Machine learning approach for quality of experience aware networks

Citation for published version (APA):

Menkovski, V., Exarchakos, G., & Liotta, A. (2010). Machine learning approach for quality of experience aware networks. In *Proceedings 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS'10, Thessaloniki, Greece, November 24-26, 2010)* (pp. 461-466). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/INCOS.2010.86>

DOI:

[10.1109/INCOS.2010.86](https://doi.org/10.1109/INCOS.2010.86)

Document status and date:

Published: 01/01/2010

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Machine Learning approach for Quality of Experience aware networks

Vlado Menkovski, Georgios Exarchakos,
Antonio Liotta

Electrical Engineering Department
Eindhoven University of Technology
Eindhoven, The Netherlands
{v.menkovski, g.exarchakos, a.liotta}@tue.nl

Abstract— Efficient management of multimedia services necessitates the understanding of how the quality of these services is perceived by the users. Estimation of the perceived quality or Quality of Experience (QoE) of the service is a challenging process due to the subjective nature of QoE. This process usually incorporates complex subjective studies that need to recreate the viewing conditions of the service in a controlled environment. In this paper we present Machine Learning techniques for modeling the dependencies of different network and application layer quality of service parameters to the QoE of network services using subjective quality feedback. These accurate QoE prediction models allow us to further develop a geometrical method for calculating the possible remedies per network stream for reaching the desired level of QoE. Finally we present a set of possible network techniques that can deliver the desired improvement to the multimedia streams.

Keywords—*Quality of Experience, QoE, Machine Learning, Subjective Testing*

I. INTRODUCTION

Unlike the typical data centric network services where we can estimate the quality of the service by looking at a constrained set of parameters such as bit rate, latency, jitter and error rate multimedia network services have a vast number of parameters that affect its perceived quality. The reason for this is evident; if we are streaming a video presentation or a conference call is being made over the network the purpose for this data transfer is to deliver the audio and video information in the moment of the use of the service. Due to the way we perceive the video and the audio per bit accuracy is not conveying accurately the level of quality as it would in a file transfer service.

Therefore, different lossy techniques are successful in compressing and transferring the video and audio data within the time constraints that are necessary. However, due to this complexities associated with the perception of audio and video we cannot as easily estimate the quality of the service, by only looking at error rates, signal to noise ratios or latency.

The available bit rate is used by a compressed signal that delivers particular video resolution, audio fidelity, video frame rate and audio quantization frequency. As the bit rate goes down these parameters also go down, but more often

we see different impairments in the content such as blockiness and errors, which is due to the compression or the transport errors. All this, of course, is not perceived independently of the content that is being watched as well as the terminal on which is being watched. Lower complexity videos, with less movement, as news broadcast show fewer impairments than higher complexity videos on the same resource restrictions. In addition the viewing environment such as the size and type of the screen significantly affects the perception of the impairments as well.

So, delivering standard quality rather than standard bit rate is far more important in multimedia services mainly because these two are not so tightly coupled as in data centric services.

The necessary condition for efficiently delivering high quality of network multimedia service is to understand how quality is perceived, what are the parameters that affect it and by how much. This will allow for implementation of network management techniques that can keep the parameters in the desired range and deliver the desired quality to the viewer.

Quality of Experience (QoE) is a way to quantify the experience of a user using the service [1][2]. Although there are a variety of methods in accuracy and complexity to estimate the QoE of a service [3], the most accurate one is the subjective studies. On the one hand objective studies rely on the signal distortion at the encoding or transport stage. On the other hand, subjective studies incorporate a complex and expensive process of rating the multimedia quality by test subjects in a tightly controlled environment.

To circumvent this process we present a Machine Learning technique that builds models in an online learning fashion from customer feedback. These models map the network and application level parameters as well as the environmental conditions with the subjective responses from the customers.

We have decided to use Hoeffding Trees as a basis for our online learning algorithm that delivers a model that can be built quickly and be adaptable enough to incorporate changes in the environment. From the decision trees generated by the online learning algorithms we develop a geometrical approach to find the possible remedies per stream or data point to improve the QoE. The remedies

include the parameter values that need to be changed in order for the QoE to be improved.

