Enhancing Machine Learning based QoE Prediction by Ensemble Models

Pedro Casas†, Michael Seufert†, Nikolas Wehner†, Anika Schwind*, Florian Wamser*

†AIT – Austrian Institute of Technology, Vienna, Austria
{pedro.casas, michael.seufert.fl, nikolas.wehner.fl}@ait.ac.at

*University of Würzburg, Institute of Computer Science, Würzburg, Germany
{forename.surname}@informatik.uni-wuerzburg.de

***Abstract*—The number of smartphones connected to wireless networks and the volume of wireless network traffic generated by such devices have dramatically increased in the last few years, making it more challenging to tackle wireless network monitoring applications. The high-dimensionality of network data provided by current smartphone devices opens the door to the massive application of machine learning approaches to improve different wireless networking applications. In this paper we study the specific problem of Quality of Experience (QoE) prediction for popular smartphone apps, using machine learning models and in-smartphone measurements. We evaluate and compare different models for the analysis of smartphone generated data, including single models as well as machine learning ensembles such as bagging, boosting and stacking. Results suggest that, while decision-tree based models are the most accurate single models to predict QoE, ensemble learning models, and in particular stacking ones, are capable to significantly increase accuracy prediction and overall classification performance.**

I. Introduction

The ever-increasing number of mobile devices connected to wireless networks is heavily modifying the traffic observed in these networks. The traffic volumes and big data generated by smartphones opens the door to novel data-driven network management paradigms, in which network operation can be dramatically enhanced and simplified by the automatic analysis of network measurements. The high-volume and high-dimensionality of mobile network data provided by current network measurement systems calls for the massive application of machine learning approaches to improve data-driven networking.

There is however a major challenge in applying machine learning models at large-scale for handling network measurements; selecting the best machine learning model for a specific problem is a complex task - it is commonly accepted that there is no silver bullet for addressing different problems simultaneously. Indeed, even if multiple models could be very well suited to a particular problem, it may be very difficult to find one which performs optimally for different data distributions and statistical mixes.

Partly funded by EU project MONROE (H2020-2014-ICT-644399, open call project Mobi-QoE) and by WWTF project Big-DAMA (WWTF-ICT15-129).

Deep-learning models are today widely used in multiple signal processing problems, particularly in image processing, where they have shown an outstanding performance. However, neural-networks based models, and particularly deep-learning models, have an inherent problem linked to model visibility and interpretation: a deep-learning model is a black-box which can automatically perform feature selection from input raw data and provide highly accurate predictions, but it is very difficult to understand their functioning. Indeed, it becomes very challenging to understand the reasons of a particular classification result, and in particular to understand the input features leading to such a result, as input features derived directly from a deep neural network architecture can be in general meaningless to a domain expert. This is one of the reasons why their application to networking problems is so far quite limited. In addition, deep-learning models are highly data-eager and training them is extremely costly in terms of computational power, which might term them unsuitable for different networking problems which require periodical re-training or where labeled data are difficult to get.

In this paper we pose ourselves a simple question: which type of machine learning model should be generally used in the analysis of wireless network measurements? Intuition suggests that rule-based models could be in principle a good match for wireless network analytics, as network protocols are highly structured and operate in a rule basis. We therefore present a comparative analysis of different machine learning models, applied to the specific problem of Quality of Experience prediction in smartphones.

We consider standard and well known machine learning models, which shall ease the interpretation of results and make them more applicable to common networking practitioners. These models include decision-trees - single trees and random forests, naïve bayes models, neural networks, support vector machines and nearest neighbor search models. We additionally consider collaborative- or ensemble-learning models, covering the three basic paradigms available in the ensemble-learning domain: bagging, boosting and stacking. Rather than finding the best model to explain the data, ensemble-learning methods build a set of models and then decide between them with some combinatorial approach, seeking complementarity among models. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. In

principle, if no single model covers the true prediction behind the data, an ensemble can give a better approximation of that oracle, true prediction model. An ensemble of models also exhibits higher robustness with respect to uncertainties in training data, which is highly beneficial for generalization of results.

