

Article

A Multivariate Time-Series Based Approach for Quality Modeling in Wireless Networks

Leonardo Aguayo ¹, Sergio Fortes ^{2,*}, Carlos Baena ², Eduardo Baena ² and Raquel Barco ²

¹ Departamento de Engenharia Elétrica, Campus Universitário Darcy Ribeiro, Universidade de Brasília, Brasília, DF 70910-900, Brazil; aguayo@unb.br

² Departamento de Ingeniería de Comunicaciones, Andalucía Tech, Campus de Teatinos s/n, Universidad de Málaga, 29071 Málaga, Spain; jcbg@ic.uma.es (C.B.); ebm@ic.uma.es (E.B.); rbm@ic.uma.es (R.B.)

* Correspondence: sfr@ic.uma.es

Abstract: This work presents a method for estimating key quality indicators (KQIs) from measurements gathered at the nodes of a wireless network. The procedure employs multivariate adaptive filtering and a clustering algorithm to produce a KQI time-series suitable for post-processing by the network management system. The framework design, aimed to be applied to 5G and 6G systems, can cope with a nonstationary environment, allow fast and online training, and provide flexibility for its implementation. The concept's feasibility was evaluated using measurements collected from a live heterogeneous network, and initial results were compared to other linear regression techniques. Suggestions for modifications in the algorithms are also described, as well as directions for future research.

Keywords: mobile networks; modeling; KQI; QoE; machine-learning



Citation: Aguayo, L.; Fortes, S.; Baena, C.; Baena, E.; Barco, R. A Multivariate Time-Series Based Approach for Quality Modeling in Wireless Networks. *Sensors* **2021**, *21*, 2017. <https://doi.org/10.3390/s21062017>

Academic Editor: Joanna Kolodziej

Received: 28 January 2021

Accepted: 2 March 2021

Published: 12 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Real-time service quality assessment is one of the critical aspects of modern wireless networks. However, its practical implementation faces some challenges. One of them is scalability: the sustained increase in the number of nodes demands fast, stable, and distributed data processing. IoT and M2M scenarios encountered in 5G and beyond systems put even more pressure on the monitoring systems to cope with the constant increment of traffic volume.

A second aspect is a need, from the service perspective, to measure quality indicators at different protocol or abstraction levels. Examples are interference level at the physical layer, delays at the MAC and network layers, and net data rate at session or application layers. The latest application layer indicators, also called key quality indicators (KQIs) [1], are especially relevant for the management of cellular networks. KQIs directly measure the service performance that the user experiences when executing particular services (e.g., web browsing, file download, video streaming [2], and video games [3]). A service provider can then use selected KQIs to improve its network operation and plan further expansions. Furthermore, when the final user of the system is a person, KQIs can be used to estimate subjective indicators, such as the mean opinion score (MOS) for voice services and PEVQ (perceptual evaluation of video quality) [4].

A third difficulty is implementing mechanisms that provide real-time assessment of the network status, potentially impacting the service provider's OPEX and CAPEX. The scope of self-organized networks (SONs) [5] aims to address these demands.

Although indispensable for these three aspects, acquiring the KQIs implies facing considerable obstacles. Firstly, KQIs are out of the operator's measurable scope, as they are part of the application layer. Although monitoring applications or deep packet inspection might be a solution to access such measures, the growing use of end-to-end high layer

encryption and concerns about users' privacy can make those approaches unsuitable for the task.

In this context, modeling KQIs from lower layer (accessible by the operator) measurements can be vital to estimate the application specific user performance. We look for computationally efficient algorithms capable of providing online results and suitable to be implemented in different network nodes, such as UEs, eNBs, and EPCs [6]. Simultaneously, the outputs of these algorithms are also time-series. The availability of time-indexed data is useful for additional post-processing by the operations and management (OAM) subsystem, where flexibility is a desirable feature to enable new choices of digital signal processing algorithms and the application of different machine learning (ML) techniques.

ML techniques are expected to play a relevant role in the management of wireless networks, as described by Ali et al. [7] and by Boutaba et al. [8]. Recent examples in the literature are given by Wang et al. [9], where the authors presented a robust architecture for artificial intelligence-enabled Internet of Things (AIoT) systems, and by Fiandrino et al. [10], which described an ML based general framework to optimize the operation of 5G and beyond networks. Machine learning mechanisms, such as recursive neural networks (RNNs), are also being applied to address other challenges in the context of cellular networks, as detailed by Wang et al. [11] on the security issue of voice cloning.

Narrowing the range of applications, one of the purposes of ML in wireless networks is the automation of service provisioning with a focus on the quality of service (QoS) and quality of experience (QoE), where KQIs provide relevant information for numerical processing by the ML algorithms. In this field, the work by Herrera-Garcia et al. [1] addressed the problem of KQI modeling using an ML approach based on regression techniques, but it did not consider an online, dynamic construction of the models. Baena et al. [12] extended the regression approach to consider the time dependency of KQI modeling in a video services scenario. However, the method requires prior knowledge of some parameters related to network configuration. Fan et al. [13] also proposed an ML based method to map KPIs to KQIs using sliding-window partitioning and random forest algorithms, but again without an explicit model for dynamic regression. Additional studies by Fortes et al. [14–17] have focused on the use of other high layer contextual information (e.g., location, social data) to support the management process in correlated yet different contexts (mainly SON), without explicit reference to multivariate time-series processing algorithms.

Going beyond the described state-of-the-art, the present paper defines a novel framework for estimating KQIs based on multivariate adaptive filters in conjunction with an ML clustering algorithm. The main distinction between this work and the ones mentioned above is twofold: First, it contains an explicit description of a time based approach suitable for online KQI modeling. This method has the flexibility to be used in conjunction with other ML techniques, and we provide suggestions of different algorithms that can be used within the general framework. The second distinction is the possibility to obtain distinct KQI models from measurements when facing a nonstationary environment or when there are different service configurations.

Adaptive filtering algorithms have suitable characteristics for dynamic KQI modeling, such as online training, flexibility to provide linear and non-linear mappings, the capability to handle multivariate time-series, and fast execution. Furthermore, it naturally provides an indicator of accurate modeling in a stationary environment, namely the estimation error obtained in the training phase. Proper handling of this estimation error in conjunction with a clustering algorithm is the basis of the framework, and to our best knowledge, it has not yet been applied to the problem of KQI modeling in wireless networks.

The rest of this paper is organized as follows: Section 2 provides the mathematical formalization for the problem and the notation used in the rest of the text. Section 3 presents the general description of the framework, while Section 4 details the algorithms used for KQI modeling in a non-stationary context. Section 5 shows (i) an initial assessment of the proposed approach using collected data from a real environment and (ii) open research

challenges using the same general framework, but with different algorithms and slight variations. Finally, Section 6 summarizes the conclusions of this work.

2. Notation, Hypothesis, and Simplifications

Measurements gathered from wireless networks may be very different in nature and are originated from distinct network elements with different purposes. Here, we make one basic distinction between measurements, parameters, and auxiliary data.

Here, we call measurements what we get from the network. Typical representatives are:

- Data from the PHY/MAC protocol layers that can be obtained from the nodes. Typical examples are RSSI (received signal strength indicator) measured in eNB and drive tests logs.
- Classical KPIs from eNBs related to mobility and connection management, such as handover success rate and dropped call rates.

We call parameters data whose values:

- Can be set in equipment, such as: UE transmitter power; number of RF carriers; antenna downtilt;
- Numbers that characterizes a service: the minimum data rate for data services; maximum acceptable delay; file data size; network bandwidth. The popularization of software-defined radio (SDR) equipment can naturally blur the distinction between these two subgroups.

Finally, we call auxiliary data information that can be gathered from other sources than the wireless network, such as service type (video streaming, text messages, VoIP) and georeferenced information (demographic profiles, RF propagation environment).

In the rest of the text, we will use the notation defined in Table 1.

Table 1. Mathematical notation: main symbols with short descriptions.

Symbol	Meaning
$\mathcal{M}, \mathcal{K}, \mathcal{S}, \mathcal{A}$	general sets of (respectively): data measurements, KQIs, network status and actions
$\mathcal{F}_{XY}(\cdot)$	mapping between sets \mathcal{X} and \mathcal{Y}
n, j, i	indexes used, respectively, for: time instant, input data channel, and time-series segment
M_n	multivariate measurements gathered at time n , dimensions $d_m \times 1$
m_n^j	j^{th} (from d_m) measurement gathered at instant n
$m_{L,n}^j$	j^{th} measurement time-series window from time $n - L + 1$ up to n , dimensions $1 \times L$
KQI_n^{target}	selected KQI to be modeled, at time n
$\{\mathbf{w}_M\}$	generic weights of adaptive filters
$\mathbf{w}_{M,n}^j$	j^{th} (mapping) adaptive filter weights at time n , dimensions $L \times 1$
$\mathbf{w}_{p,n}^j$	j^{th} (prediction) adaptive filter weights at time n , dimensions $L \times 1$
$e_n^{p,j}$	j^{th} prediction error at time n
e_n^M	mapping estimation error at time n

Using these definitions and notation, we can state our main working hypothesis:

Hypothesis 1. Time stamps that come from different sources are consistent, i.e., the algorithms perform on the same sequence of time instants $[t_1 \ t_2 \ \dots \ t_n]$ for all measurements.

Hypothesis 2. The set \mathcal{K} can be obtained from \mathcal{M} . In this front, some effort has been made to organize the set of measurement reports in standardization forums [18–20].