Finally we present a discussion on network management and overlay techniques for implementing these remedies. This work covers the key components of the network management loop of multimedia network services. We present here the basis for future work in implementing the details or the implementation of a fully closed multimedia network management loop.

II. OBJECTIVE QOE ESTIMATION

Researchers have devised a number of objective models for estimating QoE of a service. In an effort to standardize these models, the International Telecommunication Union (ITU) has developed a classification [4] based on the focus of each model: *parametric*, *bit-stream*, *media layer* and *hybrid* models.

- Parametric models work on the network level monitoring the network performance and transport error statistics to detect signal distortions that affect perceived QoE.
- Bit-stream models extract and analyse content characteristics from the coded bit-stream to derive the quality.
- Media layer models analyse the media signal via the Human Visual System (HVS) to predict the QoE.

All these models and any combination of them (hybrid) can be computationally expensive and are only partial approaches to the QoE prediction issue. Each model focuses on specific aspects of QoE excluding a number of other factors that may affect it. The lack of a benchmark or standard procedure for comparing models makes the QoE estimation research even more difficult [3]. However, the basic principles behind objective approaches are enough to give an indication of their weakness. Even though the type of content and the way the content is perceived by HVS are important factors for estimating QoE, objective approaches ignore them.

Some methods try to detect signal distortions by comparing original and received signals pixel by pixel (e.g. Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) methods). However, these methods do not take into account the HVS [5]. Hence, even a single-pixel shift in the image in any direction would significantly deteriorate the PSNR value but viewers most likely would not notice any loss of quality.

Other models try to identify the impact of signal transport on the delivered quality by monitoring the Quality of Service (QoS) parameters. Authors of [6] find this method inefficient with weaker results. Thus, they describe a three-layer approach to QoE prediction involving:

- network QoS (NQoS) parameters in the first layer
- Application QoE (AQoS) parameters like resolution, frame rate, color, codec type etc... in the second layer, and
- spatial and temporal perception and acoustic bandpass [6] parameters in the third (perception)

layer. In this layer QoE is a function of NQoS and AQoS (1).

$$QoE = f(AQoS, NQoS) \quad (1)$$

Authors of [6] also claim that a combination of all QoS parameters would yield a more efficient QoE prediction model than examining each one independently.

III. SUBJECTIVE QOE ESTIMATION

Subjective studies match better to the concept of subjective QoE estimation; they capture more accurately the experience and satisfaction levels of viewers. This makes them an appropriate benchmarking to compare QoE prediction methods against. However, these studies need a tightly controlled environment and a carefully selected group of people that statistically represent the viewers population of the service. A more detailed set of guidelines for executing subjective studies is provided by the ITU [7].

Subjective testing is a procedure that requires effort, time and resources for their design and execution. In an effort to reduce the need for subjective studies, [8] and [9] propose a statistical method for predicting QoE of unseen cases based on some initial limited subjective tests. The authors of that method conducted subjective studies using the method of limits [10]. The aim of that method is to detect at which point the perceived quality of a video jumps from acceptable to unacceptable and vice versa. Thus, test subjects (viewers) watch the same video multiple times in ascending or descending quality and mark the point on which the change of quality became noticeable. Via a discriminate analysis [11] of the results of these studies, the authors of [8] and [9] developed models for predicting the quality on unseen cases. Though this approach reduces the need for cumbersome subjective tests, it suffers from limited accuracy of the produced prediction models. In an effort to address these weaknesses, [12] proposes ML-based prediction models (Decision Trees (DT) [13] and Support Vector Machines (SVM) [14]) which outperform the discriminate analysis approach of [8] and [9].

C4.5 [15] is a Decision Tree induction algorithm; its input is training data and outputs a DT prediction model which, based on [12], performs better than the others tested. Decision Tree models (Figure 1) consist of splitting rules in nodes and class values in leaves. The splitting rules examine the value of a single attribute and transfer the decision responsibility to the appropriate branch. Assuming two QoE classes, *acceptable* and *not acceptable*, the leaves of DT in Fig. 1 will be "Yes" or "No". The performance of QoE prediction models built with C4.5 approach was evaluated with cross-validation technique in [12]. The results conclude to a prediction performance of over 90% datapoints correctly estimated. Based on the experiments of [12], using the cross-validation technique [16] to evaluate the prediction performance of the models built with C4.5 approach reach accuracy over 90% datapoints correctly estimated.