This paper builds on top of our recent work on machine learning for QoE prediction [2], where we study the performance of single machine learing models for QoE prediction in smartphones, and on ensemble-learning models for network analytics [6], where we explore the application of ensemble-learning techniques to network security and anomaly detection problems. The reminder of the paper is organized as follows. Sec. II presents a partial overview on the related work. In Sec. III we briefly describe the evaluated machine learning models. Sec. IV describes the evaluated QoE prediction problem and characterizes the corresponding dataset. Sec. V presents the experimental results of the study, benchmarking the accuracy of the proposed models. Finally, Sec. VI concludes the paper.

II. State of the Art

This paper deals with machine learning for QoE monitoring and analysis, specially on cellular networks and mobile devices; there is an assorted list of tools to measure network performance: some examples are Mobilyzer [15] and Netalyzr [16]. In [17] we introduced YoMoApp, an app to passively monitor YouTube QoE-related features in smartphones. In [18] we describe an on-line monitoring system for YouTube QoE in cellular networks using in-network measurements only. QoE Doctor [13] measures mobile app QoE, using active measurements at both application and network layers. Close to our work, authors in [14] propose an approach to evaluate mobile apps QoE using passive in-network and in-device measurements.

In [2], [3] we study the same dataset analyzed in this work, either using single machine learning models [2] or basic statistical data analysis techniques [3].

A recent trend in QoE-based network monitoring considers the analysis of encrypted network traffic. Indeed, there has been a sort of re-vival for low-level network-based QoE monitoring rather than relying on application-layer metrics. As an example of such approaches based on encrypted network-layer measurements, authors in [4] evaluate a machine learning-based architecture that estimates YouTube QoE from features derived from packet sizes, inter-arrival times, and throughput. A similar approach is presented in [5], where authors rely on real cellular network measurements to predict typical QoE indicators for streaming services (e.g., played resolutions, stalling events), based on features such as round-trip times, packet loss and chunk sizes. Here, the authors also used machine learning as a promising technique for large-scale quality monitoring and prediction.

III. Machine Learning Models

In the context of supervised learning there are several approaches for predictive model training based on labeled data. The performance of a particular algorithm or predictor depends on how well it can assimilate the existing information to approximate the oracle predictor, i.e. the ideal optimal predictor defined by the true data distribution. However, knowing a priori which algorithm will be the best suited for a given problem is almost impossible in practice. One could say that each algorithm learns a different set of aspects of reality from the training datasets, and then their respective prediction capability also differs between problems.

In the study we compare six standard machine learning models previously used in the literature for the analysis of network measurements, including: (i) decision-trees (CART), (ii) Naïve Bayes (NB), (iii) Multi-Layer Perceptron (MLP) Neural Networks, (iv) Support Vector Machines (SVM),(v) Random Forest (RF) and (vi) Nearest Neighbors (k-NN). To improve prediction results, we additionally study more complex models, following the ensemble learning paradigm. The ensemble learning theory permits to combine multiple single models to form a (hopefully) better one. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. There are multiple approaches to ensemble learning, including bagging [8], boosting [9], and stacking [11]. All three are so-called "meta-algorithms", defining different approaches to combine several machine learning techniques into one single predictive model referred to as *meta learner*, to either decrease the variance (bagging), decrease the bias (boosting) or improve the predictive performance (stacking). We briefly describe all these approaches next.

A. Decision Trees

Classification And Regression Trees (CART) [12] define a classification technique based on a tree graph, where inner nodes correspond to a condition on a feature and leaves are the outcome (i.e., the class). A CART represents a very popular classification algorithm due to its simplicity (it can be easily converted into a rule-based classification system) and readability (it can be graphically represented). The training follows a top-down greedy algorithm that works by iteratively splitting the nodes, using normally an information gain based metric as optimization criterion.

B. Naïve Bayes

Naïve Bayes (NB) is a very simple classifier based on Bayesian statistics [12]. Despite its simplicity, it is widely used as it is very efficient in a number of scenarios, especially in high-dimensional datasets. It works by assuming that features are mutually independent, which is not true in most cases, hence the adjective naïve. This assumption allows for an easy calculation of the class-conditional probabilities, using maximum likelihood estimation.

C. Neural Networks

Multi-Layer Perceptron (MLP) is an artificial neural network composed of multiple layers of neurons, each of them generally represented by a non-linear function [12]. The layers are fully connected in a feed-forward scheme. Each neuron employs an activation function that maps the weighted inputs to the output that is passed to the following layer. The weights, originally set to random values, are iteratively adjusted during the training phase, using a well-known approach referred to as back-propagation.

