QoE-driven Anomaly Detection in Self-Organizing Mobile Networks using Machine Learning

Chetana V. Murudkar* and Richard D. Gitlin, *Life Fellow, IEEE*
Innovation in Wireless Information Networking Lab (iWINLAB)
Department of Electrical Engineering,
University of South Florida
Tampa, Florida 33620, USA.
Email: cvm1@mail.usf.edu, richgitlin@usf.edu
**Sprint Corp.,* USA. Email: chetana.v.murudkar@sprint.com

***Abstract*— Current procedures for anomaly detection in self-organizing mobile communication networks use network-centric approaches to identify dysfunctional serving nodes. In this paper, a user-centric approach and a novel methodology for anomaly detection is proposed, where the Quality of Experience (QoE) metric is used to evaluate the end-user experience. The system model demonstrates how dysfunctional serving eNodeBs are successfully detected by implementing a parametric QoE model using machine learning for prediction of user QoE in a network scenario created by the *ns-3* network simulator. This approach can play a vital role in the future ultra-dense and green mobile communication networks that are expected to be both self-organizing and self-healing.**

***Keywords*— Machine learning, ns-3, QoE, SON**

I. Introduction

With the exponential growth in mobile traffic data, network operators are in a dire need of a technology that can meet the demanding requirements of ever-increasing network usage marked by a radical change in user behavior triggered by the proliferation of bandwidth-hungry applications. As the emerging bandwidth-intensive technologies such as 5G, Internet of Things (IoT), Virtual Reality begin to be deployed, the network operators need to ensure that networks are intelligent, scalable and robust enough to provide the excellent user experience that consumers expect. One promising solution to address these concerns is the deployment of self-organizing networks (SON).

Self-organizing mobile networks can be divided into three primary categories: self-configuration, self-optimization, and self-healing. These are commonly denoted as self-x functions. The self-healing function is activated whenever a fault or failure occurs. Its objective is to continuously monitor the system in order to ensure a fast and seamless recovery by detecting the failure events, diagnosing them and triggering the appropriate compensation mechanisms, so that the network can return to proper functioning [1]. One of the salient functions of self-healing is to be able to correctly detect dysfunctional nodes or sites that cause outages and degradation in the network. Currently, alarm monitoring, routine checks on configuration parameters and counters, analyzing Key Performance Indicators, conducting drive tests, mobile tracing, and keeping track of customer complaints are some of the widely used detection methods that network engineers follow to detect dysfunctional nodes [2]. These processes can be time-consuming and may not always be cost-effective and resource-efficient.

The structure of this paper is as follows. In Section II, the user-centric approach used for anomaly detection in this research is described. The system model is explained in detail in Section III. Section IV presents the simulation results and observations. The paper ends with the concluding remarks in Section V.

II. User-Centric Approach

Customer experience is generally regarded as the most important criteria to drive revenues for any network operator or service provider. A good and satisfying experience leads the users to spend more time on the network, which in turn drives demand and increase revenues. Following this strategy, cellular network operators have started to use a user-centric approach to better understand the end-user perception of the quality of a provided service. The metric for this purpose is referred to as Quality of Experience. Quality of Experience (QoE) is defined by the Telecommunication Standardization Sector of International Telecommunication Union (ITU-T) as "the overall acceptability of an application or service, as perceived subjectively by the end-user" [3]. In other words, QoE describes the degree of the end-user's "delight or annoyance" while using a product or service. There are different types of approaches that can be used for QoE assessment. These approaches can be classified into subjective tests, objective tests and hybrid tests methods. There are various types of evaluation models based on these approaches proposed for QoE estimation in the literature. Parametric QoE estimation models are currently the most popular candidates for quantifying QoE levels in an indirect and user-transparent way in mobile networks. Parametric models use network parameters and metrics for QoE estimation [4]-[5]. Parametric QoE models are derived by performing subjective experiments that may include laboratory tests or crowdsourcing and by performing statistical analysis on the results. The derived models may then be used to generate formulas which can be used to compute QoE given specific input parameters [6].

The proposed method in this research is to use a parametric model to predict the QoE of users in the network and to use their QoE scores to detect serving eNodeBs that are dysfunctional, thus demonstrating a user-centric approach for the use case of anomaly detection in self-healing and self-organizing networks. This is a resource-efficient method as it not only provides information about the users' experience, which is extremely crucial for a network operator, but also uses this information to identify nodes that are not functioning well enough to serve the users in their vicinity. This way, the network operators can prioritize their recovery operations on nodes that need immediate attention versus the ones that may still be manageable given their QoE scores, and thus avoid overengineering.

