020.

[13] X. Tao, Y. Duan, M. Xu, Z. Meng and J. Lu, "Learning QoE of Mobile Video Transmission With Deep Neural Network: A Data-Driven Approach," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1337–1348, June 2019, doi: 10.1109/JSAC.2019.2904359.

[14] H. Martinez, M. C. Q. Farias, A. Hines, "NAViDAd: A No-Reference Audio-Visual Quality Metric Based on a Deep Autoencoder," 27th EUSIPCO, Sep. 2019.

[15] S. Wang, S. Bi and Y. -J. A. Zhang, "Deep Reinforcement Learning With Communication Transformer for Adaptive Live Streaming in Wireless Edge Networks," in IEEE Journal on Selected Areas in Communications, vol. 40, no. 1, pp. 308–322, Jan. 2022, doi: 10.1109/JSAC.2021.3126062.

[16] Online, "ML5G-PS-012: Classification of Home Network Users to Improve User Experience," https://challenge.aiforgood.itu.int/match/matchitem/73, 2022.

[17] Online, "PS-012.3: Spears-9," https://github.com/ITU-AI-ML-in-5G-Challenge/ML5G-PS-012-Team-Spears-9-Classification-of-Home-Network-Users-to-Improve-User-Experience, 2022.

[18] Online, "PS-012.2: Xdding (jygaced)," https://github.com/ITU-AI-ML-in-5G-Challenge/ML5G-PS-012-Xdding, 2022.

[19] Online, "PS-012.1: Data Rangers," https://github.com/ITU-AI-ML-in-5G-Challenge/Team-DataRangers-home-network-users-experience-classification, 2022.

[20] E. Zdravevski, P. Lameski, R. Mingov, A. Kulakov and D. Gjorgjevikj, "Robust histogram-based feature engineering of time series data," 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, 2015, pp. 381–388, doi: 10.15439/2015F420.

[21] W. S. Saif, T. Alshawi, M. A. Esmail, A. Ragheb and S. Alshebeili, "Separability of Histogram Based Features for Optical Performance Monitoring: An Investigation Using t-SNE Technique," in IEEE Photonics Journal, vol. 11, no. 3, pp. 1–12, June 2019, Art no. 7203012, doi: 10.1109/JPHOT.2019.2913687.

[22] K. R. Aravind, P. Raja, K. V. Mukesh, R. Aniirudh, R. Ashiwin and C. Szczepanski, "Disease classification in maize crop using bag of features and multiclass support vector machine," 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2018, pp. 1191–1196, doi: 10.1109/ICISC.2018.8398993.

[23] Online, "TSFresh," https://tsfresh.com, 2021.

[24] Lucas Plagwitz, Alexander Brenner, Michael Fujarski, and Julian Varghese, "Supporting AI-Explainability by Analyzing Feature Subsets in a Machine Learning Model," Studies in Health Technology and Informatics, Volume 294: Challenges of Trustable AI and Added-Value on Health, 2022. doi:10.3233/SHTI220406.