D. Support Vector Machines

Support Vector Machines (SVM) are non-probabilistic binary classifiers [12]. SVM is considered one of the most powerful supervised classification algorithm. It works by representing each feature vector in a multidimensional space and trying to find a linear separation (i.e., an hyperplane) for the classes. In some cases, however, a linear separation of the space is not possible, hence it uses the so-called kernel trick, which implicitly increases the dimensionality of the space, resulting in an easier separation in a much higher dimensional space, due to the increased sparsity.

E. Random Forest

Random Forest (RF) is an ensemble technique based on multiple instances of decision trees, each one based on a different part of the training set, randomly selected. These instances are called bootstrapped samples. The final outcome is generally decided by majority voting among all the bootstrapped samples.

F. k Nearest Neighbors

The k-Nearest Neighbors algorithm (k-NN) is a non-parametric approach used for either classification or regression. In both cases, the input consists of the k closest training examples in the feature space. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

G. Bagging and Boosting Algorithms

Bagging - for Bootstrap Aggregation, decreases the variance of the prediction model by generating additional training data from the original dataset. Bagging trains each model in the ensemble using a randomly drawn subset of the training set, and each model in the ensemble is then combined in an equal-weight majority voting scheme. Increasing the training data size using a single input dataset does not improve the prediction accuracy, but narrows the prediction variance by strongly tuning the outcome.

Boosting involves incrementally building an ensemble by training each new model instance based on the performance of the previous model. Boosting is a two-steps approach, where one first uses subsets of the original data to produce multiple models, and then *boosts* their performance by combining them, also using majority voting. Different from bagging, boosting subset creation is not random but depends upon the performance of the previous models, and every new subsets contain the misclassified instances by previous models.

We take decision-tree based models for both bagging and boosting, which is a very common approach. In the case of bagging, we consider a Bagging Tree model. We take an AdaBoost [10] Tree model for boosting, which uses decision trees as first level learners. AdaBoost (short for Adaptive Boosting) trains subsequent models in favor of those instances misclassified by previous ones. AdaBoost is sensitive to noisy data and outliers, but in general, it can be less susceptible to over-fitting.

Table I: Input features for QoE prediction.

KPI Name	KPI Description (U – reported by user)
MOS	overall user experience **(U)**
ISP	cellular network operator
RAT	radio access technology
SIG	avg. signal strength
TH_{max}	max. session downlink flow throughput
TH_{avg}	avg. session downlink flow throughput
DUR	session duration
VOL	session volume
$FLOW_{ratio}$	ratio (# flows up)/(# flows down)
CELL	cell id
LOC	user location context **(U)**

H. Stacking

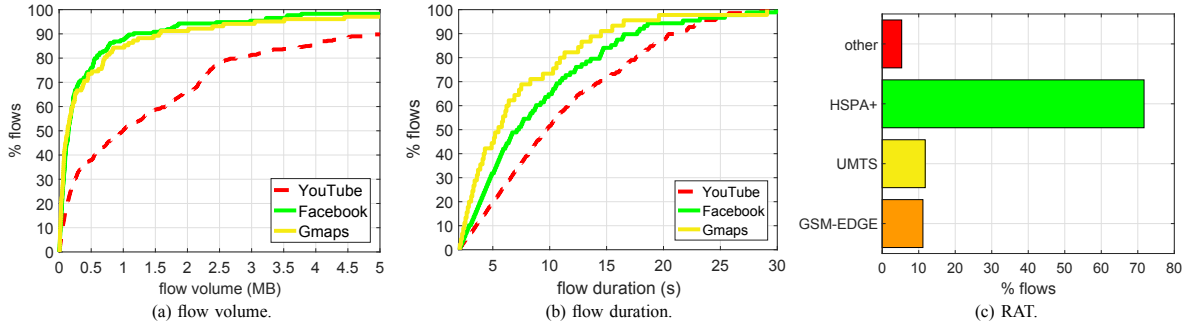
While bagging and boosting generally use the same type of model in all the different training steps (e.g., decision trees), stacking aims at exploring the input data space through *base models* of different type. Stacking is the ensemble learning model that really makes use of a meta learner, which uses the output of the base learners as input for prediction. The point of stacking is to explore a space through the different properties of different models, each of them capable to learn some part of the problem, but not the whole space. The meta leaner is said to be stacked on the top of the other based models, hence the name.