III. System Model

The proposed method is demonstrated by using the network simulator *ns-3* [7] to simulate a network and applying machine learning using Python to implement QoE score prediction using a parametric QoE model. The QoE scores are then used for anomaly detection to identify dysfunctional eNodeBs. The process can be explained using the flowchart shown in Fig. 1.

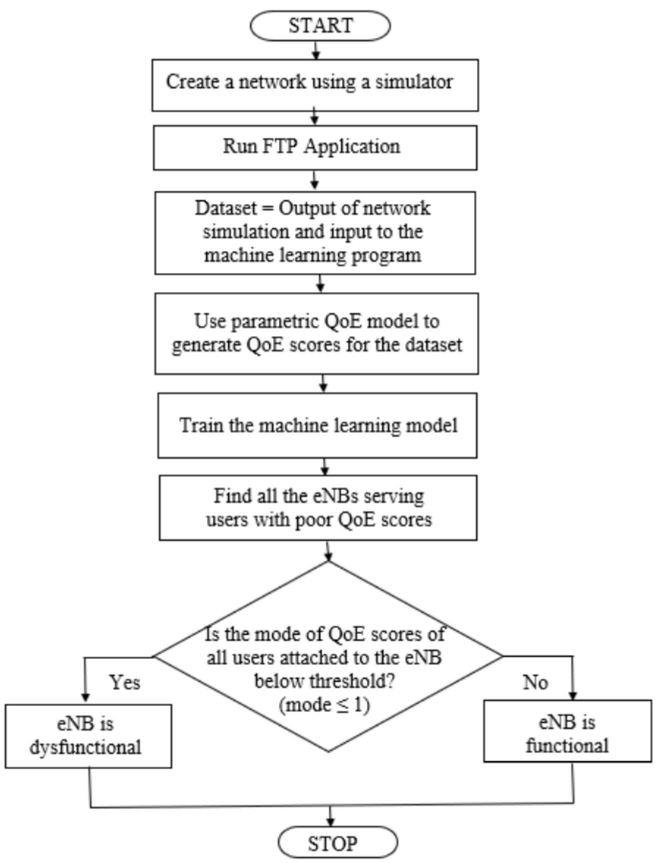

Fig. 1 Flowchart describing the system model.

The *ns-3* simulator is a discrete-event network simulator intended primarily for research and educational use. In brief, *ns-3* provides models of how packet data networks work and perform, and provides a simulation engine for users to conduct simulation experiments. Some of the reasons to use *ns-3* include the ability to perform studies that are more difficult or not possible to perform with real systems, to study system behavior in a highly controlled, reproducible environment, and to get insight into the workings of a particular network [7]. The proposed method has used the LTE-EPC Network Simulator module of *ns-3* to generate realistic data. An end-to-end network scenario is created where end users interact with a remote host that is accessed over the internet to run the most commonly used applications like file downloads and uploads. The output generated is used as the input dataset for the machine learning program.

There are multiple parametric QoE models which can be used for a reliable estimation of QoE for various types of services. One of the most commonly used application by users is the file download application. The protocol used by this application is the File Transfer Protocol (FTP). File transfer services are considered to be elastic services, whose utility function is an increasing, strictly concave, and continuously differentiable function of throughput [8]. The principal characteristic of FTP services is that there is no need for a continuous and in-sequence packet arrival. Taking into account that the delay expected by the end-user is proportional to the size of the downloaded file, the most dominant factor that affects the QoE level is the data rate. The parametric model that provides the Mean Opinion Score (MOS) for an FTP service used to determine the QoE is as follows:

$$MOS_{FTP} = \begin{cases} 1 & u < u^- \\ b_1 . \log_{10}(b_2 . u) & u^- \leq u < u^+ \\ 5 & u^+ \leq u \end{cases} \quad (1)$$

where u represents the data rate of the correctly received data. The values of b_1 and b_2 coefficients are obtained from the upper (u^+) and lower rate (u^-) expectations for the service [5], [6], [8].

The machine learning algorithm used to train the model proposed in this research is a *decision tree* algorithm. *Decision tree* algorithm falls into the category of supervised machine learning where the task of the machine learning model is to predict target values from labeled data. The input is referred to by terms such as independent variables or features. The output is referred to by terms such as dependent variables or target labels or target values.