3. Reference Model

The proposed approach is based on the global architecture of Figure 1. From the measurements \mathcal{M} obtained from the network, with the aid of information in auxiliary databanks, a mapping $\mathcal{F}_{MK}(\cdot)$ is built to construct the set of KQIs \mathcal{K} . Further processing can be used to obtain the maps $\mathcal{F}_{KS}(\cdot)$ to obtain the network status \mathcal{S} from the KQIs, as well $\mathcal{F}_{SA}(\cdot)$, which maps \mathcal{S} to the set of network parameters and actions \mathcal{A} .

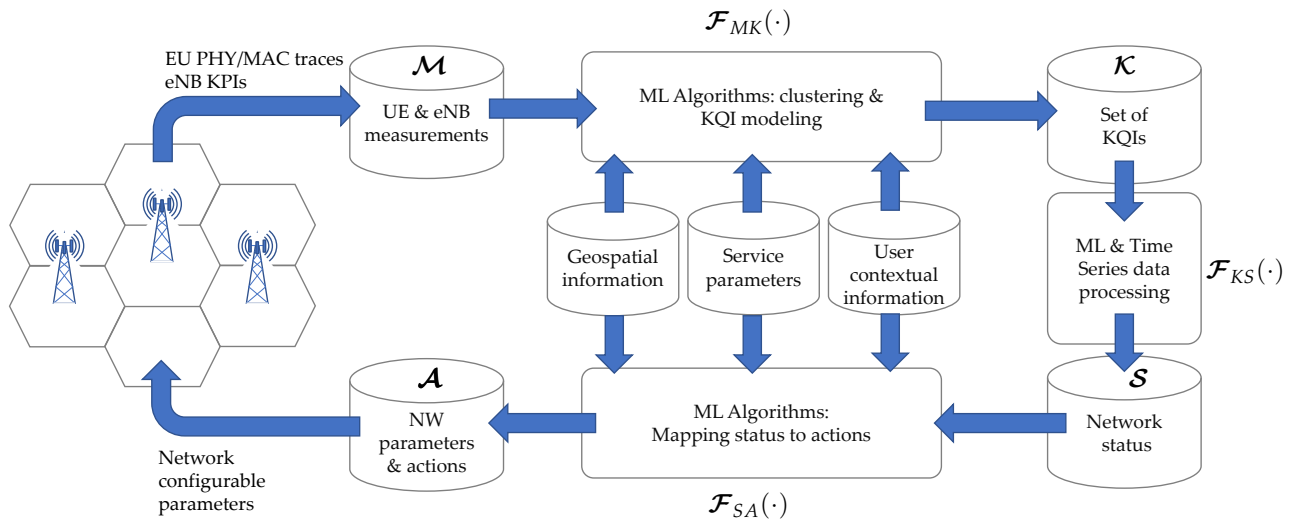


Figure 1. General information flow for self-organized network (SON) status prediction and assessment through key quality indicator (KQI) modeling and processing.

The present work focuses on the problem of building the map $\mathcal{F}_{MK}(\cdot)$, using time-stamped information originated from measurements at the physical layer, as well from service parameters.

Before going to the formal aspects, we briefly discuss qualitatively the main framework designed to obtain $\mathcal{F}_{MK}(\cdot)$. The procedure can be divided into four steps: mapping, segmentation, clustering, and operation. Here, it is assumed that the set of training time-series is a good representation of the measurements (or input) space and that there is also a set of KQI time-series available for training, representative of the output space.

To produce the KQI mapping, we use a multichannel linear adaptive filter [21,22] with d_m inputs, each one from a specific measurement $m_n^1, m_n^2, \dots, m_n^{d_m}$ collected at instant n . A selected $\text{KQI}_n^{\text{target}}$ is used to train the bank of filters, and the coefficients $\mathbf{w}_{M,n}^j$, $j = 1, 2, \dots, d_m$, build the map $\mathcal{F}_{MK}(\cdot)$.

It is worth mentioning that if the inputs are non-stationary, the adaptive filter will track the changes in the input space, and the set $\{\mathbf{w}_M\}$ will convey only the latest representation of $\text{KQI}_n^{\text{target}}$ at the output space. If there is a need to cope with changes of regime, a mechanism for detecting them is needed. In such a case, a possible solution is to store the correspondent set of weights for each stationary section, as discussed further. This step is called segmentation.

In parallel, the same input sequence M_n is used to build a set of autoregressive models $\mathbf{w}_{P,n}^j$, $j = 1, 2, \dots, d_m$. The purpose of this second filter bank is to provide a representation of the input space that can be used in the operation phase. Again, a set of d_m adaptive predictors can be used to extract the parameters that represent the input space.

At the end of this step, both sets of weights, generically referred to as $\{\mathbf{w}_M\}^{\text{train}}$ and $\{\mathbf{w}_P\}^{\text{train}}$, are stored to be used in the next phases. The full architecture is depicted in Figure 2.

The third phase, clustering, is aimed at extracting only the non-redundant models on the sets $\{\mathbf{w}_M\}^{\text{train}}$ and $\{\mathbf{w}_P\}^{\text{train}}$. This step is not strictly mandatory, but it was adopted as

part of the framework with the objective of reducing the number of parameters needed to rebuild KQI_n^{target} from the information available from the input space.

At the last step, the operation phase, new samples of the measurements are fed into a bank of predictors, using the same architectural structure used in the training step. Now, the system is constantly producing $\{\mathbf{w}_P\}^{\text{oper}}$, which can be compared with $\{\mathbf{w}_P\}^{\text{train}}$ to recover the correspondent $\{\mathbf{w}_M\}^{\text{train}}$. A criterion of the minimum distance between $\{\mathbf{w}_M\}^{\text{train}}$ and $\{\mathbf{w}_M\}^{\text{oper}}$ is used to select the best mapping in real time.

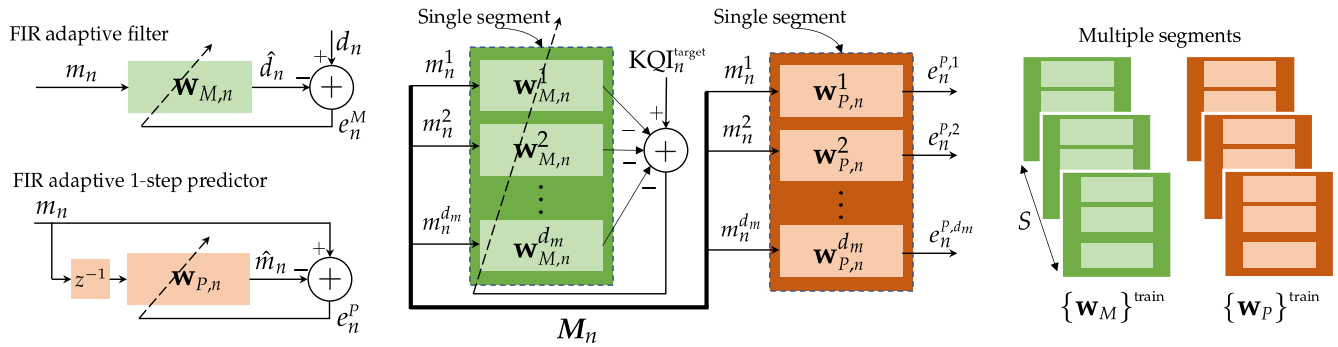


Figure 2. Adaptive filters used to construct representations of the measurement set M_n (inputs) and output space (KQIs). Left: representation of a single adaptive filter with external training data d_n (top), used as adaptive predictor (bottom); middle: multivariate adaptive filters (KQI mapping and multichannel predictors), for a single time-series segment; right: complete set of filter weights after the training phase.

The idea behind the model is simple, and it can be used in conjunction with anomaly detection (ND) schemes [23–26] to identify changes in the input space in the non-stationary scenario. This can be done in both the training and operation phases. In the training phase, if there are significant alterations in the input space, new elements can be added to the $\{\mathbf{w}_M\}^{\text{train}}$ and $\{\mathbf{w}_P\}^{\text{train}}$ sets. Changes in M_n , such as a modification of a service parameter, will result in different weights due to the tracking nature of the adaptive filters.

These new elements can be labeled with an index that represents a quasi-stationary section of a general non-stationary behavior. The strategy is to build not only a single map, but a dictionary of maps (or an atlas) that provides the system a set of representations of the output space \mathcal{K} as functions of the input space \mathcal{M} , for different service or network conditions. In the operation phase, the ND mechanism can be used as a trigger to identify a new network operation regime not yet represented by $\{\mathbf{w}_M\}^{\text{train}}$ and request new maps to be added to the current atlas.

One possible approach to build new maps from the available data, in the training phase, is the utilization of the prediction errors:

$$e_n^{p,j} = m_n^j - \hat{m}_n^j \quad (1)$$

$j = 1, 2, \dots, d_m$, where \hat{m}_n^j is the estimation of the j th measurement at instant n made by the adaptive predictor. Analogously, the estimation error e_n^M can also be used. By monitoring the values of $e_n^{p,j}$ and e_n^M , it is possible to make a decision about whether a new map is needed or not.

4. Proposed Algorithms

This section describes the specific algorithms developed to implement the KQI modeling functionality as defined in the previous section. These include the mechanisms associated with the determination of linear mappings, namely adaptive filters.