General ensemble learning approaches might be prone to over-fitting the data. In [1] a simple stacking learning algorithm named *Super Learner* is proposed as a possible solution for this over-fitting limitation. It proposes a method to minimize the over-fitting likelihood using a variant of cross-validation. In addition, the Super Learner provides performance bounds, as it performs asymptotically as good as the best available single hypothesis predictor, for each predicted pattern.

In the study, we consider two flavors of Super Learner for stacking, using five of the six aforementioned single models as base learners (i.e., CART, NB, MLP, SVM and k-NN): a simple majority voting based algorithm (Stacking MV), where the output of the base learners are equally weighted to decide on the final output, and GML (Generic Machine Learning), which basically computes weights in an exponential fashion, using the classification accuracy of each base learner. This approach permits to reduce the influence of low accuracy base predictors.

IV. QoE Prediction from Smartphone Data

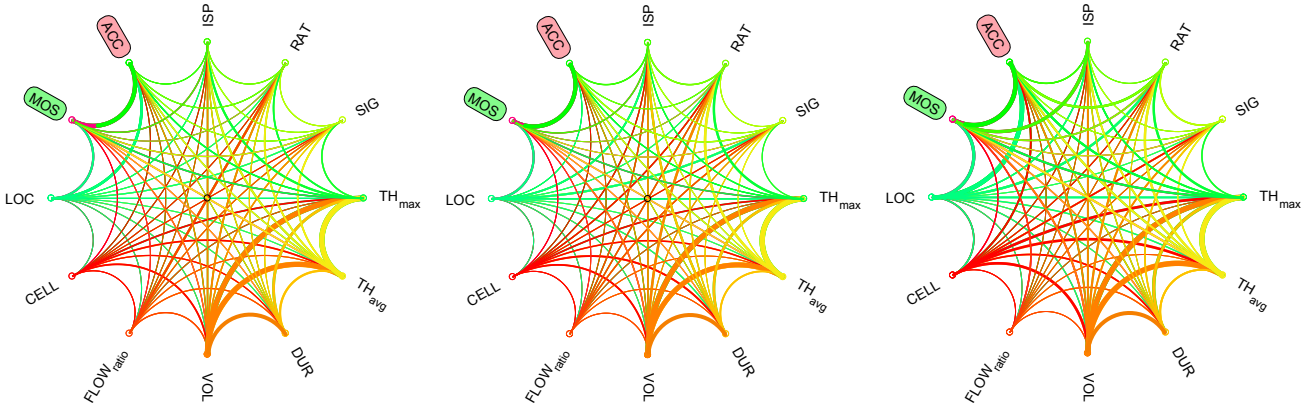
For the sake of QoE prediction in cellular traffic, we use network and QoE measurements collected in a user field trial taking place in 2015 and detailed in [2], where 30 users equipped with their own devices connected to their preferred cellular operators evaluated three apps as part of their normal daily Internet activity during two weeks: YouTube (watching short videos); Facebook (timeline and photo-album browsing), and Gmaps (satellite maps browsing). QoE feedback was reported for each session through a customized QoE crowdsourcing app, according to a discrete, 5-levels ACR Mean



(a) flow volume. (b) flow duration. (c) RAT.

(d) avg. downlink flow throughput. (e) max. downlink flow throughput. (f) signal strength.

Figure 1: Empirical distribution of multiple input QoS features measured at the network layer.



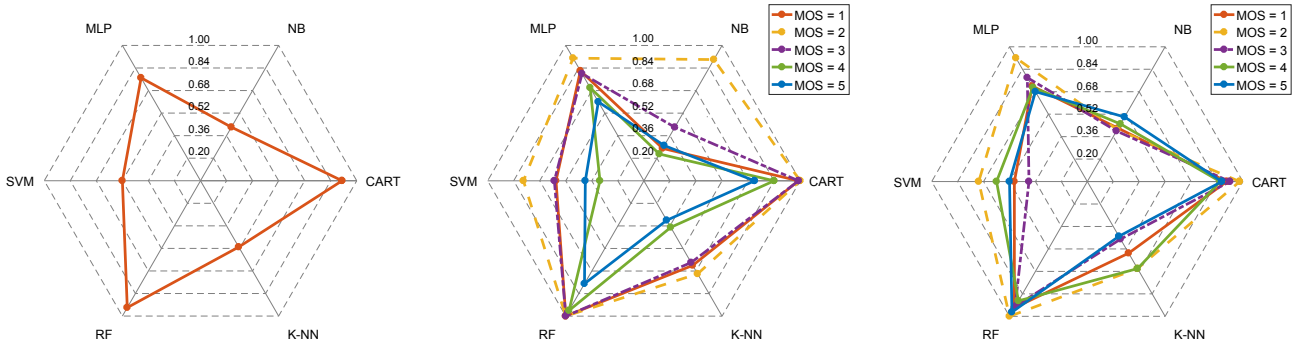
(a) YouTube Feature Correlation. (b) Facebook Feature Correlation. (c) Gmaps Feature Correlation.

Figure 2: Inter features linear correlation for each app. The thicker the edge connecting two nodes, the higher the correlation between the corresponding features.

Opinion Score (MOS) scale, ranging from "bad" (i.e., MOS = 1) to "excellent" (i.e., MOS = 5). Users additionally report the acceptablity of the service as a binary feedback, stating whether he would continue using the application under the corresponding conditions or not (note than in this paper we only focus on the MOS predictions).

In addition, each device has a passive flow-level traffic monitor which records flow-level network traffic statistics, associating flows to apps generating them. The 10 different session-based KPIs in Tab. I are derived from the flow-based measurements, which are then synchronized to the QoE feedbacks (MOS scores) using time stamps. The KPIs include features such as average and maximum flow throughput per session, flow size, duration, average signal strength, RAT, ISP, locations, etc. The prediction problem consists in predicting the correct MOS score value (5-classes classification problem), using the session-based KPIs as input. Full details on the dataset are available in [2].

Fig. 1 depicts the empirical distributions of some of the key features measured at the network layer. Figs. 1(a,b) show that, as expected, YouTube flows are way larger than Facebook and Gmaps ones, but their durations are rather similar, which could be linked to the specific characteristics of the applications themselves. There is also a clear difference in the achieved downlink flow throughputs, as reflected in Figs. 1(d,e). Fig. 1(c) shows that the biggest share of flows are transmitted over fast HSPA+ and UMTS connections, but a non-negligible fraction of flows are carried over EDGE, potentially resulting



(a) Global Accuracy. (b) Recall. (c) Precision.

Figure 3: Global accuracy, recall and precision achieved by the base ML models for QoE prediction.

in poor QoE. Finally, Fig. 1(f) shows that about 70% of the flows are transmitted with a signal strength higher than -90 dBm, which is normally considered as an excellent coverage threshold, whereas only 5% of the flows correspond to signal strength below -105 dBm, which is considered as bad coverage.

To conclude, and to better understand the relations between the considered features and the prediction target - MOS scores, Fig. 2 shows the absolute magnitude of the inter-features linear correlation values as well as the linear correlations to both MOS and acceptability scores, considering circular plots. The thicker the edge connecting two nodes, the higher the correlation between the corresponding features.

Not surprisingly, the most relevant features are the average and the maximum session throughput. Bulky sessions (i.e., bigger VOL) are generally better perceived and translate into higher acceptability, which might be linked to the specifics of the contents being consumed - e.g., HD video or better resolution maps, and the better usage of the available bandwidth. As expected, signal strength is also positively correlated to QoE, as the higher the signal strength, the better the connection performance - e.g., higher throughput and lower latency. Interestingly, longer YouTube sessions experience a worse QoE; a deeper analysis of longer YouTube sessions shows that many of them are rather small, suggesting the occurrence of stalling. ISP also appears as a relevant feature, and specially for acceptability, suggesting that service quality is different for the different operators on the field trial, as well as the corresponding user expectations. Recall that participants had their own data contracts, so a better look into the characteristics of these contracts in terms of agreed performance and cost would shed light on this. Location is also relevant, and in particular for the case of Gmaps, where correlations between QoE and LOC are much higher.

V. Evaluation and Discussion

In this section we evaluate and compare the performance achieved by the presented ML models. For the sake of training and testing, we consider 10-fold cross validation in all the results presented in this section. Parameters on each different algorithm are calibrated based on best-performance, grid search tests. We start by comparing the performance achieved by the six base learning models, and then present a full comparison including also the ensemble-learning approaches.

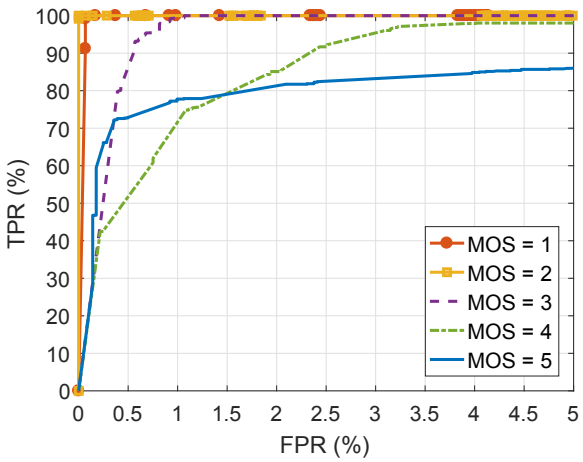


Figure 4: Performance of RF model - ROC curves.

A. Single Base Learning Models

Fig. 3 reports the prediction performance achieved by the six single base learning models for QoE prediction. Performance is measured in terms of global classification accuracy (i.e., correctly classified instances), as well as per class recall and precision. Fig. 3 clearly shows that decision-tree based models, and in particular RF ones, represent by far the most accurate approach, for all the different quality levels. Still, as we see more in detail next, predictions are not flaw-less and some important mis-classifications occur.

To dig deeper into the RF out-performance, Fig. 4 depicts the ROC curves and the corresponding confusion matrices obtained with the RF model. Fig. 4 shows that the RF model is very accurate to correctly spot out bad quality sessions (i.e., MOS = 1, 2 and 3), but is less accurate to correctly predict higher quality ones; in particular, as depicted in Fig. 5, excellent quality sessions (i.e., MOS = 5) are often misclassified as good ones (MOS = 4), and as average ones (i.e., MOS = 3) to a lesser extent.

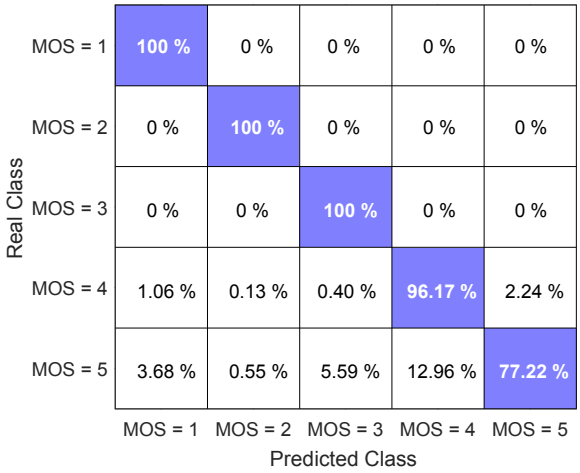


Figure 5: Performance of RF model - Confusion matrix.

B. *Including Ensemble Learning Models*

To conclude with the study, we now include the ensemble learning models within the analysis. Tab. II reports the results obtained with all the models, using the Area Under the ROC Curve (AUC) as performance metric. The machine learning community most often uses the ROC AUC statistic for model comparison, which is simple and informative. Tab. II reveals again that predicting excellent QoE sessions (i.e., MOS = 5) is more challenging than for the rest of the quality levels. The CART and random forest models alone provide already very good results, as also shown in Fig. 3. Still, the GML model is capable to boost the prediction of excellent QoE w.r.t. both CART and random forest by at least 5% to 10%, suggesting a better fit for this scenario. Indeed, results highlight the advantages of the stacking models, and in particular, the GML model.

VI. Concluding Remarks

We have demonstrated the outstanding performance of decision trees for the analysis of smartphone measurements in predicting the QoE of the end users. Indeed, decision-tree-based models provide in general better results in terms of accuracy and prediction than other models, with a much smaller computational overhead for decision trees as compared to other models based on neural networks or support vector machines. Decision-tree based models represent therefore a very appealing machine learning model for QoE prediction, not only because of their high accuracy and low computational cost, but also due to a series of embedded properties, such as model visibility, robustness to input noise, etc.

We have also shown the advantages of ensemble learning techniques to improve prediction accuracy, and particularly of the stacking GML approach. We found that not only the GML based model has the ability to perform as well as the best available single base level learner, but often achieves better results. This includes also the case of both bagging and boosting models, which are also outperformed by the stacking models. Performance improvements are higher in scenarios where the performance of the base predictors are relatively low; when first learners performance is already high, there is little room for improvement. We believe that this study would enable a broader application of machine learning models to the study of QoE prediction in network monitoring systems.

Table II: ROC AUC for QoE prediction.

MOS	1	2	3	4	5
CART	0.972	0.974	0.963	0.972	0.900
Naïve Bayes	0.766	0.874	0.714	0.707	0.703
MLP	0.916	0.951	0.918	0.852	0.798
SVM	0.812	0.928	0.742	0.717	0.734
Random Forest	**0.992**	0.989	0.987	0.988	0.960
k-NN	0.849	0.917	0.765	0.756	0.657
Bagging Tree	0.971	0.977	0.996	0.982	0.955
AdaBoost Tree	0.972	0.978	**0.997**	0.992	0.973
Stacking MV	**0.992**	0.965	0.984	0.990	0.971
GML	**0.992**	**0.996**	**0.997**	**0.995**	**0.985**

References

[1] M. Van der Laan, et al., "Super learner", in Statistical applications in genetics and molecular biology, vol. 6(1), 2007.

[2] P. Casas et al., "Predicting QoE in Cellular Networks using Machine Learning and in-Smartphone Measurements", in *QoMEX*, 2017.

[3] P. Casas et al., "Next to You: Monitoring Quality of Experience in Cellular Networks from the End-devices", IEEE Trans. on Net. and Service Mgmt., vol. 13(2), 2016.

[4] I. Orsolic et al., "A machine learning approach to classifying YouTube QoE based on encrypted network traffic", Multimedia Tools and Applications, vol. 76(21), 2017.

[5] G. Dimopoulos et al., "Measuring Video QoE from Encrypted Traffic", in *ACM IMC*, 2016.

[6] J. Vanerio, et al., "Ensemble-learning Approaches for Network Security and Anomaly Detection," in *SIGCOMM Big-DAMA workshop*, 2017.

[7] T. Dietterich, "Ensemble learning", The handbook of brain theory and neural networks, vol. 2, MIT Press: Cambridge, MA, 2002.

[8] L. Breiman, "Bagging Predictors", Machine Learning, vol. 24(2), 1996.

[9] Y. Freund, et al., "Experiments with a New Boosting Algorithm", in *ICML*, 1996.

[10] Y. Freund, et al., "A decision-theoretic generalization of on-line learning and an application to boosting", Jour. of Comp. and Sys. Sciences, vol. 55(1), 1997.

[11] D. Wolpert, "Stacked Generalization", Neural Nets., vol. 5(2), 1992.

[12] T. Nguyen et al., "A Survey of Techniques for Internet Traffic Classification using Machine Learning", IEEE Comm. Surv. & Tut., vol. 10(4), 2008.

[13] Q. A. Chen, et al., "QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis," in *ACM IMC*, 2014.

[14] V. Aggarwal, et al., "Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements," in *ACM HotMobile*, 2014.

[15] A. Nikravesh, et al., "Mobilyzer: An open platform for controllable mobile network measurements," in *ACM MobiSys*, 2015.

[16] C. Kreibich, et al., "Netalyzr: Illuminating the edge network," in *ACM IMC*, 2010.

[17] F. Wamser, et al., "Understanding YouTube QoE in Cellular Networks with YoMoApp – a QoE Monitoring Tool for YouTube Mobile," in *ACM MOBICOM*, 2015.

[18] P. Casas, et al., "YOUQMON: A System for On-line Monitoring of YouTube QoE in Operational 3G Networks," ACM SIGMETRICS Performance Evaluation Review, vol. 41, 2013.