The basic idea behind *decision tree* methods is that, based on the original data, a set of partitions are created so that the best class (in classification problems) or value (in regression problems) can be determined by creating decision rules which could be a set of if-then-else rules deduced from the data features [9]. The type of *decision tree* algorithm used in the proposed method is a *Classification and Regression Trees*

(CART) algorithm which can be explained as follows [9]: Given training vectors $\boldsymbol{x}_i \in R^n$, $i = 1,\ldots, l$ and a label vector $\boldsymbol{y} \in R^l$, a *decision tree* recursively partitions the space such that the samples with the same labels are grouped together. Let the data at node m be represented by Q. For each candidate split $\theta = (j, t_m)$ consisting of a feature j and threshold t_m, partition the data into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets. This means the data represented by Q is now divided into two subsets $Q_{left}(\theta)$ and $Q_{right}(\theta)$ that are computed using the equations given below:

$$Q_{left}(\theta) = (x, y) | \, x_j \leq t_m \tag{2}$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta) \tag{3}$$

where the division operator '\' is used to denote left division. The impurity at m is computed using an impurity function $H()$, the choice of which depends on the task being solved (classification or regression)

$$G(Q, \theta) = \frac{n_{left}}{N_m} H\left(Q_{left}(\theta)\right) + \frac{n_{right}}{N_m} H\left(Q_{right}(\theta)\right) \tag{4}$$

The parameters are selected such that they minimize the impurity

$$\theta^* = argmin_\theta G(Q, \theta) \tag{5}$$

The subsets $Q_{left}(\theta^*)$ and $Q_{right}(\theta^*)$ are recursively computed until the maximum allowable depth is reached. The maximum allowable depth is a computational choice specified to control the complexity of the tree and prevent overfitting via pruning.

Classification and Regression Trees are used for constructing prediction models from data. These models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. The recursive partitioning can be represented graphically as a *decision tree* which is easy to visualize and interpret. Classification trees are designed for dependent variables that take a finite number of unordered values, with prediction error measured in terms of misclassification cost and regression trees are designed for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values [10]. The proposed method uses the regression criteria for determining the locations for future splits such that for a node m in a region R_m with N_m observations, the common criteria used to minimize are Mean Squared Error (MSE) and Mean Absolute Error (MAE). Mean Squared Error minimizes the L2 error using mean values at terminal nodes and can be expressed as follows:

$$\bar{y}_m = \frac{1}{N_m} \sum_{i \epsilon N_m} y_i \tag{6}$$

$$H(X_m) = \frac{1}{N_m} \sum_{i \epsilon N_m} (y_i - \bar{y}_m)^2 \tag{7}$$

Mean Absolute Error minimizes the L1 error using median values at terminal nodes and can be expressed as follows:

$$\bar{y}_m = \frac{1}{N_m} \sum_{i \epsilon N_m} y_i \tag{6}$$

$$H(X_m) = \frac{1}{N_m} \sum_{i \epsilon N_m} | \, y_i - \bar{y}_m | \tag{8}$$

where X_m is the training data in node m .

To perform anomaly detection, the predicted QoE score for every user, using the machine learning model described above, is used to filter out all the users with poor QoE scores (QoE ≤ 1). The serving eNBs of such users with poor QoE undergo an individual check where it can be determined that if the mode of the QoE scores of all the users connected to that particular eNB is less than or equal to the threshold which in this case is set to 1, then the eNB is declared dysfunctional but if the mode of the QoE scores of all the users connected to a particular eNB is greater than the threshold, the eNB is declared functional.

IV. Simulation Results And Observations

The values of the primary parameters used to configure the network scenario created in the *ns-3* simulation are given in Table I.

TABLE I. NETWORK SIMULATION CONFIGURATION PARAMETERS

Parameters	**Value**
Number of users	50
Number of eNodeBs	5
eNodeB Bandwidth	20 MHz
Transmit Power of functional eNB	46 dBm
Transmit Power of dysfunctional eNB	30 dBm
Application Type	FTP

The end to end network simulation is demonstrated using NetAnim [7] and is depicted in Fig. 2 and Fig. 3.