Adaptive filters make use of a recursive, sample based rule to update their parameters (or weights \mathbf{w}). If the input environment is a stationary process, after a certain number of iterations, the weights will converge (in some statistical sense) to values regarded as optimum when they minimize a certain cost function $J(\mathbf{w})$. Among different options for

the algorithms, there are two common families of adaptive filters that provide solutions for the recursive problem of weight update, based on the formulation of $J(\mathbf{w})$. The first is LMS based, derived from a statistical approach, where the steepest descent algorithm is widely used. The second is RLS based, where variations of the recursive least-squares algorithm are applied. The area of adaptive filtering is mature [27], and it was chosen due to: (i) simplicity of training, (ii) well-known properties of convergence and stability, and (iii) broad choice of options among existing algorithms.

4.1. Determination of Linear Mappings

The input of the multichannel adaptive filter responsible for providing $\mathcal{F}_{MK}(\cdot)$ is a multivariate time-series with measurements $m_n^j, j = 1, 2, \dots, d_m$. For the sake of simplicity, we chose a finite impulse response (FIR) configuration with the same order L for each channel. A buffer of L past samples of each input channel j is then used to produce the correspondent j^{th} output, as depicted in Figure 2. The output of all d_m filters is now combined to produce $\widehat{\text{KQI}}_n^{\text{target}}$:

$$\widehat{\text{KQI}}_n^{\text{target}} = \sum_{j=1}^{d_m} \mathbf{m}_{L,n}^j \mathbf{w}_{M,n}^j. \quad (2)$$

The difference between the estimative and the selected $\text{KQI}_n^{\text{target}}$, the mapping error, is:

$$e_n^M = \text{KQI}_n^{\text{target}} - \widehat{\text{KQI}}_n^{\text{target}}. \quad (3)$$

To provide processing speed and avoid numerical instability, we used the least-mean squares (LMS) algorithm to update the coefficients of each channel:

$$\mathbf{w}_{M,n+1}^j = \mathbf{w}_{M,n}^j + 2\mu e_n^M \mathbf{m}_{L,n}^j, \quad (4)$$

where μ is the step-size parameter of the LMS algorithm. Section 5.9 contains a brief discussion on different algorithms that may be used to update the coefficients. After convergence, the $\mathcal{F}_{MK}(\cdot)$ mapping is stored as the set $\{\mathbf{w}_M\}^{\text{train}}$ of all coefficients $\mathbf{w}_{M,n}^j$.

For the multichannel adaptive filter used to build the j^{th} autoregressive model $\mathbf{w}_{P,n}^j$ associated with the j^{th} measurement, we calculated the corresponding prediction error $e_n^{P,j}$:

$$e_n^{P,j} = m_n^j - \mathbf{m}_{L,n-1}^j \mathbf{w}_{P,n}^j. \quad (5)$$

We also used the LMS rule to update the regression coefficients for each channel $j, j = 1, 2, \dots, d_m$:

$$\mathbf{w}_{P,n+1}^j = \mathbf{w}_{P,n}^j + 2\mu e_n^{P,j} \mathbf{m}_{L,n-1}^j, \quad (6)$$

Again, all prediction filters have the same order L , and after convergence, the autoregressive coefficients constitutes the set $\{\mathbf{w}_P\}^{\text{train}}$. We also point out that there is no need to use the same order L for both mapping and prediction filters.

For the stationary case, both sets $\{\mathbf{w}_M\}^{\text{train}}$ and $\{\mathbf{w}_P\}^{\text{train}}$ will contain d_m elements, each one of length L . Due to the tracking nature of adaptive filters, in a non-stationary environment, there is a need to identify changes in the input data that lead to a different $\mathcal{F}_{MK}(\cdot)$. Therefore, a new time-series segment should be detected, and the corresponding mapping and predictor weights must be stored. Using i as the index for the segments, the weights for prediction and mapping, as well the prediction and estimation errors are expressed respectively as $\mathbf{w}_{P,n}^{j,i}, \mathbf{w}_{M,n}^{j,i}, e_n^{P,j,i}$, and $e_n^{M,i}$. If S is the number of segments, by the final training phase, there will be an atlas of S linear mappings for KQI estimation, as well as S banks of autoregressive models. A set of $S \cdot L \cdot d_m$ parameters will then represents $\mathcal{F}_{MK}(\cdot)$.

4.2. Detection of Quasi-Stationary Segments

The detection of a new quasi-stationary state s^{i+1} can be performed by different anomaly (or novelty) algorithms, such as PCA [28], convolutional neural networks [29], and the Kullback–Leibler divergence [25]. The proposed approach uses the available set of $d_m \times S$ prediction errors $e_n^{p,j,i}$ and a threshold γ_{MAX} as follows: for all i segments, verify if:

$$\gamma_i = \sum_{j=1}^{d_m} \|e_n^{p,j,i}\|^2 \geq \gamma_{MAX} \quad (7)$$

is true. If $\gamma_i \geq \gamma_{MAX}$ for all segments, none of the previous i mappings is considered suitable, and a new segment is added. We justify the use of the prediction errors because a regime change in a single channel may lead to a different $\mathcal{F}_{MK}(\cdot)$. Now, the corresponding coefficients $\mathbf{w}_{M,n}^{j,i+1}$ and $\mathbf{w}_{P,n}^{j,i+1}$ are updated according to Equations (4) and (6). No further modifications are made on any other filter coefficients, and the process continues while there is available training data.

Other criteria can be used to detect a new segment using the information available from the training, such as a weighted combination of $e_n^{p,j,i}$ (in j index) to favor selected measurements. Utilize the mapping error $e_n^{M,i}$ in conjunction with the prediction errors is also possible, but in our approach, only the prediction errors are available in the operating phase. Another option is to define a dissimilarity metric in the feature space, i.e., use the distance between $\{\mathbf{w}_{M,n}\}^{\text{train}}$ or $\{\mathbf{w}_{P,n}\}^{\text{train}}$ and their past versions at some previous instant $n - T$. Here, a threshold is also needed.

4.3. Clustering Phase

After segmentation, there is a total of $S \geq 1$ segments. Clustering can be performed on $\{\mathbf{w}_P^i\}^{\text{train}}$ or simultaneously in both sets $\{\mathbf{w}_P^i\}^{\text{train}}$ and $\{\mathbf{w}_M^i\}^{\text{train}}$ ($i = 1, 2, \dots, S$). The latter option is preferred, as the clustering procedure should preserve the mappings $\mathcal{F}_{MK}(\cdot)$. For instance, when using the self-organizing map (SOM) algorithm, its inputs would be:

$$\mathbf{x}_{in}^i = \begin{bmatrix} \mathbf{w}_M^i \\ \mathbf{w}_P^i \end{bmatrix}, \quad (8)$$

for $i = 1, 2, \dots, S$. For moderate values of S , e.g., $S \leq 50$, simpler methods of clustering can be performed, such as k -means.

4.4. Operation Phase

In this stage, new measurements are presented to the system. The mapping information is encoded in the sets $\{\mathbf{w}_M^i\}^{\text{train}}$ and $\{\mathbf{w}_P^i\}^{\text{train}}$. A prediction of the measurements is performed in this phase in the same fashion as in the training phase. The coefficients $\{\mathbf{w}_P\}^{\text{oper}}$ are compared with all $\{\mathbf{w}_P^i\}^{\text{train}}$, and the segment with the nearest distance is selected, i.e., choose the segment indexed by s^* such that:

$$\left\| \mathbf{w}_P^{s^*,\text{train}} - \mathbf{w}_P^{\text{oper}} \right\|^2 \quad (9)$$

is minimum. Using the associated map coefficients $\mathbf{w}_M^{s^*,\text{train}}$, the estimation of the KQI is produced. The complete procedure, implemented with the LMS algorithm, is presented as pseudocode in Algorithm 1.

The computational cost of the whole procedure and the memory requirements will strongly depend on the implementation choices. For instance, the number of arithmetic sums and multiplications for a typical LMS based algorithm is linear with the filter order L . Considering the multivariate approach with d_m measurements, it remains linear, but now with order $L \times d_m$. The same general observation applies to the clustering algorithms, such as k -means, where the complexity depends on the number of clusters, the dimension of the input data set (here L), and the number of elements of the input data (here $S \times d_m$).

Algorithm 1 KQI modeling using LMS multivariate adaptive filtering as pseudocode.

```

% Initialization
Define LMS parameter  $\mu$ , the order  $L$  of adaptive filters, and threshold  $\gamma_{MAX}$ 
Assign small random values to all  $\{\mathbf{w}_M^i\}^{train}$  and  $\{\mathbf{w}_P^i\}^{train}$  coefficients
% Training Phase
for Every time step  $n$  do
  % ADAPTIVE FILTERING
  Select the  $i^{th}$  category to update the coefficients  $\{\mathbf{w}_M^i\}^{train}$  and  $\{\mathbf{w}_P^i\}^{train}$ 
  Update mapping and prediction errors from Equations (3) and (5)
  Update filter coefficients using Equations (4) and (6)
  % TEST FOR NEW STATIONARY SEGMENT
  for All  $i$  current segments do
    Check if a new segment is needed using Equation (7)
    if  $\gamma_i \geq \gamma_{MAX}$  then
      Add a new segment, and select it for weight update
    end if
  end for
% Clustering Phase
Perform simultaneous clustering on the  $\{\mathbf{w}_P\}^{train}$  and  $\{\mathbf{w}_M\}^{train}$  sets
% Operating Phase
for Every time step  $n$  do
  Update prediction filter coefficients  $\{\mathbf{w}_P\}^{oper}$  using Equation (6)
  Calculate the distances from  $\{\mathbf{w}_P\}^{oper}$  to all entries in the clustered  $\{\mathbf{w}_P\}^{train}$  set
  Select index  $s^*$  corresponding to the least distance according to Equation (9)
  Use  $s^*$  to recover the corresponding  $\{\mathbf{w}_M^{s^*}\}^{train}$  mapping coefficients
  Use Equation (2) to estimate the desired KQI
end for

```

The next two sections show preliminary computational results with the purpose of assessing the framework concept. A systematic analysis, in different scenarios and with detailed statistical analysis of the results, is reserved for future investigations.