Figure 1. Decision Tree prediction model

IV. PREDICTING QOE AT RUN-TIME

The ML-based QoE prediction techniques described so far require initial training data from subjective studies but do not deploy a mechanism for continuous update of the built prediction models. Online QoE prediction approach overcomes those restrictions via continuously collecting real-time user feedback and updating the models using Online Learning ML techniques. An application domain of this method is live streaming as subjective studies in advance are not possible.

Data collection at run time consists of network probing for QoS values and user feedback. The user feedback together with the corresponding QoS values for a specific snapshot of the service comprises a datapoint. If both are available then the system feeds the Online Learning algorithm (learner) that appropriately updates the prediction model. If user feedback is not available and in parallel to this model adaptation mechanism, the model is used for predicting QoE based on the available QoS values. In other words we have a model that estimates the QoE based on available QoS data, and improves over time as more and more feedback is available. Therefore, changes to the environment such as terminal types, content types will be detected via user feedback. This feedback will gradually help the model adapt to the new situation.

Using subjective studies for initial training data lacks of automation and adaptability as their design and execution require human intervention. In large-scale networks where all online nodes can be potential content providers with a large number of direct or indirect content consumers, subjective studies would limit the scalability of the network. For instance, the high heterogeneity of P2P Streaming networks would be a serious obstacle to apply subjective studies.

With the view to large-scale systems, user feedback can produce an even larger number of datapoints. As of [17], a

DT model that could handle this volume of data is Hoeffding Tree [18]. Upon receiving a datapoint, this model processes it and appropriately adapts itself. Thus, all datapoints are sequentially processed making Hoeffding Trees suitable for Online QoE prediction. However, the tree can never have a complete view of the system as datapoints arrive one after the other. Hence, every splitting rule cannot decide with full confidence but only with some probability. The Hoeffding Tree deals with the issue of the number of examples needed to make a split decision by relying on a statistical result known as Hoeffding bound. We make n observations of a random variable r with a range R and determine the computed mean of r to be $\bar{r}$. The Hoeffding bound states with probability $1 - \delta$ that the true mean of the variable is $\bar{r} - \epsilon$ whereby

$$\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n}} \dots \dots \dots (2)$$

If we define the attribute selection criterion as $G(X)$, then $\Delta G = G(x_1) - G(x_2) > 0$ assuming that the x_1 attribute is more favorable (with larger information gain) than x_2 . Now, given the desired δ , the Hoeffding guarantees that x_1 is a better selection with probability δ if n examples are seen where $\Delta G > \epsilon^2$. Experiments done in [17] show that the online learning approach using Hoeffding Trees delivers accurate models from small amount of user feedback data and adapts to changes in the environment quickly and efficiently.

V. BUILDING QOE REMEDY PLANS

Once QoE has been successfully estimated, management actions might be necessary to reestablish good quality of the content at the user end. QoE-aware network management aims at maintaining the perceived QoE at satisfactory levels and minimization of consumed resources. To maintain a target QoE, the service manager has to identify the reasons that caused the QoE deterioration and the parameters which, if properly modified, will bring the QoE back to satisfactory levels. In this section, we introduce a geometric technique that based on the QoE prediction model estimates the minimum needed changes in the measured stream parameters to improve the QoE.

One of the strengths of DT compared to other ML prediction models is their intelligibility. A DT in a way represents a set of rules stacked in a hierarchical way. Simple decision trees commonly define just a few rules that are deduced from the data and used for classification, but when the number of rules grows the size of the DT also grows, and with that, it loses its intelligibility. It is also possible to represent a DT model in the geometric space, defined by the dataset parameters. Consider each of the dataset parameters as a dimension in a hyperspace. Each of the datapoints form the dataset can be represented as a point in this hyperspace. The DT is represented by hyper regions formed by the leaves of the DT (Figure 2). Each node in the DT represents a split or a hyperplane that splits the hyperspace, until we reach a leaf, which carves out a hyper region.

Figure 2. Simple decision tree in 2D space

These hyper regions (as well as the leaves in the DT) are associated with a class label membership. So every datapoint or point in the hyperspace belongs to one of the regions and as such is classified with the corresponding class label. In our particular case the hyper regions are associated class labels i.e. QoE estimates.