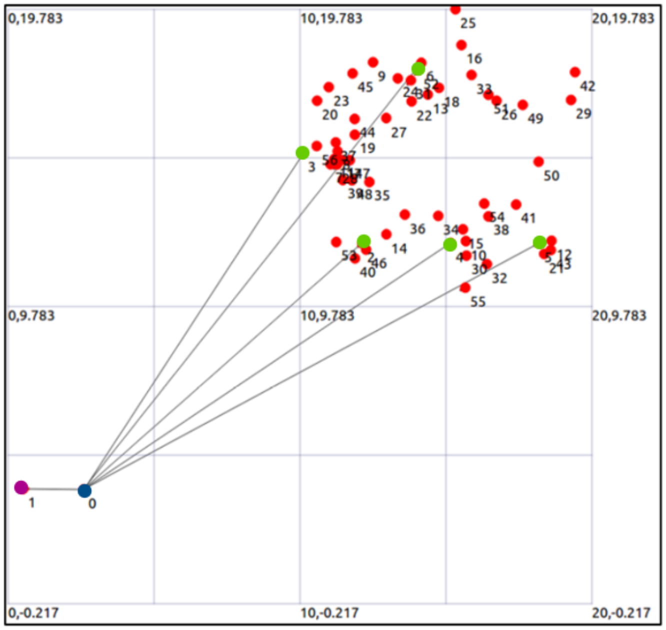

Fig. 2 NetAnim Set up – node 0 (blue) represents the packet data network gateway, node 1 (purple) represents the remote host, green nodes 2, 3, 4, 5, and 6 represent the eNBs, and red nodes 7 to 56 represent randomly distributed UEs. The figure describes the initial layout before simulation begins.

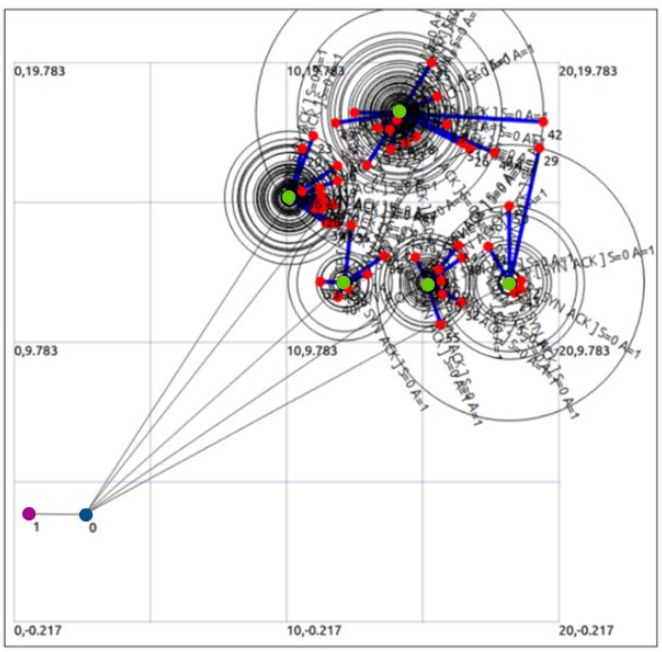

Fig. 3 NetAnim Set up – node 0 (blue) represents the packet data network gateway, node 1 (red) represents the remote host, green nodes 2, 3, 4, 5, and 6 represent the eNBs, and red nodes 7 to 56 represent randomly distributed UEs. The figure describes an end-to-end network scenario where users are connected to the network by connecting to the closest eNBs to transfer data to and from the remote host.

The output generated by the network simulation is used to create the input dataset for the machine learning model whose target values are computed using the parametric QoE model expressed in (1). The machine learning model is trained using the input dataset. The trained machine learning model is then used for QoE prediction.

The accuracy scores of the training and testing sets of the machine learning model using the MSE as the regression criterion given in (7) for incremental values of maximum allowable depth are shown in Fig. 4. The accuracy scores of the training and testing sets of the machine learning model using the MAE as the regression criterion given in (8) for incremental values of maximum allowable depth are shown in Fig. 5. It is observed that the accuracy improved with the increase in maximum allowable depth. The *decision trees* for the regression criteria MSE and MAE are graphically represented in Fig. 6 and Fig.7 respectively.

All the users with poor QoE are found and the set of eNBs that served these users are isolated. The mode operation is then performed on each of these eNBS to find out which of these are truly dysfunctional. In other words, if the QoE scores of maximum number of users served by a particular eNB have a QoE score less than or equal to one, such an eNB is declared to be dysfunctional but if the maximum number of users served by a particular eNB have a QoE score above 1, then the eNB is declared to be functional.