5. Concept Evaluation: Experiments, Results, and Discussion

The first set of computational experiments was conducted to assess the capability of the framework to model a chosen KQI from a single set of measurements. The data were based on the execution of the service file downloaded via FTP, and all experiments were conducted in the UMAHetNet network [30]. The dataset, as described by Herrera-Garcia et al. [1], the details of which are further described throughout this section, provides a key example of KQIs at the application layer under variable configurations and radio situations. It also allows for a direct comparison with other regression mechanisms applied in that work.

In this way, a single campaign dataset was used for training the system and to produce $\{\mathbf{w}_M^i\}^{train}$ and $\{\mathbf{w}_P^i\}^{train}$. The purpose of this setup is to validate the capability to perform segmentation and use the segments to recover the correct $\{\mathbf{w}_M^i\}^{train}$ from $\{\mathbf{w}_P^i\}^{oper}$. The same data are used in Figures 3–8.

5.1. Measurement Dataset

Five measurements and three KQIs from the FTP service were used, as depicted in Table 2. Among these, three of them are related to RF MAC/PHY layer parameters, namely RSSI, RSRP (reference signal received power), and RSRQ (reference signal received quality), and two of them are parameters related to the provisioning of the FTP service: network bandwidth and file size. KQIs to be modeled are transmission rate, session total time, and session setup time.

Table 2. Dataset used in the experiments from UMAHetNet. RSRP, reference signal received power; RSRQ, reference signal received quality.

Signal	Type	Variable
RSSI	RF PHY	m_n^1
RSRP	RF PHY	m_n^2
RSRQ	RF PHY	m_n^3
Network bandwidth	Parameter (4 values)	m_n^4
File size	Parameter (8 values)	m_n^5
Transmission rate	FTP KQI	k_n^1
Session setup time	FTP KQI	k_n^2
Session total time	FTP KQI	k_n^3

5.2. Data Pre-Processing

Due to the large difference in the order of magnitude of the measurements and parameters, a normalization procedure was applied in order to mitigate numerical bias. All data were normalized to the range $[-1, 1]$. It is worth mentioning that the final results may change if a different normalization procedure is applied or if another range is used. Furthermore, data from the lower layers are typically contaminated with noise and subjected to large variations due to the dynamic nature of the wireless channel. To this end, in these experiments, all PHY measurements (RSSI, RSRP, and RSRQ) were also filtered by a simple moving average filter with a window length of five samples.

5.3. Time-Series Segmentation and Prediction of Measurements

A graph of the number of segments, or categories, obtained in the training phase for the current example is depicted in Figure 3. The events that triggered the inclusion of a new segment were large prediction errors. In this case, the figure shows that the number of segments obtained at the end of the training phase was $S = 16$. It is interesting to note that, in this particular case, new segments were added due the changes of measurements m_n^4 and m_n^5 (see Figure 4).

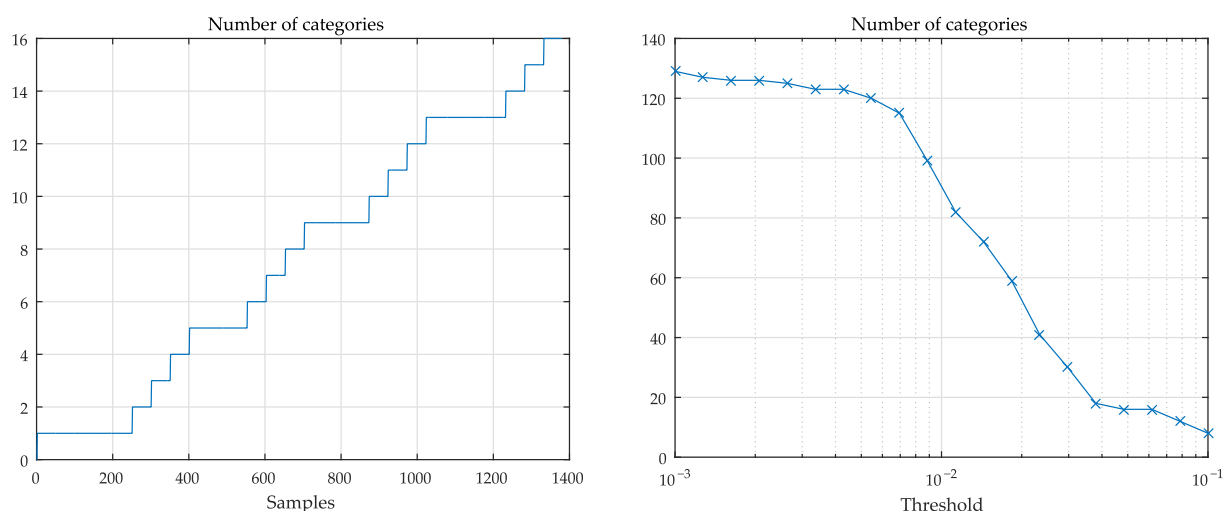


Figure 3. (left) evolution of the number of categories during the training phase. Compare with Figure 4. (right) typical behavior of the number of categories as a function of γ_{MAX} .

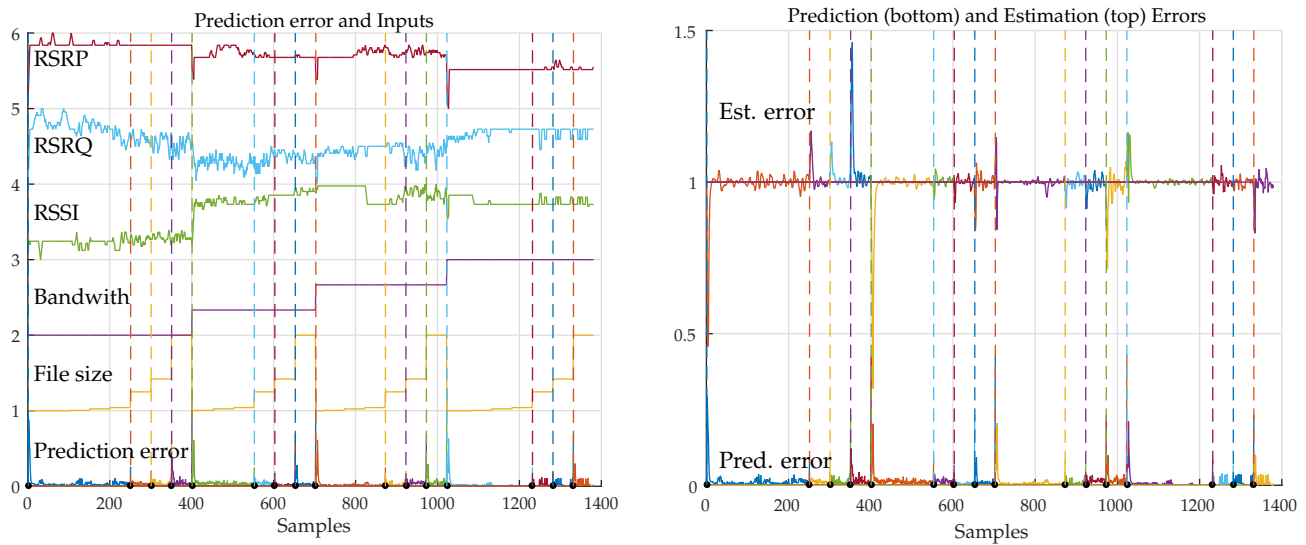


Figure 4. (left) input signals and sum of prediction errors (not in scale). (right) corresponding prediction and estimation errors. Dashed lines correspond to events with large prediction errors.

Figure 3 (right) also illustrates the typical dependence of the number of categories S according to the threshold γ_{MAX} , as a result of the procedure described in Section 4.2. As the number of categories is ultimately determined by the threshold γ_{MAX} , it is recommended to perform an intermediary step to optimize the threshold value. A brief analysis of this issue can be found in Section 5.9.

The prediction for m_n^j , $j = 1, 2, \dots, 5$, was implemented using one-step ahead predictors of order $L = 4$. Further adjustment of filter order and its effect on the final results are also expected to be performed in a further parameter optimization phase. As an example of the performance of the predictors, Figure 4 depicts, on the left, the five measurements and the sum of their correspondent prediction errors. At the bottom of the figure, it is possible to note the larger prediction errors $e_n^{p,j}$ at transitions of file size or bandwidth due to parameter changes. The graph on the right shows the same prediction errors at the bottom and, with a displacement of one unit in the y-axis, the correspondent mapping estimation errors e_n^M for each category.

At the end of the procedure, a dictionary with all $\{\mathbf{w}_P\}^{\text{train}}$ and $\{\mathbf{w}_M\}^{\text{train}}$ is produced. As an example, Figure 5, gives the first six sets of weights (first row: $\{\mathbf{w}_M\}^{\text{train}}$; second row: $\{\mathbf{w}_P\}^{\text{train}}$). It is possible to notice, even visually, that segments 1 and 2 are similar, as well as Segments 5 and 6. This justifies the possible need for a clustering procedure on the set of filter coefficients at the end of the training phase. Alternatively, it is possible to use robust techniques with the objective of avoiding point anomalies that may unnecessarily increase the number of mappings.