Fig. 3 presents an algorithm we devised to automate the procedure of mapping a DT in the hyperspace. The hyperspace can help the design of remedy plans for maintaining the perceived QoE of a service. The DT is a collection of rule paths starting from the root. Each of these paths set the edges (borders) of a hyperregion and their resulting leaf is the class label of that region. That is, each hyper region contains a set of split rules that define the hyper-surface, which carves out the hyper region. The split rules are either inequalities i.e. $Parameter1 \geq Value1$ or equalities i.e. $Parameter1 = Value1$ depending on whether $Parameter1$ is continual or categorical. If the leaf is on the left side of a continual $Parameter1$ split then the split

inequality will be ‘more than or equal to’, if it is on the right side the split inequality will be ‘less than’.

It is easy to determine the class a datapoint belongs to just by checking the values of the parameters against the splitting rules framing the *HyperRegion*-s. All datapoints located within the same hyper region are classified to the same class as the leaf of the corresponding path of the DT. Once a datapoint has been classified and a QoE level estimated, actions need to be taken to recover the satisfactory QoE level. If the datapoint is classified in a hyper-region labeled “Yes” (a.k.a. *acceptable* QoE) no action is necessary. Otherwise (datapoint classified at a “No” region), the remedy plan algorithm has to calculate the distances from that datapoint to the subset of regions $\bar{\Phi}$ labeled as “Yes”. The distance to each of the desired regions is the difference in parameter values that are needed in order to move the datapoint to the desired regions.

The output of that remedy plan algorithm is a set of distance vectors, which define the size of the change (negative, zero, positive) that has to be applied on every parameter. For clarity purposed we provide an example from the laptop dataset of [12]. Fig. 1 is a snapshot of the prediction model built from this dataset. Assuming the datapoint of Table I, the QoE prediction based on the prediction model will be “Not Acceptable” as the splitting rules drive the decision to a leaf “No”. Since the V. Framerate is less than 12.5 and the V.Bitrate is less than 32 the datapoint reaches a leaf with “Not Acceptable” class associated with it.

TABLE I. EXAMPLE DATAPPOINT

Video SI	Video TI	V. Bitrate	V. Framerate
67	70	32	10

The algorithm above produces a set of distance vectors which, however, are not all applicable. There are parameters that characterize the type of the content such as the Video SI (Video Spatial Information [7]) and the Video TI (Video Temporal Information [7]) and cannot be changed. In our example, the aim is to increase the V.Bitrate and

Start from the root node and call a recursive method *FindLeaves*

FindLeaves:

- 1) If the node has children
 - a) Call *FindLeaves* on each child
 - b) Add the SplitRule on each of the Hyper Regions ($\bar{\Phi}$) that are returned
 - i) If the leaf split is categorical add a Split Rule: Attribute = ‘value’
 - ii) If the leaf split is continual add on the leaves from the left side SplitRule: Attribute < value, and on the leaves from the right side Attribute > value
 - c) Return the set of Hyper Regions ($\bar{\Phi}$)
- 2) Else, node has no children (leaf)
 - a) Create an Hyper Region object
 - i) Assign the class of the leaf to the $\bar{\Phi}$
 - ii) Return $\bar{\Phi}$

Figure 3. DT to Hyper Region algorithm

V.Framerate. Aiming at a V.Bitrates for this particular datapoint increased by one step to 64kb/s has no effect as the datapoint would result again to a “No” leaf in the DT. However, a small increase to the V.Framerate up to 15f/s is able to push QoE level back to acceptable levels without adding more bandwidth.

We can deduce a rule from the model that a video with these characteristics needs to have higher V.Framerate for it to be perceived with high quality. However, this rule is not easily evident from only looking at the model. We can also imagine a system with large number of attributes that we can change where tuning this attributes the right way becomes an increasing problem. Further down this line of reasoning, if we want to make a system-wise improvement that will increase the QoE of most streams we cannot easily derive which parameters are best to be increased and by how much.

In the case of the example datapoint the algorithm returns the two possible paths:

- Increasing the Framerate to above 12.5f/s
- Increasing the V. Bitrate to above 32kb/s and the Video TI to above 87

Modifying Video TI is not an option because it is defining the type of content. Thus, the only option is to increase the frame rate. In a general case, there can be many different paths to a hyper region with the desired class.

To automate the process we can assign cost functions to the change of the attribute values and automatically calculate the cheapest way to reach the desired QoE. In this manner attributes that are not changeable, such as the Video TI, can have infinite value of the cost function.

Given a datapoint and a target label the algorithm produces a set of change vectors. Each of the change vectors applied to the datapoint moves the datapoint to a hyper-region classified with the target label. In other words, each change vector is one possible fix for the datapoint.

$$\bar{\Phi} = FindLeaves(DT, QoE) \quad (3)$$

$$\Delta\phi_i = Distance(\Phi_i, \bar{d}) \quad (4)$$

$$\Delta\phi_{optimum} = \min_i (Cost(\Delta\phi_i)) \quad (5)$$

In (3), $\bar{\Phi}$ is a set of regions with a targeted QoE value. The distance function in (4) calculates the vector of distances for each attribute to the target region in $\bar{\Delta\phi}$. The optimal distance vector is the one with minimal cost (5) for the given input datapoint $\bar{d}$. The *Cost* function in (5) is dependent on the application. Each system has explicit and implicit costs associated with changes of specific parameters.

VI. DISCUSSION ON QOE REMEDIES APPLICABILITY

Improving the perceived QoE of a stream can be a complex task depending on the remedies to which the prediction models concluded. On the contrary to e.g. packet loss, latency or jitter, modifications to video frame/bit rate can be straightforward in a client-server environment.

However, in volatile large-scale resource distribution systems like Peer-to-Peer Networks streaming multimedia from one point to another may involve a number of intermediate nodes (Fig. 4). In these cases, optimizations of frame or bit rates get complicated.

Figure 4. Client-Server (left) vs. P2P (right) Streaming scenario

In general, a P2P video streaming scenario involves a set of nodes that receive and relay the stream forming a chain of consumers/providers. While in client-server model clients have a unique source of a stream, in P2P environments the same stream may reach a node via multiple paths. Based on the QoE feedback from clients, a server may choose to improve the video framerate or bitrate of a stream. The applicability of these improvements are restricted by bottlenecks in the connection between server and client. Assuming two directly connected nodes B and C of a delivery path in a P2P network, the situation is similar as B relays the stream to C; similar restrictions and bottlenecks apply in B-C connection. However, there are further restrictions that require different remedy implementations.

Assuming that the relay node B receives a stream from A at certain video frame and bit rate, B cannot deliver to C the same video at higher rates. If perceived QoE at C is unsatisfactory, the QoE remedy models may suggest increase of frame and/or bit rate of the received stream at C. However, B cannot deliver higher rates than the ones it receives. If previous relay nodes cannot contribute to these remedies, C may request the delivery of the same stream from multiple nodes in parallel to reach the expected received frame or bit rates.

This parallelism can be expressed in stream, chunk or even frame level. In very simple situations, two or more streams of the same video may simultaneously reach a client (Fig. 5). Stream redundancy helps a node to recover bigger portions of lost information but wastes bandwidth. To avoid useless redundancy, the client may request different chunks of the same stream from different nodes. There are a number of different strategies for splitting a stream into chunks inducing also different costs. Another option to materialize a remedy in this scenario is the delivery of non-overlapping frame streams of the same video that collectively construct the whole video at the destination.

Each of these strategies comes with a cost function which also depends on the application domain they are applied to. For instance, in video conference applications latency is in higher priority than bandwidth consumption, hence, strategies with low time overhead are preferred.

Figure 5. (a) Single delivery path, (b) stream redundancy, (c) frame delivery parallelism and (d) chunk delivery parallelism

On the other hand, file storage distributed networks try to optimize the bandwidth consumption. All the categories of remedy implementation strategies mentioned above use parallelism to the delivered information. They require the discovery of alternative sources of a part of the stream and coordination among all simultaneous sources. The necessary protocols that will locate and select the nodes that can deliver the required QoE are at the moment under development and testing.

To establish parallelism, a node willing to improve its QoE has to address the problem of discovery of potential stream sources. Candidate stream providers are node preceding the video requestor in the delivery path. Some of these nodes may reject the request for streaming if that would deteriorate the QoE they deliver to other nodes. Thus, the QoE prediction models need to be deployed on every node and monitor both incoming and outgoing streams QoE. The selection of the most appropriate remedy depends on its cost function but its implementation may prove quite costly.

Apart from QoS optimization from node to node, QoE-aware content delivery, e.g. in P2P streaming networks, may optimize the utilization of resources. The ML-based QoE prediction models presented in this paper do not only handle the perceived QoE recovery but resource minimization, as well. They can propose reverse remedies that can bring lower resource consumption without any deterioration of the perceived QoE. Any remedy has to bring the whole system in a relatively stable situation avoiding parameter values that with only slight changes can push the QoE to unacceptable levels.

VII. CONCLUSIONS AND FUTURE WORK

This paper tries to address the problem of QoE-aware service provision. This problem is addressed in three phases: QoE prediction, estimation of remedy plans, remedy materialization. The first phase aims at minimizing the required subjective studies keeping the accuracy as high as possible. The second phase outputs a set of plans that if implemented would bring the delivered QoE back to acceptable levels. Though these two phases have a more

concrete form the last one requires further investigation and testing.

Our current research focuses on ways to identify the most suitable and inexpensive remedy plan. This is a quite challenging task because of the complexity of cost functions the heterogeneity of domains our methodology can be applied to. In parallel, there is ongoing research on P2P Networks to apply similar techniques. We investigate the ideas that QoE-based rerouting and increase of delivery parallelism can maintain QoE and optimize resource utilization. These ideas are applicable to situations with limited flexibility of QoS parameters i.e. P2P Networks or any application over large-scale non-centrally controlled or single-owned packet networks.

REFERENCES

- [1] R. Jain, "Quality of experience," *IEEE Multimedia*, vol. 11, 2004, pp. 96–95.
- [2] S. Winkler and P. Mohandas, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics," *Broadcasting, IEEE Transactions on*, vol. 54, 2008, pp. 660–668.
- [3] S. Winkler, "Video Quality Measurement Standards - Current Status and Trends," *Proceedings of ICICS 2009*, Macau, PRC: 2009.
- [4] A. Takahashi, D. Hands, and V. Barriac, "Standardization activities in the ITU for a QoE assessment of IPTV," *Communications Magazine, IEEE*, vol. 46, 2008, pp. 78–84.
- [5] S. Winkler, *Video Quality and Beyond*, Symmetricom, 2007.
- [6] M. Siller and J. Woods, "QoS arbitration for improving the QoE in multimedia transmission," *Visual Information Engineering, 2003. VIE 2003. International Conference on*, 2003, pp. 238–241.
- [7] R.I. ITU-T, "910," *Subjective video quality assessment methods for multimedia applications*, 1999.
- [8] F. Agboma and A. Liotta, "Addressing user expectations in mobile content delivery," *Mobile Information Systems*, vol. 3, Jan. 2007, pp. 153–164.
- [9] F. Agboma and A. Liotta, "QoE-aware QoS management," *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, Linz, Austria: ACM, 2008, pp. 111–116.
- [10] G.T. Fechner, E.G. Boring, H.E. Adler, and D.H. Howes, *Elements of psychophysics / Translated by Helmut E. Adler ; Edited by David H. Howes [and] Edwin G. Boring ; with an introd. by Edwin G. Boring*, New York :: Holt, Rinehart and Winston, 1966.
- [11] W.R. Klecka, *Discriminant analysis*, SAGE, 1980.
- [12] V. Menkovski, A. Oredope, A. Liotta, and A. Cuadra Sánchez, "Predicting Quality of Experience in Multimedia Streaming," *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, Kuala Lumpur, Malaysia: 2009, pp. 52–59.
- [13] J. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, Mar. 1986, pp. 81–106.
- [14] A.J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, 2004, pp. 199–222.
- [15] J.R. Quinlan, *C4.5*, Morgan Kaufmann, 2003.
- [16] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *International Joint Conference on Artificial Intelligence*, LAWRENCE ERLBAUM ASSOCIATES LTD, 1995, pp. 1137–1145.
- [17] P. Domingos and G. Hulten, "Mining high-speed data streams," *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 71–80.
- [18] R. Kirkby, "Improving Hoefding Trees," Department of Computer Science, University of Waikato, 2007.