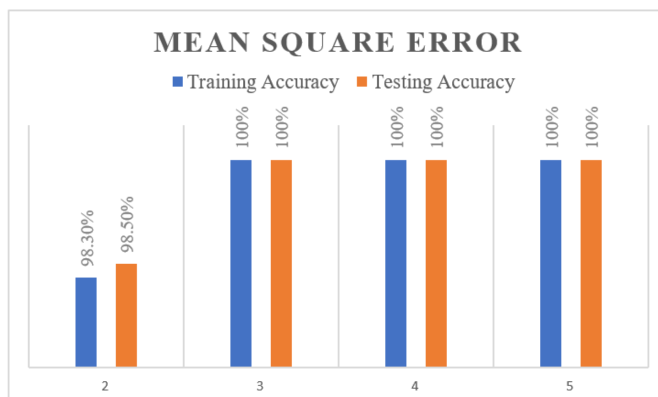

Fig. 4 Training and testing accuracy for the regression criterion MSE across varying values of maximum allowable depths.

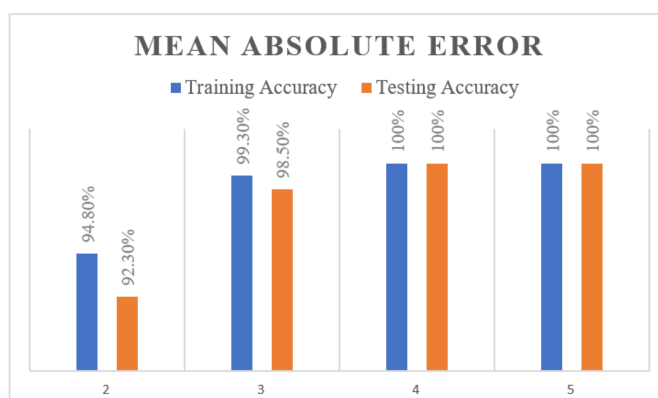

Fig. 5 Training and testing accuracy for the regression criterion MAE across varying values of maximum allowable depths.

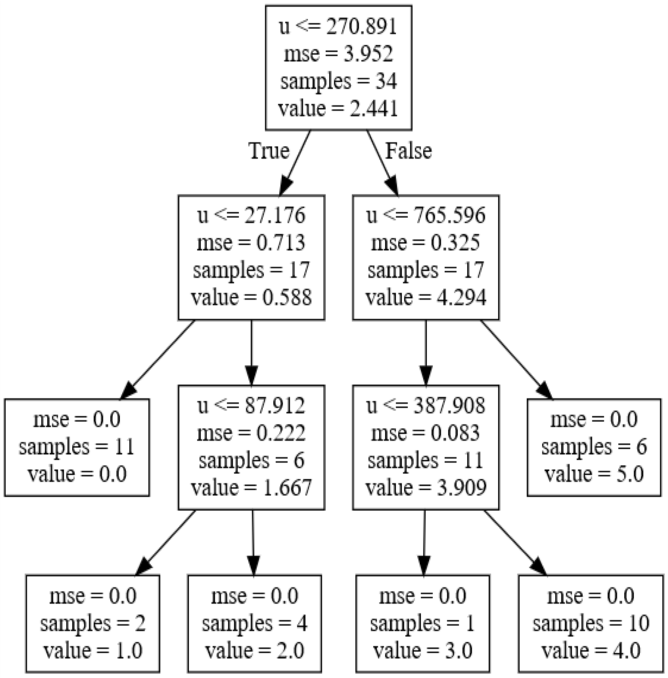

Fig. 6 Tree representation for regression criterion MSE.