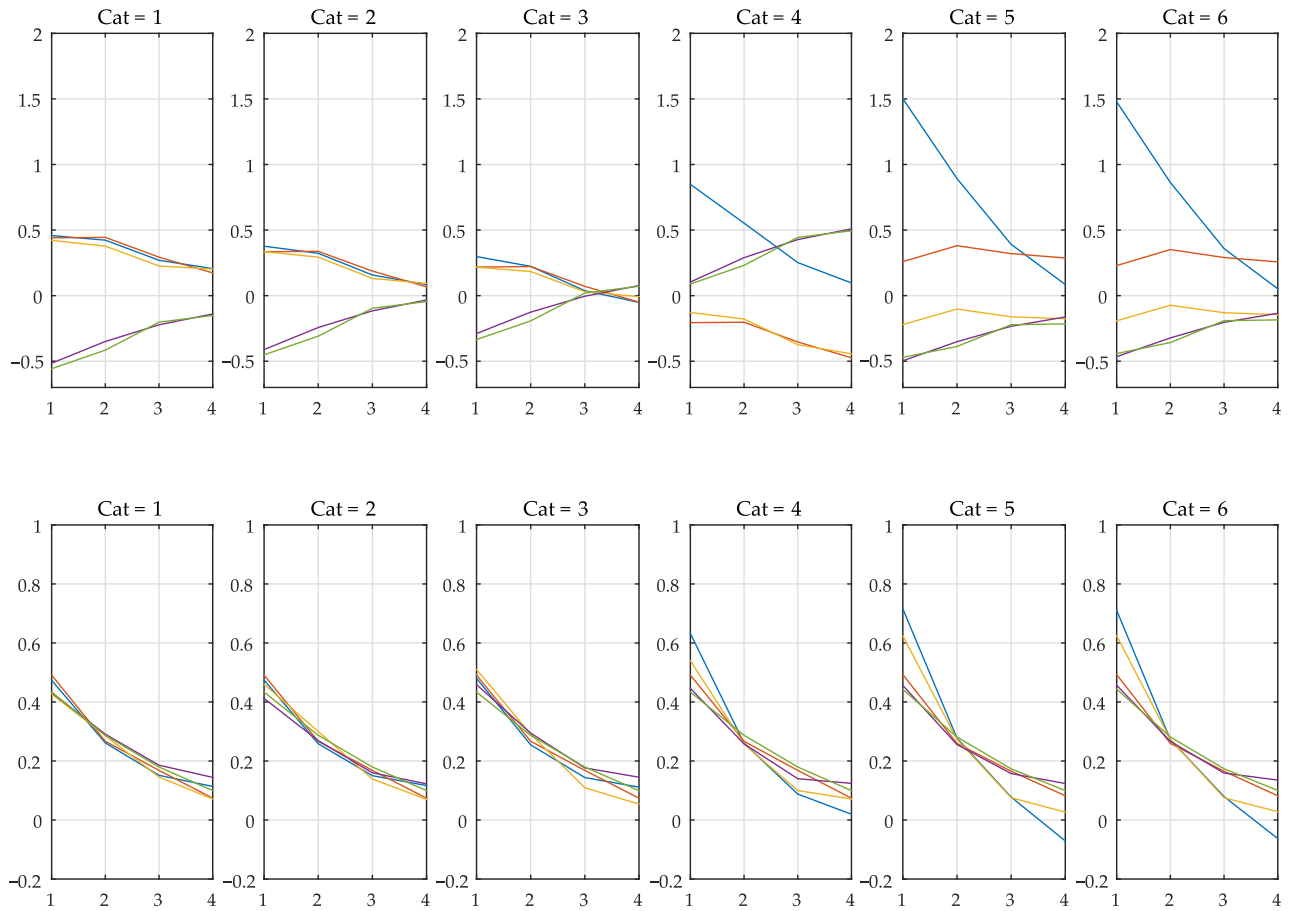


Figure 5. Map (first row) and prediction (second row) weights in the training phase, for the first six (from 16; see Figure 3) time-series segments. Each line in a graph represents the j -th group of coefficients associated with measurement m_n^j .

5.4. Clustering

For this example, the number of segments is $S = 16$. This is a relatively low number, and just for illustration purposes, we applied a k -means algorithm on the sets of $\{\mathbf{w}_P\}^{\text{train}}$ and $\{\mathbf{w}_M\}^{\text{train}}$. Now, the raw data used as input to the clustering algorithm must be associated with the joint set of weights, according to Equation (8).

As the filter order is L and the number of predictors is d_m , the elements of $\{\mathbf{w}_P\}^{\text{train}}$ and $\{\mathbf{w}_M\}^{\text{train}}$ were stored as $L \times d_m$ matrices. A first option would be to perform the clustering directly on the matrices, but here, we used a different method for visualization purposes. For each input vector \mathbf{x}_{in}^i , we extracted two numbers: its mean α^i and its standard deviation σ^i .

$$\mathbf{x}_{in}^i = \begin{bmatrix} \mathbf{w}_M^i \\ \mathbf{w}_P^i \end{bmatrix} \mapsto \begin{bmatrix} \alpha^i \\ \sigma^i \end{bmatrix} \quad (10)$$

The pair of numbers $[\alpha^i \ \sigma^i]^T$ was then submitted to a k -means clustering algorithm with five classes. Results are depicted in Figure 6, where it is possible to notice the grouping of sequential indexes, in accordance with Figure 3.

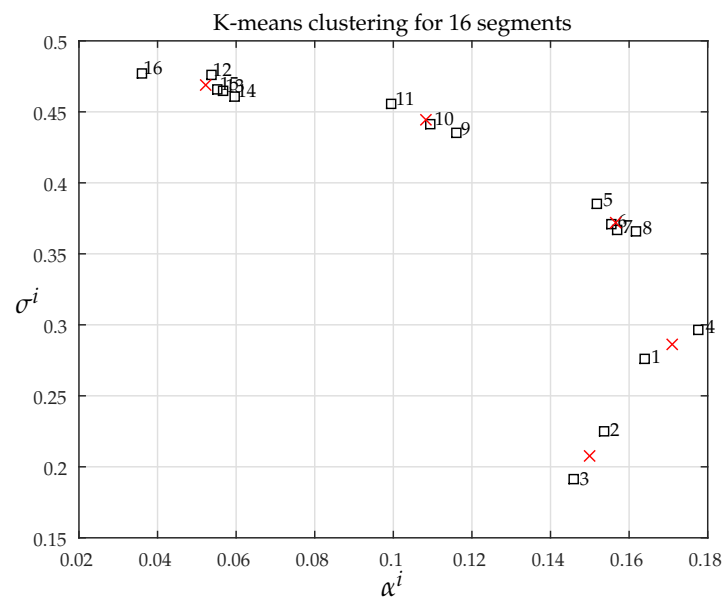


Figure 6. Result of k -means clustering with five categories. Cluster centers are indicated by red crosses.

At this stage, further analysis and improvements could be proposed, but have not been tested, such as performing the clustering taking into account the relative frequency of utilization of each segment.

5.5. Mapping and Estimation Errors

In Figure 7, it is possible to verify the performance of the LMS based multivariate adaptive filter that produces the set of coefficients $\{\mathbf{w}_M\}^{\text{train}}$. The chosen KQI is “session total time”, expressed as a linear combination of RSSI, RSSQ, RSSP, bandwidth, and file size. In this phase, it is expected that the estimation performs well, as long as the step-size parameter μ and the filter order are properly set. In the particular case studied here, the waveforms m_n^4 and m_n^5 have constant values, leading to potential numerical convergence problems if the filter order is too high and if stabilization procedures are not taken.

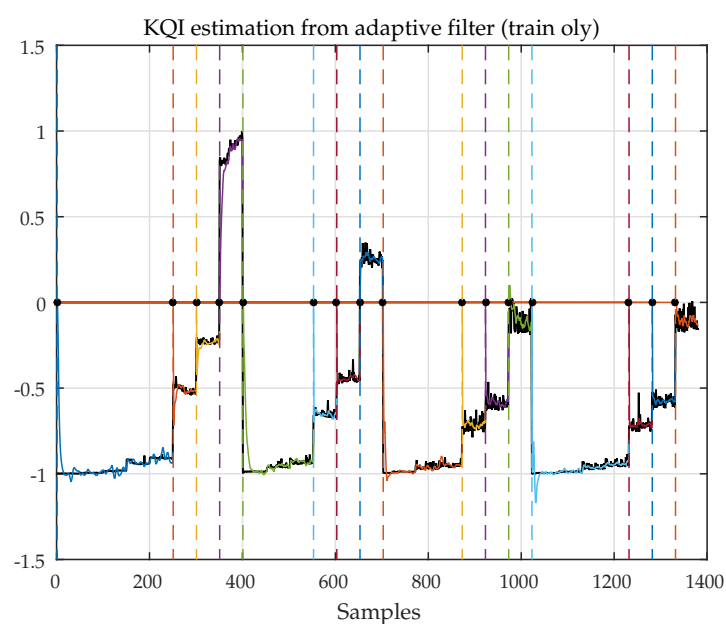


Figure 7. KQI “FTP session total time” as a function of RSSI, RSSQ, RSSP, bandwidth, and file size (colored by segment). Black: original data. Result obtained using the LMS based multivariate FIR adaptive filter.

5.6. KQI Reconstruction from the Prediction Coefficients

Figure 8 shows the session total time KQI recovered from the prediction coefficients. In the operation phase, coefficients $\{\mathbf{w}_p\}^{\text{oper}}$ are compared with $\{\mathbf{w}_p\}^{\text{train}}$. The closest set of coefficients in $\{\mathbf{w}_p\}^{\text{train}}$ is selected, and the corresponding $\{\mathbf{w}_M\}^{\text{train}}$ are used to recover the KQI from the inputs. The reliability of the system depends on the consistent pairing of $\{\mathbf{w}_p\}^{\text{train}}$ and $\{\mathbf{w}_M\}^{\text{train}}$: if there are similar prediction weights associated with different mapping weights, the ambiguities in the operation phase will result in poor recovery.

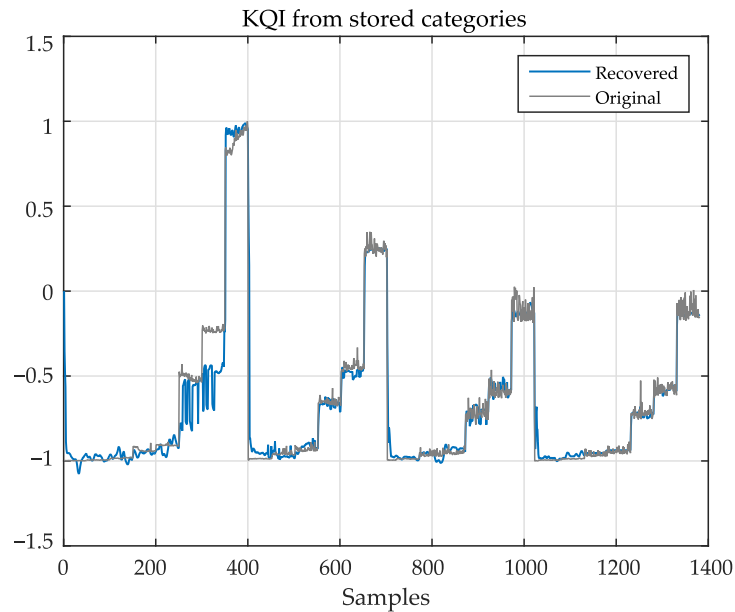


Figure 8. Blue: recovered KQI from $\{\mathbf{w}_p\}^{\text{oper}}$. Black: original data; blue: recovered KQI. Compare with Figure 7.

5.7. Setup for Validation

For comparison purposes, from the same dataset used so far, $n_{\text{MAX}} = 400$ samples were selected, now divided into disjoint training and testing subsets. As the predictor filters require a time window of length L from the input series to produce an estimate, the test samples cannot be randomly chosen as a single measurement M_n . With the objectives of (i) providing a preliminary assessment of the method and (ii) maintaining the sequential operation of the adaptive filters in the operation phase, one out of p samples from the dataset were taken sequentially through all measurements. This extraction results in two disjoint subsequences, both ordered in time, with lengths $n_{\text{MAX}} \cdot (p-1)p^{-1}$ and $n_{\text{MAX}} \cdot p^{-1}$. The first subsequence is used as the training set and the second as the testing set.

Figure 9 depicts results using LMS filtering. In this example, we set $p = 10$ to obtain the graph at the left, where it is possible to verify that the reconstructed signal follows the changes of the original KQI used for training. The coefficient of determination R^2 was used to evaluate the algorithm's performance. This figure of merit measures how much the variance of a dependent variable (i.e., the estimated KQI) can be predicted from an independent one (i.e., the measured KQI). This is a widely extended measure to quantify the quality of regression mechanisms [31]. R^2 absolute values are defined in the range $[0, 1]$, where a value of one implies a perfect fit.

In this scenario, the approach provided a value of $R^2 = 0.95$, showing a very high performance. In the same Figure 9, on the right, it is possible to verify some degradation of the performance if there is a reduction of p . As smaller values of p reflect an increase in the proportion between testing and training sets, this behavior is expected.

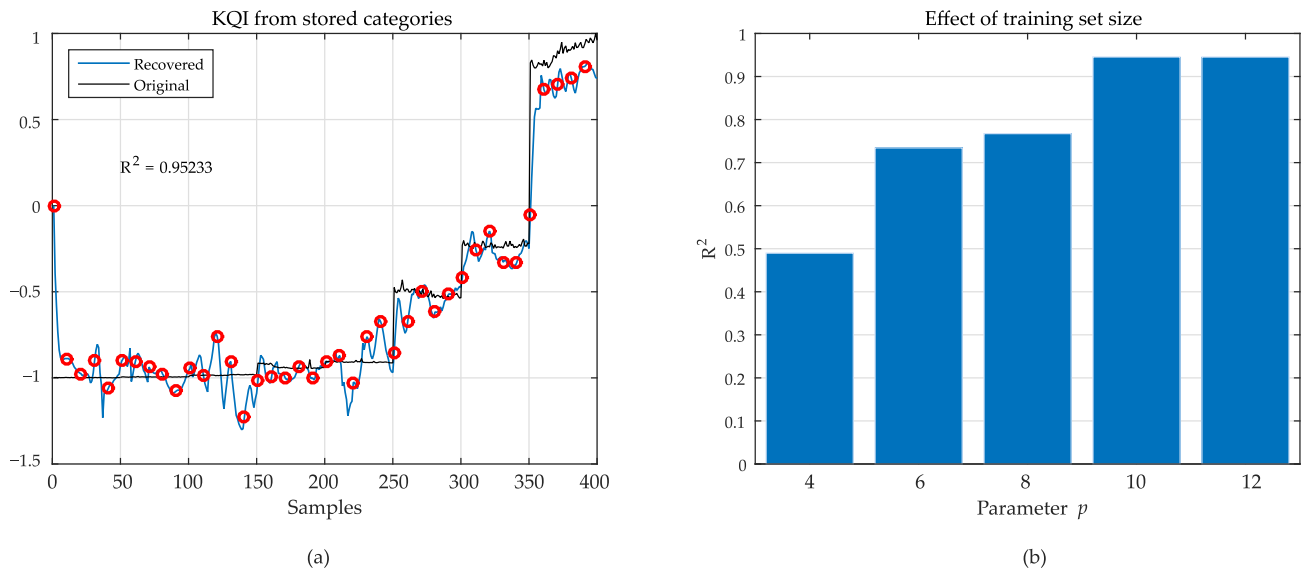


Figure 9. Recovered KQI from $\{\mathbf{w}_p\}^{\text{oper}}$ at left (a). Black: original data; blue: recovered KQI. Red dots are samples used for testing, not present in the training data. At right (b), the effect of the size of the training set in R^2 .

5.8. Optimization of Free Parameters

Table 3 contains a list of parameters that can be optimized in the training phase. From them, we selected the threshold γ_{MAX} , directly related to the number of detected segments.

In this evaluation, the number of segments was limited to a maximum value S_{MAX} , with five different values: 4, 8, 12, 16, and 20. For each value of S_{MAX} , the threshold was changed from 10^{-3} to 10^{-1} . The effect on the mean squared value of the estimation error:

$$e_n^{\text{est}} = \text{KQI}_n^{\text{target}} - \widehat{\text{KQI}}_n^{\text{target}} \quad (11)$$

is depicted in Figure 10.

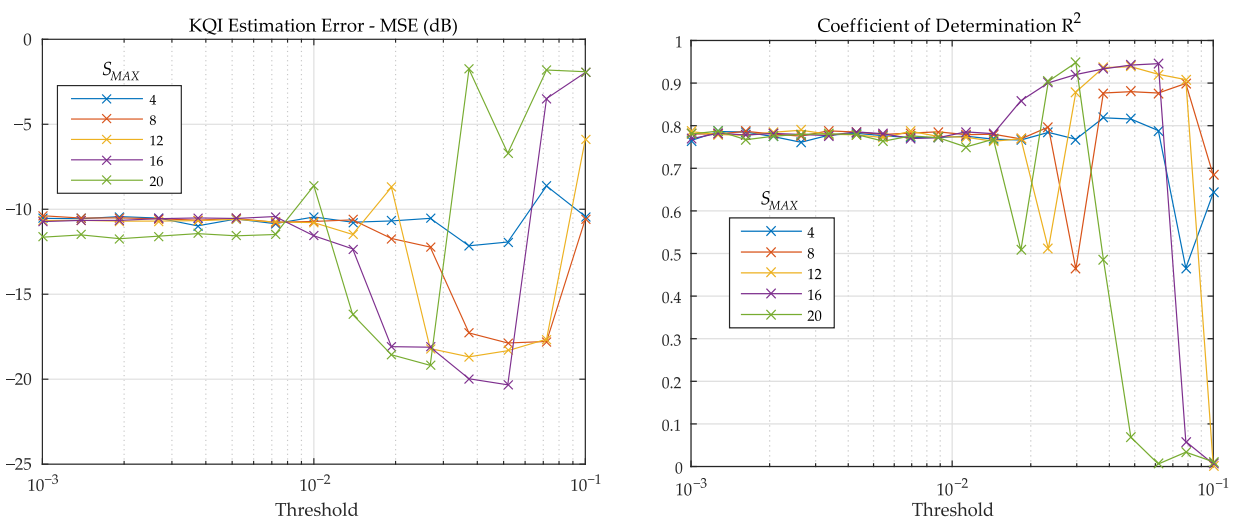


Figure 10. Estimation MSE (left) and coefficient of determination R^2 (right) as a function of prediction error threshold γ_{MAX} using different values for the maximum number of segments S_{MAX} .

Table 3. Free parameters to be optimized.

Parameter	Meaning	Algorithm
μ	LMS step size	LMS
L	Order of FIR filters	LMS
γ_{MAX}	Prediction error threshold	Segmentation

Optimization of γ_{MAX} is important to achieve good performance, but the other parameters from Table 3 also have an effect on R^2 (or equivalently, on MSE). In particular, the order of the FIR filters L cannot be made too large to avoid numerical instabilities. Furthermore, the LMS step μ has an optimal value that depends on the eigenvalues of the correlation matrix of the input data [32]. Results from Figure 10 were obtained with $L = 4$ and $\mu = 0.15$, and similar outcomes were observed with simulations within the ranges of $L \in (2, 8)$ and $\mu \in (0.05, 0.25)$.

Table 4 shows the results from this framework, labeled AMVTS (adaptive multivariate time-series), compared with other linear regression techniques using the R^2 parameter as a metric for comparison. We chose linear regression (LR), stepwise linear regression (SW-LR), and support vector regression (SVR), as detailed by Herrera-Garcia et al. [1], where the authors used the same dataset. As can be seen, AMVTS obtains values up to 0.95 for “transmission rate”, improving the good performance of SVR for this KQI by 6%. Moreover, AMVTS achieves the same performance with the best “session setup time” KQI technique, where the performance was 0.57. AMVTS performed worse only with the KQI “session total time”, where all linear techniques had poor results due to the fact that these KQIs’ values were nearly constant and completely dependent on the computational speed of the FTP server.

Table 4. Comparison with other linear regression techniques using the R^2 performance metric (results from Herrera-Garcia et al. [1]). AMVTS, adaptive multivariate time-series; SW-LR, stepwise linear regression.

KQI	AMVTS	LR	SW-LR	SVR
Transmission rate	0.95	0.66	0.89	0.82
Session setup time	0.57	0.14	0.34	0.57
Session total time	0.03	0.03	0.05	−0.01

5.9. Discussion and Alternative Approaches

This section identifies and discusses the open research lines generated by the present study, establishing a roadmap for future works and improvements. These focus on three lines: variations of the general framework, application of the framework to the SON context, and further research.

5.9.1. Variations on the General Framework

In the presented framework, the time-series segmentation treats service parameters as components of the multivariate time-series. If the number of combinations of the service parameters is discrete and relatively small, one alternative is to keep all measurements and KPIs as the only signals to be used as the inputs of the adaptive filters and use the information in the service parameters as pointers to specific subsets of $\mathbf{w}_{P,n}^{j,i}$ and $\mathbf{w}_{M,n}^{j,i}$. Clustering can then be performed to identify which group of parameters leads to similar representations.

As an example, in the specific case reported (refer to Table 2), only m_n^1, m_n^2, m_n^3 would be utilized to produce local approximations of $\mathcal{F}_{MK}(\cdot)$, and the $8 \times 4 = 32$ possible combinations of m_n^4 and m_n^5 would define 32 sets of $\mathbf{w}_{P,n}^{j,i}$ and $\mathbf{w}_{M,n}^{j,i}$.

Another interesting possibility is the utilization of the reverse mapping $\mathcal{F}_{MK}^{-1}(\cdot)$, using the KQIs as the input time-series and the measurements as the outputs. A content

service provider (SP), such as a streaming video service, may need to know if the wireless network can provide a pre-defined level of quality. In this scenario, the SP would infer the KPIs and the PHY/MAC measurements' values from the KQIs and adjust all service configurations accordingly.

The proposed KQI modeling can be modified by changing its main algorithms, but maintaining the general framework. Some suggestions are:

- Choice of adaptive filter algorithms: Different options for LMS and recursive least squares (RLS) families can be found in Diniz [33], Sayed [34] and Haykin [32]. It is also possible to use a non-linear adaptive filtering approach, such as radial basis functions (RBFs), multilayer perceptrons (MLPs), and Volterra filters. The FIR structure can also be changed to IIR (Infinite-Impulse Response) as detailed by Regalia in [35], order-recursive lattice filters, or a stated based approach using Kalman filtering and its variations as described by Haykin in [36].
- Detection of quasi-stationary segments: A suggestion to extend the proposed method is implementing prediction filters that compute not only samples one step ahead, but also p steps ahead. A vector of p prediction errors can be used to provide a more reliable decision on creating a new category.
- Clustering strategies: A possibility beyond the minimization of redundancy between different time segments is to perform the clustering, not within a segment, but among the measurements m^j , to reduce the input dimensionality d_m .
- Different options to build representations of \mathcal{M} and \mathcal{K} spaces. The proposed method builds the feature representation of measurements and KQIs via adaptive filter coefficients. These are not the only option, and the efficiency of other indirect features can be investigated. Immediate suggestions based on the speed of calculations are spectral analysis via FFT (with a link to segmentation as pointed by Perron in [37]) and wavelets. The choice of the best feature space is this multivariate time-series problem is not obvious, and a deeper look at the subject may be worth the effort.

5.9.2. Application of the Framework to the SON Context

Self-healing (SH) techniques can make use of the proposed algorithms. The network “homeostasis” is provided through a series of self-regulation mechanisms through closed-loop mechanisms, following the same general schematic of Figure 1.

A first impression of the direct applicability of the proposed framework for SH purposes is the use of $\mathcal{F}_{KS}(\cdot)$, to monitor the system status in the eNB scope. Once the KQI time-series is obtained, the same adaptive prediction and mapping approach can produce a time-series \mathbf{s}_n representing the network status. After proper identification of common faulty states, self-recovering procedures can then use these states to circumscribe a faulty node's effects in its neighborhood. The monitoring of the current status and prediction of its future values also allows the change of posture, from a reactive standpoint reacting to a failure or misconfiguration, to a functional perspective seeking automated preventive actions.

Closing the loop via $\mathcal{F}_{SA}(\cdot)$ is challenging due to several reasons:

- The temporal granularity of the sets \mathcal{S} and \mathcal{A} can be very different, and the information loopback cannot be unstable. This reinforces the need for reliable maps obtained from $\mathcal{F}_{MK}(\cdot)$ and $\mathcal{F}_{KS}(\cdot)$.
- The actions should work in harmony with collaborative techniques, such as coordinated multipoint (CoMP), ICIC (inter-cell interference cancellation), and network coding.
- One single status \mathbf{s}_n may be related to a set of actions in \mathcal{A} . In this scenario, the time horizon of observation of \mathbf{s}_n must be properly tuned such that a specific sequence of states is properly mapped to their correspondent sequences in \mathcal{A} .
- The action performed in a specific cell may affect its neighboring cells. One interesting technique that could be assessed to cope with the spatial nature of effects is adaptive diffusion mechanisms [38], which optimize cost functions over a network of nodes.
- Typical networks are deployed in a multi-vendor environment, and node parameters are usually vendor-specific. Furthermore, the management information base (MIB) of

network nodes may have hundreds of parameters. In this scenario, full automation can be costly to implement and face resistance from the service providers.

5.9.3. Further Research

Two topics still need a detailed assessment of their technical feasibility. The first is focused on the measurements and database building via a surplus of computing power at network nodes to provide better KPIs and KQIs. Some $\mathcal{F}_{MK}(\cdot)$ maps can be built as background processes called by the operating system at the network mobile nodes. This decentralized approach requires ML algorithms and auxiliary databases in Figure 1 to be, if not entirely, at least partially distributed. The OAM wireless system could be responsible for implementing other intensive data processing layers to cope with the complex interrelationships of information, such as spatial interdependence and user profiles. Related to this front line, another interesting possibility is to use the Age of Information concept [39] to relax the implicit constraints present in the recursive weight update of the multichannel adaptive filters.

A second topic is the investigation of new mappings from KQI to quality of experience (QoE). As quality is inherently service-dependent and possesses a subjective nature, QoE models are built typically for a specific traffic type, such as the ITU recommendations for video services [40]. If \mathcal{Q} is a comprehensive set or a representative indicator of the user's final experience, a suggestion would be the search for new mappings $\mathcal{F}_{KQ}(\cdot)$ from KQIs \mathcal{K} to \mathcal{Q} using the same adaptive modeling and segmentation strategies adopted to build $\mathcal{F}_{MK}(\cdot)$.

6. Conclusions

The proposed method can produce KQI estimates from network measurements, with a performance comparable to other linear regression models. Its general framework allows flexibility to build the measurement-to-KQI maps by utilizing other choices than LMS based adaptive FIR filters, such as RLS based ones or non-linear adaptive filtering. In a non-stationary environment, the continuous generation of new KQI models is possible, but as the size of the atlas required to represent different mappings increases, we recommend using clustering techniques to reduce the number of parameters. Its applicability to the SON context is also possible, where the proper handling of the measurements timestamps could be adopted under the lens of the Age of Information paradigm [41], integrating the concept of the freshness of the data in the construction of the models and utilizing a restricted set of actions to be performed at the nodes before closing the information loop.

Author Contributions: Conceptualization, L.A., S.F., and C.B.; methodology, L.A., S.F., and C.B.; software, L.A. and C.B.; validation, S.F., L.A., and E.B.; formal analysis, L.A., S.F., and C.B.; investigation, L.A., S.F., C.B., and E.B.; resources, R.B., L.A., and S.F.; writing—original draft preparation, L.A. and S.F.; writing—review, editing, L.A., S.F., R.B., C.B., and E.B.; supervision, L.A., R.B., and S.F.; project administration, L.A., S.F., and R.B.; funding acquisition, R.B., S.F., and L.A. All authors read and agreed to the published version of the manuscript.

Funding: This work was partially funded by Junta de Andalucía and ERDF in the framework of the projects IDADE-5G (UMA18-FEDERJA-201, “Programa Operativo FEDER Andalucía 2014–2020”) and SMART - “Sistema de Monitorización y análisis Automático de Redes de Telecomunicación” (“Incentivos a los agentes públicos del Sistema Andaluz del Conocimiento, para la realización de proyectos de I+D+i, en el ámbito del Plan Andaluz de Investigación, Desarrollo e Innovación PAIDI 2020”). It was also partially funded by Fundação de Apoio à Pesquisa do Distrito Federal (FAPDF, Edital 01/2019) and Universidad de Málaga (Comisión De Investigación Y Transferencia, Ayuda D.3).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Herrera-Garcia, A.; Fortes, S.; Baena Martinez, E.; Mendoza, J.; Baena, C.; Barco, R. End-To-End Network Management: Modeling Key Quality Indicators. *IEEE Veh. Technol. Mag.* **2019**, *14*, 76–84. [\[CrossRef\]](#)
- Baena, E.; Fortes, S.; Barco, R. KQI Performance Evaluation of 3GPP LBT Priorities for Indoor Unlicensed Coexistence Scenarios. *Electronics* **2020**, *9*, 1701. [\[CrossRef\]](#)
- Peñaherrera-Pulla, O.S.; Baena, C.; Fortes, S.; Baena, E.; Barco, R. Measuring Key Quality Indicators in Cloud Gaming: Framework and Assessment Over Wireless Networks. *Sensors* **2021**, *21*, 1387. [\[CrossRef\]](#)
- Mari-Altozano, M.L.; Luna-Ramírez, S.; Toril, M.; Gijón, C. A QoE-Driven Traffic Steering Algorithm for LTE Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 11271–11282. [\[CrossRef\]](#)
- 3rd Generation Partnership Project (3GPP), Self-Organizing Networks (SON): Concepts and Requirements. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2031> (accessed on 15 January 2021).
- 3rd Generation Partnership Project (3GPP), Physical Layer Procedures for Shared Spectrum Channel Access. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3435> (accessed on 15 January 2021).
- Ali, S.; Saad, W.; Rajatheva, N.; Chang, K.; Steinbach, D.; Sliwa, B.; Wietfeld, C.; Mei, K.; Shiri, H.; Zepernick, H.J.; et al. 6G White Paper on Machine Learning in Wireless Communication Networks. *arXiv* **2020**, arXiv:2004.13875.
- Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Solano, F.; Rendon, O. A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *J. Internet Serv. Appl.* **2018**, *9*, 1–99. [\[CrossRef\]](#)
- Wang, K.; Xu, P.; Chen, C.M.; Kumari, S.; Shojafar, M.; Alazab, M. Neural Architecture Search for Robust Networks in 6G-enabled Massive IoT Domain. *IEEE Internet Things J.* **2020**. [\[CrossRef\]](#)
- Fiandrino, C.; Zhang, C.; Patras, P.; Banchs, A.; Widmer, J. A Machine-Learning-Based Framework for Optimizing the Operation of Future Networks. *IEEE Commun. Mag.* **2020**, *58*, 20–25. [\[CrossRef\]](#)
- Wang, E.K.; Liu, X.; Chen, C.M.; Kumari, S.; Shojafar, M.; Hossain, M.S. Voice-Transfer Attacking on Industrial Voice Control Systems in 5G-Aided IIoT Domain. *IEEE Trans. Ind. Inform.* **2020**, *1*. [\[CrossRef\]](#)
- Baena, C.; Fortes, S.; Baena, E.; Barco, R. Estimation of Video Streaming KQIs for Radio Access Negotiation in Network Slicing Scenarios. *IEEE Commun. Lett.* **2020**, *24*, 1304–1307. [\[CrossRef\]](#)
- Fan, G.; Shi, W.; Guo, L.; Zeng, J.; Zhang, K.; Gui, G. Machine Learning Based Quantitative Association Rule Mining Method for Evaluating Cellular Network Performance. *IEEE Access* **2019**, *7*, 166815–166822. [\[CrossRef\]](#)
- Fortes, S.; Aguilar-García, A.; Barco, R.; Barba, F.; Fernández-Luque, J.; Fernández-Durán, A. Management architecture for location-aware self-organizing LTE/LTE-A small cell networks. *IEEE Commun. Mag.* **2015**, *53*, 294–302. [\[CrossRef\]](#)
- Fortes, S.; Garcia, A.A.; Fernandez-Luque, J.A.; Garrido, A.; Barco, R. Context-Aware Self-Healing: User Equipment as the Main Source of Information for Small-Cell Indoor Networks. *IEEE Veh. Technol. Mag.* **2016**, *11*, 76–85. [\[CrossRef\]](#)
- Fortes, S.; Palacios, D.; Serrano, I.; Barco, R. Applying Social Event Data for the Management of Cellular Networks. *IEEE Commun. Mag.* **2018**, *56*, 36–43. [\[CrossRef\]](#)
- Fortes, S.; Baena, C.; Villegas, J.; Baena, E.; Asghar, M.Z.; Barco, R. Location-Awareness for Failure Management in Cellular Networks: An Integrated Approach. *Sensors* **2021**, *21*, 1501. [\[CrossRef\]](#)
- 3rd Generation Partnership Project (3GPP) Telecommunication management; Subscriber and equipment trace: Trace control and configuration management. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2009> (accessed on 15 January 2021)
- 3rd Generation Partnership Project (3GPP) Telecommunication management; Subscriber and equipment trace: Trace data definition and management. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2010> (accessed on 15 January 2021)
- 3rd Generation Partnership Project (3GPP) Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2025> (accessed on 15 January 2021)
- Bellanger, M. Adaptive Processing of Multidimensional Signals: From Principles to Simulation. In Proceedings of the IFAC Proceedings Volumes 4th IFAC Symposium on Adaptive Systems in Control and Signal Processing, Grenoble, France, 1–3 July 1992; Volume 25, pp. 499–504. [\[CrossRef\]](#)
- Bellanger, M. *Adaptive Digital Filters*, 2nd ed.; Signal Processing and Communications Series; Marcel Dekker: New York, NY, USA, 2001.
- Markou, M.; Singh, S. Novelty detection: A review. Part 1: statistical approaches. *Signal Process.* **2003**, *83*, 2481–2497. [\[CrossRef\]](#)
- Markou, M.; Singh, S. Novelty detection: A review. Part 2: neural network based approaches. *Signal Process.* **2003**, *83*, 2499–2521. [\[CrossRef\]](#)
- Barz, B.; Rodner, E.; Guanche, Y.; Denzler, J. Detecting Regions of Maximal Divergence for Spatio-Temporal Anomaly Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1088–1101. [\[CrossRef\]](#)
- Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection for Discrete Sequences: A Survey. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 823–839. [\[CrossRef\]](#)
- Widrow, B.; Stearns, S.D. *Adaptive Signal Processing*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1985.
- Jiang, R.; Fei, H.; Huan, J. A Family of Joint Sparse PCA Algorithms for Anomaly Localization in Network Data Streams. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 2421–2433. [\[CrossRef\]](#)

-
29. Munir, M.; Siddiqui, S.A.; Dengel, A.; Ahmed, S. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time-Series. *IEEE Access* **2019**, *7*, 1991–2005. [[CrossRef](#)]
 30. Fortes, S.; Santoyo-Ramón, J.A.; Palacios, D.; Baena, E.; Mora-García, R.; Medina, M.; Mora, P.; Barco, R. The Campus as a Smart City: University of Málaga Environmental, Learning, and Research Approaches. *Sensors* **2019**, *19*, 1349. [[CrossRef](#)]
 31. Hamilton, D.; Ghert, M.; Simpson, H. Interpreting regression models in clinical outcome studies. *Bone Jt. Res.* **2015**, *4*, 152–153. [[CrossRef](#)]
 32. Haykin, S. *Adaptive Filter Theory*, 3rd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1996.
 33. Diniz, P.S.R. *Adaptive Filtering: Algorithms and Practical Implementation*; Springer: Berlin/Heidelberg, Germany, 2007.
 34. Sayed, A.H. *Adaptive Filters*; Wiley-IEEE Press: Hoboken, NJ, USA, 2008.
 35. Regalia, P.A. *Adaptive IIR Filtering for Signal Processing Control*, 3rd ed.; CRC Press: New York, NY, USA, 1994.
 36. Haykin, S. *Kalman Filtering and Neural Networks*; John Wiley & Sons: New York, NY, USA, 2001.
 37. Perron, P.; Yamamoto, Y. Estimating and Testing Multiple Structural Changes in Linear Models Using Band Spectral Regressions. *Econom. J.* **2013**, *16*, 400–429.
 38. Chen, J.; Sayed, A. Diffusion Adaptation Strategies for Distributed Optimization and Learning Over Networks. *IEEE Trans. Signal Process.* **2012**, *60*, 4289–4305. [[CrossRef](#)]
 39. Kaul, S.; Yates, R.; Gruteser, M. Real-time status: How often should one update? In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2731–2735.
 40. International Telecommunication Union. *Parametric Bitstream-BASED Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport*; ITU-T P.1203 10/2017; International Telecommunication Union: Geneva, Switzerland, 2017.
 41. Costa, M.; Codreanu, M.; Ephremides, A. On the Age of Information in Status Update Systems With Packet Management. *IEEE Trans. Inf. Theory* **2016**, *62*, 1897–1910. [[CrossRef](#)]