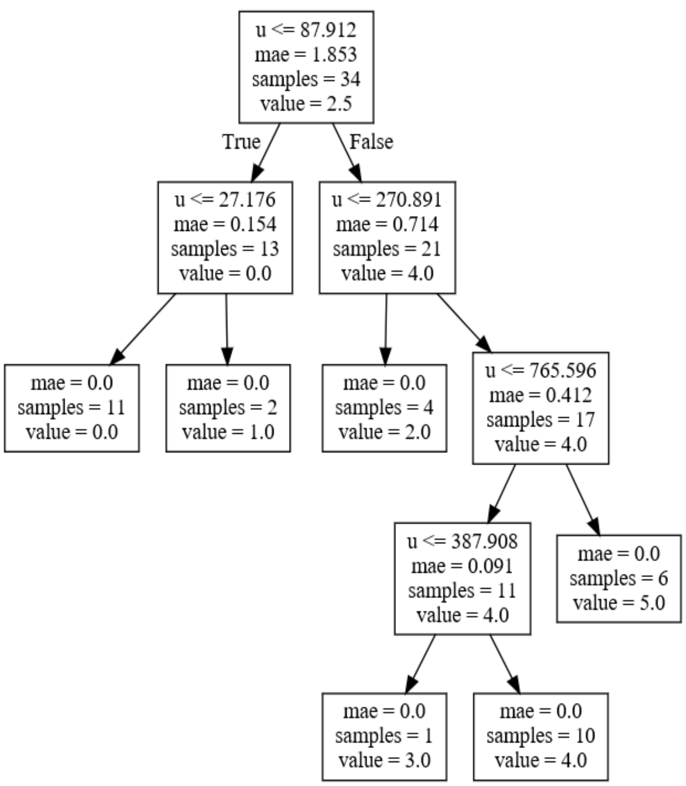

Fig. 7 Tree representation for regression criterion MAE.

V. CONCLUSIONS

This paper introduces a user-centric approach to implement the use case of anomaly detection for a self-healing function in SON networks. The proposed system model uses a network simulator, a parametric QoE model and a machine learning algorithm to demonstrate and evaluate this approach. The *decision tree* algorithm used for machine learning is an optimized version of *CART*. Perfect accuracy of training and testing sets for QoE prediction is achieved by identifying an optimum value for maximum allowable depth for regression criteria MSE and/or MAE. The QoE prediction results are further used to identify dysfunctional serving nodes. The proposed methodology and its demonstration realize a novel method for QoE-driven Anomaly Detection in Self Organizing Networks using Machine Learning that will facilitate self - healing networks. The proposed method can play an important role in supporting future networks with features like enhanced mobile broadband, extreme densification, and energy efficiency where a large number of serving nodes may be partially turned off for certain intervals to attain energy efficiency. Using the proposed system model, a serving node will be identified to be dysfunctional only if the predicted QoE scores of most or all of the users connected to a node are below a certain threshold and thus, help prevent tagging partially turned off nodes as dysfunctional. In our future work, we plan to implement other machine learning algorithms and compare them to study their effectiveness and the scalability of the proposed system model.

REFERENCES

[1] Paulo Valente Klaine, Muhammad Ali Imran, Oluwakayode Onireti, Richard Demo Souza, "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks," *IEEE Communications Surveys & Tutorials*, volume: 19, issue: 4, 2017.

[2] Raquel Barco, Pedro Lazaro, and Pablo Munoz, "A Unified Framework for Self-Healing in Wireless Networks," *IEEE Communications Magazine*, volume: 50, issue: 12, 2012.

[3] ITU-T Recommendation P.10/G.100 Amendment 2, "Vocabulary for performance and quality of service," July 2008.

[4] Eirini Liotou, Dimitris Tsolkas, Nikos Passas, and Lazaros Merakos, "Quality of Experience Management in Mobile Cellular Networks: Key Issues and Design Challenges," *IEEE Communications Magazine*, volume: 53, issue: 7, 2015.

[5] Eirini Liotou, Dimitris Tsolkas, Nikos Passas, Lazaros Merakos, "A Roadmap on QoE Metrics and Models," *23rd International Conference on Telecommunications (ICT)*, 2016.

[6] Dimitris Tsolkas, Eirini Liotou, Nikos Passas, and Lazaros Merakos, "A Survey on Parametric QoE Estimation for Popular Services," *Journal of network and computer applications*, volume: 77, pages: 1-17, January 2017.

[7] *NS-3* [online]. Available: https://www.nsnam.org/

[8] Srisakul Thakolsri, Shoaib Khan, Eckehard Steinbach, Wolfgang Kellerer, "QoE-Driven Cross-Layer Optimization for High Speed Downlink Packet Access," *Journal of Communications*, volume: 4, no. 9, pp. 669–680, Oct. 2009

[9] *Sci-kit learn* [online]. Available: http://scikit-learn.org/stable/#

[10] Wei-Yin Loh. *Classification and regression trees*. [Online]. Available: https://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf