

Document downloaded from:

<http://hdl.handle.net/10251/163292>

This paper must be cited as:

Canovas Solbes, A.; Rego Mañez, A.; Romero Martínez, JO.; Lloret, J. (2020). A robust multimedia traffic SDN-Based management system using patterns and models of QoE estimation with BRNN. *Journal of Network and Computer Applications*. 150:1-14.
<https://doi.org/10.1016/j.jnca.2019.102498>



The final publication is available at

<https://doi.org/10.1016/j.jnca.2019.102498>

Copyright Elsevier

Additional Information

A Robust Multimedia Traffic SDN-Based Management System using Patterns and Models of QoE Estimation with BRNN

ALEJANDRO CANOVAS, ALBERT REGO, OSCAR ROMERO, JAIME LLORET, (SENIOR MEMBER, IEEE)

Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València. Carrer del Paranimf, 1, 46730 Grau de Gandia, València. Spain.

Corresponding author: Alejandro Canovas (e-mail: alcazol@posgrado.upv.es).

ABSTRACT Nowadays, network infrastructures such as Software Defined Networks (SDN) achieve a huge computational power. This allows to add a high processing on the network nodes. In this paper, a multimedia traffic management system is presented. This system is based on estimation models of Quality of Experience (QoE) and also on the traffic patterns classification. In order to achieve this, a QoE estimation method has been modeled. This method allows for classifying the multimedia traffic from multimedia transmission patterns. In order to do this, the SDN controller gathers statistics from the network. The patterns used have been defined from a lineal combination of objective QoE measurements. The model has been defined by Bayesian regularized neural networks (BRNN). From this model, the system is able to classify several kind of traffic according to the quality perceived by the users. Then, a model has been developed to determine which video characteristics need to be changed to provide the user with the best possible quality in the critical moments of the transmission. The choice of these characteristics is based on the quality of service (QoS) parameters, such as delay, jitter, loss rate and bandwidth. Moreover, it is also based on subpatterns defined by clusters from the dataset and which represents network and video characteristics. When a critical network situation is given, the model selects, by using network parameters as entries, the subpattern with the most similar network condition. The minimum Euclidean distance between these entries and the network parameters of the subpatters is calculated to perform this selection. Both models work together to build a reliable multimedia traffic management system perfectly integrated into current network infrastructures, which is able to classify the traffic and solve critical situations changing the video characteristics, by using the SDN architecture.

KEYWORDS Software Defined Network (SDN); Machine Learning; QoE; Traffic Multimedia Pattern;

I. INTRODUCTION

Nowadays video streaming has been wide developed and is present in most of the networks. This type of traffic, due to its bandwidth requirements, causes a high increase of traffic over the networks, making more difficult to manage and optimize the network performance. Furthermore, for video streaming is critical to have good network performance in order to provide QoS and guarantee an adequate QoE for the final user. Therefore, traffic modeling has a great interest. In this context, SDN based networks offers many advantages for video streaming over the network. Control panel and data panel are separated in SDN, offering a powerful solution to manage traffic network, instead of standard switching or routing decides how to forward packets. The SDN Controller gets statistics and other parameters from SDN switches. With this information it is possible to know the status of the network, i.e., link utilization, link capabilities, packets being transmitted, paths to destinations, etc. Specifically,

controlling and modifying the network and video streamed parameters, the QoE and QoS required values can be provided.

Thus, processing all this information with IA and machine learning we obtain a powerful tool to process the network traffic packets and manage the network. Hence, in this paper we propose a multimedia traffic management model based on QoE estimation model and traffic pattern classification for SDN networks. The QoE estimation model, which is based on Bayesian Regularized Neural Network (BRNN) using the Levenberg-Marquardt algorithm, will allow wus to make multimedia traffic classification. We also propose another model that can determine what video characteristics could be changed in order to improve the QoE in critical situations. The selection of these characteristics are based on subpatterns, which represents different network and video streaming conditions. The network conditions are defined by QoS parameters such as delay, jitter, packet loss or bandwidth. The video characteristics are defined by parameters like bitrate, framerate, width, height, etc. Both models together provide a

robust multimedia traffic management system that can be integrated in present networks to classify network traffic and solve critical situations by changing video streaming characteristics.

This study is an extension of [1]. Deeper study and further analyzes have been added. The starting point is the database used in [1] with sampling frequencies based on GOP (Group of Picture) of 2 seconds. We will also use different types of videos depending on its characteristics, format and codification as we used in [1]. By performing a deeper and better QoE estimation model, we have developed a high reliable multimedia traffic management system. The classification model based on BRNN (Bayesian Regression Neural Network) presented now will provide better results than the obtained in [1]. In order to improve the results we will analyze several parameters such as weights initialization, number of neurons of the hidden layer, number of hidden layers of the neural network (NN), gamma activation function, the weighting factor C, etc. Then, in order to try to improve the results of accuracy and recall, as well as adding more critical traffic cases. Then, additional study of traffic parameters will be made with the aim to analyze the relation among these parameters. Then, to try to improve the results of accuracy and recall, we have focused especially on the cases of critical traffic, in such a way that the model learns from an efficient number of samples of this type without reaching overfitting. For this purpose we have used cross-validation learning methods. Finally, we have studied the correlation between network parameters and video flow, to develop a model that allows us to redefine the characteristics of the video flow, thus improving QoE in critical traffic situations.

The remainder of this article is organized as follows. Section II presents the related work. In Section III, multimedia traffic classification systems are analyzed. QoE adjustment is analyzed in Section IV. In Section V, the network architecture is described and the system implementation is discussed. Finally, conclusion and future work are included in Section VI.

II. RELATED WORK

In the last years, due to the increase of network traffic, many works have been proposed to classify multimedia traffic in order to guarantee QoS and QoE. In [1] authors proposed a multimedia traffic classification model based on patterns by using video streaming and network characteristics as input parameters and video quality evaluation measures as output. They use NQI, VQM, SSIM, and PSNR metrics to conform traffic patterns. Different learning methods are studied to find out which of them provides the best result. These methods are based on NNs, vector support machines, statistics and the nearest neighbors. In [2] is presented an experimental comparison of the performance of SNR, SSIM, NQI and VQM. Depending on several parameters, the best performance can be obtained from a different method, but in general, best results are given by VQM.

In [3], it is developed a novel framework called DeepQoE to predict video QoE. The end-to-end framework first uses a combination of deep learning (DL) techniques (e.g., word embedding) to extract generalized features. Next, these features are combined and fed into a NN for representation learning. Such representations serve as inputs for classification or regression tasks.

Authors in [4] try to solve the bandwidth allocation problem in cloud computing data center networks using Software Defined Networking (SDN) paradigm. This method makes use of machine learning techniques to classify the incoming traffic flows in real-time while ensuring game flows are prioritized over others. In [5] authors use machine learning techniques to try to demonstrate how QoS metrics can be exploited to accurately estimate and predict key QoE factors. They show that context information on network congestion and basic characteristics on video streams further improves predictions.

Some authors, like in [6], evaluate the QoE for Internet of Things (IoT) applications, with particular attention to multimedia traffic. They introduce the concept of Multimedia IoT (MIoT) and define a layered QoE model aimed at evaluating and combining the contributions of each influence factor to estimate the overall QoE in MIoT applications. In [7], authors try to predict video QoE based on information directly extracted from the network packets using a deep learning model. The QoE detector is based on a binary classifier (good or bad quality) and uses seven common classes of anomalies. The proposed classifier is based on a combination of a Convolutional Neural Network (CNN), recurrent neural network, and Gaussian process classifier. A deep subjective study of video quality has been performed in [8]. It is also evaluated the performance of several state-of-the-art, publicly available full-reference video quality assessment algorithms.

Authors in [9] study the adaptive multimedia traffic control mechanism leveraging Deep Reinforcement Learning (DRL) that combines deep learning with reinforcement learning, which learns from rewards by trial-and-error. This mechanism is able to control multimedia traffic directly from experience without referring to a mathematical model.

Authors in [10] analyze several researches about application of Machine Learning (ML) techniques to IP traffic classification. They also discuss a number of key requirements for the employment of ML-based traffic classifiers in operational IP networks. In [11], authors employ three supervised machine learning algorithms, Bayesian Networks, Decision Trees and Multilayer Perceptrons for the flow-based classification of six different types of Internet traffic. Their experiments show that ML algorithms such as Bayesian Networks and Decision Trees are suitable for Internet traffic flow classification at a high speed.

In [12] authors apply a Naïve Bayes estimator to categorize traffic by application, showing its high level of accuracy. In this approach, they use samples of well-known traffic to allow the categorization. A fully automated Packet Payload Content

(PPC) based network traffic classification system is presented in [13]. The proposed system learns new application signatures in the network where classification is desired. Furthermore, that system adapts the signatures as the traffic for an application changes.

In [14] it is analyzed how different factors contribute to the Quality of Experience (QoE), in the context of video streaming delivery over cloud networks. The authors describe different methods that are often used to collect QoE datasets in the form of a Mean Opinion Score (MOS). Machine Learning methods are then used to classify a preliminary QoE, and six classifiers are evaluated and determined. Also, in [15] the authors present a user quality of experience prediction algorithm which extracts features based on user traffic pattern parameters such as bit-rate, resolution, frame rate, etc. They use three different feature selection algorithms and six different classifiers in order to optimize the features set and the corresponding ML algorithms.

L. Huixian et al. develop a traffic-analysis method using an unsupervised ML technique, where flows are automatically classified by exploiting the different statistics characteristics of flow [16]. Authors in [17] develop a machine learning based traffic classification based on statistical properties while optionally de-coupling flow classification and treatment. This de-coupled architecture allows centralized traffic classifiers to control traffic filtering and shaping by diversely located, low-performance network devices.

III. INTELLIGENT MULTIMEDIA TRAFFIC CLASSIFICATION

In this section is explained how the development of the multimedia traffic classification model was carried out. First, a QoE classification model is developed. Then, according to the results, we have analyzed and evaluated which model works better for the multimedia traffic classification.

A. DATABASE PRE-PROCESSING

The database used in this paper to develop the intelligent multimedia traffic classification system comes from the work done in [1]. It is composed by 2741 samples with 14 features per sample. Each sample is labeled according to the patterns obtained. Moreover, their values are obtained from objective QoE measurements which has a high correlation with the subjective QoE, like ISSM, VQM and NQI. PSNR is also used due to its importance for the study. In order to extract every single sample, several video transmission with different network and video conditions were sent. Each video transmitted were divided into GOPs (group of pictures) equivalents to a 2s video clip. The average values of some characteristics were extracted from each video clip. Some of these features are: frame count, stream size, proportion stream, footer size, bitrate, width, height, frame rate and minimum frame rate. Furthermore, some network parameters like delay, jitter, packet loss, jitter variation and bandwidth for each time

interval were extracted. For each video clip, several objective measurements were extracted by comparing the source and the destination videos so that the class labels could be generated. Depending on the correlation with the subjective QoE, a weight was assigned to each one of them and an average was calculated.

B. QoE ESTIMATION MODEL

An important aspect of this study is to obtain an accurate QoE estimation model with which a high-reliable multimedia traffic management system could be developed. In [1] only a classification model of the type of traffic was obtained. This classification model is addressed in the subsection C. In this paper, a wide study about QoE estimation is firstly performed to improve the results of the traffic classification model. For this study, several classification and regression algorithms have been evaluated. The problem can be addressed from two different points of view. It could be either a classification problem where discrete (1:1:5) MOS values are obtained as results or a regression problem where the result is a set of continuous values (1, 5). In this work, the problem has been turned into a classification problem by transforming continuous MOS values into discrete values. This means, the output continuous values of the regression model have been rounded to discrete values. Therefore, the regression problem changes into a classification problem. This allows to use regression models and check how good they suit the problem. In order to obtain the models, the following algorithms have been used:

Classification models based on:

- Support vector machines (CSVM)
- Neural networks (CNN)
- Naïve Bayes (CNB)
- Decision trees (CDT)
- Discriminant analysis (CDA)
- K- Nearest neighbors (CKNN)

Regression models based on:

- Gaussian process (RGP)
- Non-linear (RNL)
- Decision trees (RDT)
- Neural networks (RNN)

In order to carry out this study, MATLAB and the available machine learning libraries have been used. The array of features of each sample has been used as entry parameters in the supervised learning models, and the MOS is used as the class label. From the whole simple set, 2441 samples have been used for training and 300 for testing. Cross-validation has been used to avoid the overfitting and, thus, generalize and validate the models. Concretely, 10-fold cross-validation has been used.

The procedure of model obtaining has been as follows. Firstly, the different models have been evaluated using the default parameters. Then, the models with the best classification results have been polished to find a local or global minimum that improves the classification results.

Several scripts have been created to analyze and evaluate several learning parameters like kernel function, optimization coefficients, stopping criteria, etc. In order to evaluate the classification results of each proposed automatic supervised learning algorithm, the confusion matrix has been calculated. With the obtained values, the precision, recall, accuracy, F Score (F1) and proximity values have been calculated. These values are displayed in Fig. 1. Proximity has been included to obtain a less strict and rigorous accuracy measurement. Precision indicates how good the system can be to detect the class that is being analyzed. This means, with regard to the samples of that class, how many has been labeled correctly. Proximity allows to consider whether the system not only detects that class, but also if it is close to do it. This measurement gives an idea of how quasi-accurate the model is. It can be a big help to select the management system.

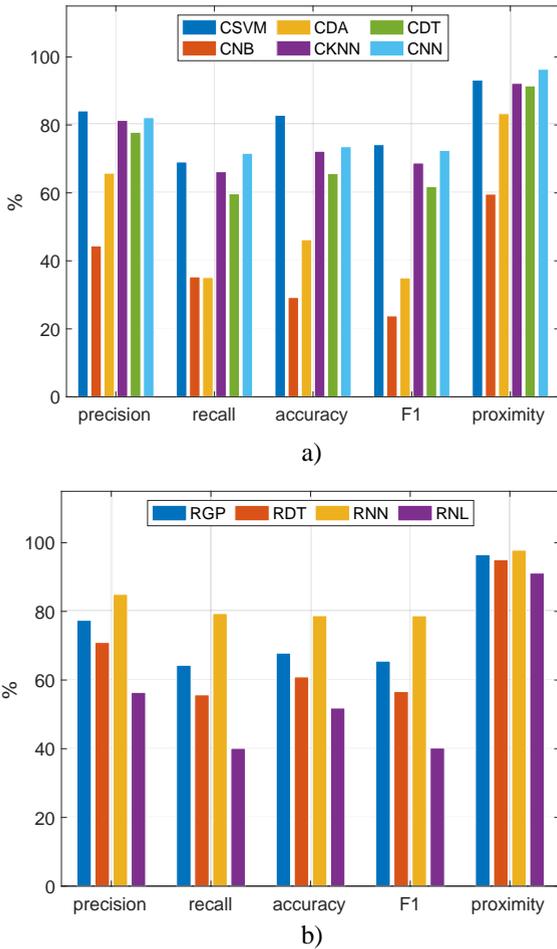


FIGURE 1. Learning results for QoE estimation in the different classification (a)) and regression (b)) models according to the confusion matrix.

As it is depicted in Fig. 1, the first measurement is precision, which indicates the positive cases detected correctly. RNN and CSVM models are the ones with the best precision results. They have obtained similar results, with an 84%. Moreover, CKNN, CDT and RGP present good results

with an 81% for CKNN and a 77% for CDT and RGP. Worse results are obtained from the models based on CNB and CDA with 44% and 65% respectively. This poor results for CNB and CDA are also obtained in the other measurements, except proximity, being unable to be greater than 46%. Since CNB presents good results with non-parametric data, it can be concluded that this is consequent with the current problem and it is not an issue at all. As regards regression (Fig. 1 b), RDT and RGQ present the worst results, except for proximity, like in the previous case. This high value in proximity indicates that the model fails but it is close to be valid. On the other hand, regarding recall, i.e. the capacity of detecting properly the positive cases, RNN presents the best results with a 79%. CSVM and CKNN present similar results with 69% and 66% respectively. The next parameter, accuracy, shows how good the system is regarding the total correct estimations. CSVM presents the best results regarding accuracy (82%) followed by RNN (78%) and CKNN (72%). In this case, unlike the other measurements, CSVM presents a better result than RNN because all the right choices, from the whole set of possibilities (TP, TN, FP and FN), have been taken into account. This difference in the results only happen in those two scores. F1 has been another parameter tested. This parameter combines precision and recall in the same measurement to indicate the quality of the method. This score emphasizes the difference between the algorithms that presents good results and those which not. CSVM and RNN are the models with best results as regards this score. Finally, as regards proximity, RNN stands out with a 97%. However, the other regression methods achieve high results with 96%, 95% and 91% for RGP, RDT and RGQ respectively. This can be observed in Fig. 1 b). CSVM, CKNN, CDT present similar values, 93%, 92% and 91% respectively. This score is quite important for this work because it shows that if the model fails with the MOS prediction, it will be close. Therefore, it will not return a MOS of 5 when the actual MOS is 2, for instance. This aspect is really important to improve or polish the system.

As a conclusion of the results obtained from the confusion matrix, the most adequate models with regard to the problem are the ones based on CSVM and RNN. In Fig. 2, the behavior of both models in the test can be analyze deeply. RNN has obtained less prediction errors than CSVM, facing the problem slightly better. When MOS = 5, both systems predict a right result, RNN with 93% and CSVM with 97% of right estimations. In the cases when RNN has failed, it has predicted MOS = 4 except in one of them, where it has predicted MOS = 3. CSVM has predicted MOS = 4 once and MOS = 2 in the other mistakes. When MOS = 4, CSVM predicts correctly in the 75% of cases, with MOS = 3 with 75% too and with MOS = 1 with 62%. It should be highlighted that, with MOS = 1, the number of errors increases significantly up to 14, a 38% of the total cases with MOS = 1. RNN predicts correctly when MOS = 4 with 70%, 77% when MOS = 3, with 78.5% when MOS = 2 and with 70% when MOS = 1. As it can be observed, the highest difference is located when MOS = 1, where RNN

predicts better. It is a remarkable problem that CSVM predicts MOS = 4 and MOS = 5 with a 50% of probability in that case, when MOS presents an actual value of 1. These facts correspond to samples 3, 6, 8, 10, 14, 27 and 37. RNN predicts only in an 18% of times MOS = 5 (samples 27 and 37), while in the other times it predicts MOS = 1. With a MOS = 2, RNN only fails returning a predicted MOS greater than 3 twice (14.2%, samples 38 and 40). However, CSVM returns a predicted MOS greater than 3 in 9 of the 12 mistakes, a 75% of the total, in samples 2, 4, 5, 6, 40, 44, 45, 46 and 58. This result is really bad for CSVM because in all of these cases, the classification model described in the next subsection would not detect a critical traffic. When MOS = 3, RNN predicts a MOS greater than 3 in 6 of the 11 samples (4, 9, 26, 31, 42 and 43). CSVM predicts this in 14 of 21 (66.6%). Finally, when MOS = 4, the results obtained are quite similar for both models. With RNN 6 of the 16 samples (37.5%) were estimated with a MOS lower than 4 and with CSVM 3 of 13 (23%). Only the sample 32 were estimated with a MOS lower than 3 with RNN, while SVM estimated none with a MOS lower than 3.

In conclusion, the estimation level of both models are quite similar. However, RNN presents better results because the wrong estimated values are not as far from the right ones as the presented by CSVM. This supports the results obtained from the proximity measurement. This fact is really important for the traffic classification, as it is explained in the next subsection.

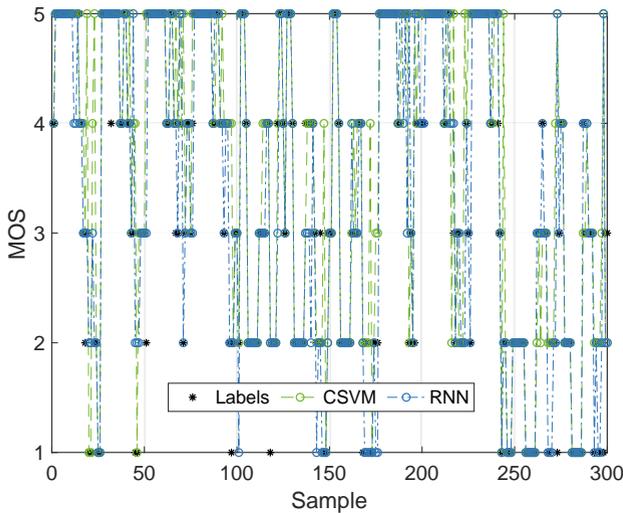


FIGURE 2. Test results of the QoE estimation for CSVM and RNN models.

C. CLASSIFICATION MODEL OF THE TYPE OF TRAFFIC

The classification model presented in this paper is based on the proposal shown in [1]. This model classifies the type of traffic into critical or not critical. The traffic is considered critical when the MOS is less than or equal to 3. This consideration was taken due to the subjective QoE, by watching a set of videos and checking when the perception of

the quality started to be low. Since the videos were labeled according to the objective QoE, it could be checked that this subjective value matched a MOS of 3 in the model presented in [1]. In order to evaluate the different learning models presented in last subsection, the following criteria are used:

- TP (True-Positive): The model classifies the traffic as critical and it matches.
- TN (True-Negative): The model classifies the traffic as non-critical and it matches.
- FP (False-Positive): The model classifies the traffic as critical but it does not match.
- FN (False-Negative): The model classifies the traffic as non-critical, but it is critical

After calculating the values of TP, TN, FP and FN, the precision, recall, accuracy, and F1 measurements of every model were obtained. Obtained results are depicted in Fig. 3.

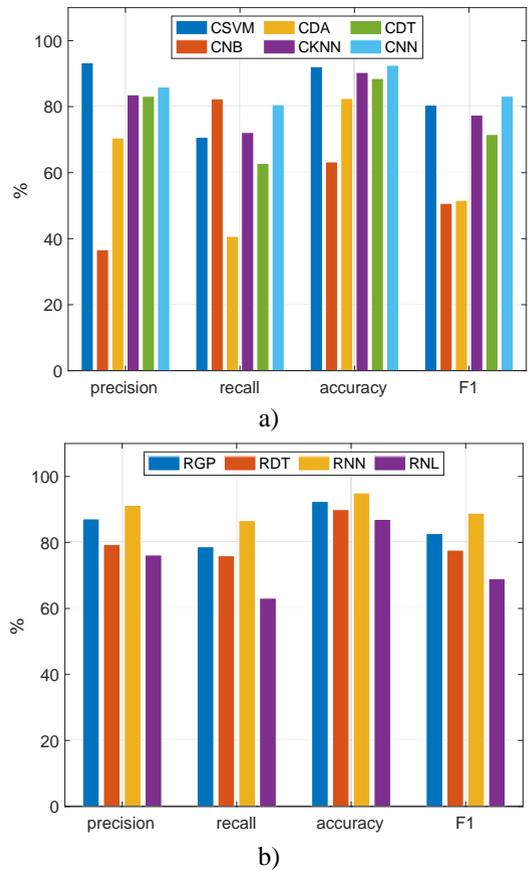


FIGURE 3. Training results for multimedia traffic classification obtained from the different classification a), and regression b) models according to the confusion matrix.

In [1], CSVM y CKNN were the best methods with an 83% and a 73% of system accuracy respectively. In this work, these values have improved to a 93% and 83% respectively, as it can be seen in Fig. 3 a). It is also remarkable the result obtained from RNN with a 91% of accuracy (Fig. 3 b)). In both, [1] and this work, the recall score presents the lowest results of all the measurements, except the 97% obtained in the statistic method

in [1] and the 82% from the CNB (Fig. 3 a)). Nevertheless, in [1] the statistic model did not work properly because it tended to extract a single value that matched a low MOS and matched the test samples. Therefore, that result was excluded. In this study, the same error happened. CNB tends to classify the traffic as critical so that the 22% of the cases in which the traffic is actually critical it would classify it that way the 51% of the times. Thereby, and due to its low accuracy, the value of CNB is excluded. Then, the best value obtained is 86% from RNN. CNN also presents a good result of 80.6%. This measure is quite important for the system. Recall depends on FN, which means, the times the model classifies the traffic as no critical when actually it is. Therefore, the number of learning samples should be increased to improve the test results in that way. This feature was already considered as future work in [1]. The fact that the result presented by CSVM, a 70%, is even lower than CKNN, with a 72%, is an important result. It can be expected, then, that CSVM will not be as efficient as RNN for the system. As regards accuracy, the best result obtained in [1] was 81% from CKNN. Although the result obtained from CKNN in this study is high (90%, the same result as RGP), it is lower than the ones presented by CSVM (92%) and RNN (95%). For the other models, the results are higher than 80% except for CNB (63%). The last analyzed score is F1, which indicates how good the models and the observations matches. The best result obtained is an 88% from RNN, while the best one in [1] was the kernel based function.

After the quantitative analysis of the different models, CSVM and RNN are the learning algorithms that presents the best results and they are the ones analyzed in the test. The result of the test is shown in Fig. 4, which shows that RNN presents a 5.6% of mistakes. A 41% of these mistakes have been made when the model has predicted the traffic as critical when it is not and 59% the other way round. Concretely, the model predicts that the traffic is critical and it is not in 7 samples (76, 122, 137, 138, 139, 140 and 192) and ten times the other way round (44, 67, 162, 193, 196, 217, 273, 288, 289 and 298). The classification errors increased for the CSVM model up to 11.3%. Unlike the other model, the majority of the errors in CSVM are produced when the algorithm classifies the traffic as no critical when it actually is. This happens in the 91.17% of times. This kind of error is more problematic due to the fact that the system would not handle several moments where the traffic transmitted is critical. Some of the samples of this case are: 17, 19, 22, 23, 97 and 225. On the other hand, the model has predicted only in 3 samples that the traffic is critical when it is not (137, 216 and 287).

In conclusion, after the evaluation of all the results presented, and thought CNN and CSVM classify with similar results, the model selected to classify multimedia traffic is RNN. The principal factor is that, although RNN has less accuracy than CSVM, it suits better the multimedia traffic classification problem. In addition, the RNN performance based on Root Mean Square Error (RMSE), which shows how

reliable the model is or how it suits the data, is the highest one with 22.88%. Thereby, the overlearning can be avoided.

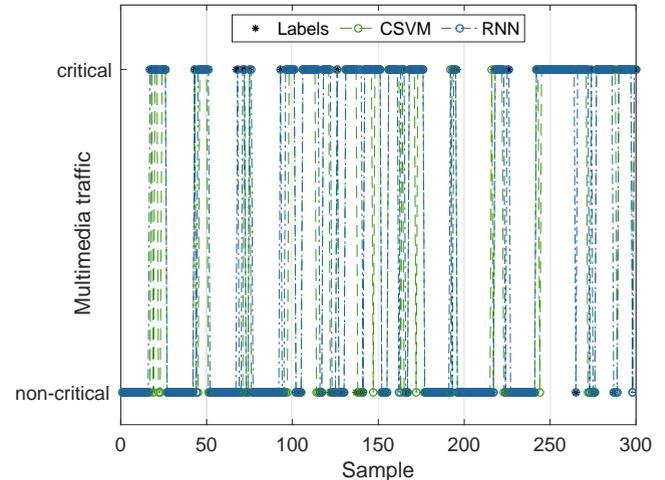


FIGURE 4. Test results of the multimedia traffic classification of CSVM and RNN models.

D. NETWORK ARCHITECTURE OF THE TYPE OF TRAFFIC CLASSIFICATION MODEL

After the study, analysis and evaluation of the different automatic and supervised learning models proposed for type of multimedia traffic classification, RNN has been chosen as the most adequate. The optimization process of this neural evolutionary network modeled by a Bayesian regulation for critical multimedia traffic prediction is described as follows:

1. Gathering network data and video characteristics within different network environments and transmission characteristics to build the dataset.
2. Optimizing the NN architecture to find the most adequate number of the neurons in the hidden layer.
3. Peer data training application to optimize the NN architecture by using the Levenberg-Marquardt algorithm with the Bayesian regulation method until the network converges.
4. Introducing the test data into the well-trained and generalized NN to validate the predictive performance.

The design of the network architecture result of these steps is presented in Fig. 5.

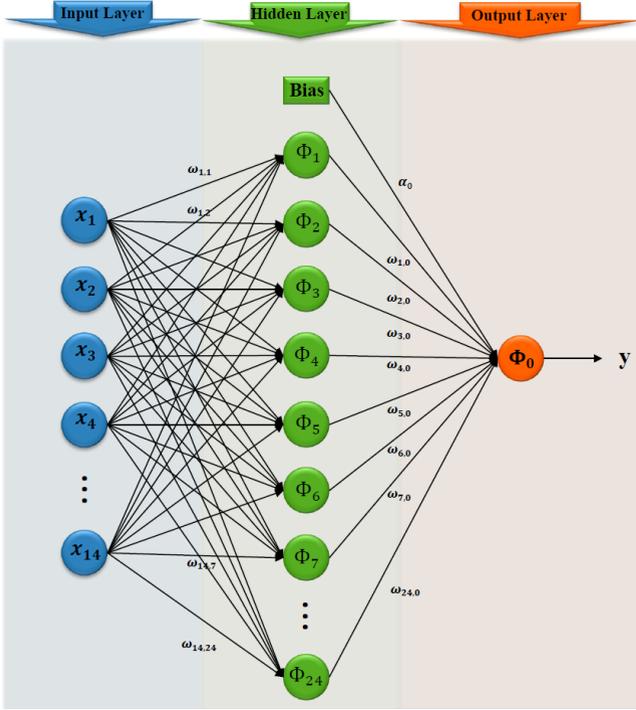


FIGURE 5. General architecture of the neuronal network used in this work.

Where J is the number of neurons in the input layer that corresponds to the set of features composed by network parameters and video characteristics. H is the number of neurons in the hidden layer and M the number of those in the output layer. This network is defined as:

Given n par of arrays

$$(\bar{x}_l, \mathbf{y}_l) = (x_{l1}, x_{l1}, \dots, x_{li}; \mathbf{y}_l), \quad l = 1, 2, \dots, n \quad (1)$$

Where n is the number of samples and i the number of features of each sample. \mathbf{y}_l uses discrete values in the interval (1:1.5). In order to calculate the values of \mathbf{y}_l the following model has been used:

$$\hat{y}_l = \Phi_0 \left(\alpha_0 + \sum_{h=1}^H \omega_{h0} \Phi_h \left(\alpha_h + \sum_{j=1}^J \omega_{jh} x_{lj} \right) \right) \quad (2)$$

Where Φ_h is the activation function of the hidden layer, ω_{jh} is the weight from the input layer to the hidden layer for the neurons between j and h , α_h is the bias value in the hidden layer. ω_{h0} and α_0 are the weights and bias from the hidden layer to the output layer and Φ_0 is the transfer function from the hidden layer to the output layer.

The Bayesian regulation minimizes a lineal combination of squared errors and weights. Moreover, it modifies the lineal combination so that the network has good generalization qualities at the end of the training [18]. This regulation has been performed with the Levenberg-Marquardt algorithm. The different weights are calculated and adjusted within the learning process to minimize the error function E_D as follows:

$$E_D = \sum_{j=1}^J (\mathbf{y}_j - \mathbf{t}_j)^2 = \sum_{j=1}^J \mathbf{e}_j^2 \quad (3)$$

Where J is the number of inputs of the data set D , and \mathbf{t} is the label value.

In order to optimize E_D , the optimum values of the weights and biases have to been determined. One of the algorithms that solve this problem is Levenberg-Marquardt. This algorithm is faster than the backpropagation algorithm. It has been designed to have a training speed close to second order algorithms. However, it has not to calculate the Hesse matrix directly, but it operates as it is explained next [19].

Hessian matrix (H) can be calculated approximately as:

$$H = J^T J \quad (4)$$

The gradient \mathbf{g} could be calculated as:

$$\mathbf{g} = J^T \mathbf{e} \quad (5)$$

Where J is the Jacobian matrix that contains the first derivations of the network errors with regard to the weights and biases and \mathbf{e} is an array of network errors. With backpropagation the Jacobian J^T is calculated and every variable is adjusted according to the Levenberg-Marquardt algorithm [20]. The Levenberg-Marquardt algorithm uses the Hessian matrix approximation with the following Newton algorithm variation:

$$\omega_{j+1} = \omega_j - (J_j^T J_j + \mu_j I)^{-1} J_j^T \mathbf{e}_j \quad (6)$$

Where \mathbf{e} is the error array and I is the identity matrix. The adaptive value of μ is being incremented until the performance of the network starts reducing.

The training stops when one of this conditions is true:

- The maximum number of repetitions is reached.
- The maximum time limit is exceeded.
- The performance is reduced up to the desired value.
- The gradient of the performance becomes lower than the minimum gradient.
- μ surpasses the maximum μ .

The optimization of the network architecture takes often a long time. The Levenberg-Marquardt algorithm is faster than any variation of the backpropagation algorithm [21, 22]. Levenberg-Marquardt is used to minimize the addition of the squared error and to overtake some of the limitations of the standard backpropagation algorithm like the overfitting problem [23]. Another different problem may happen if too few nodes are added to the hidden layer. If this happens the network will have problems in the learning process and the predicted results will be poor. In order to solve this problem, the BRNN algorithm incorporates the Bayes theorem in the regulation scheme. BRNN is, basically, a backpropagation network which combines the conventional addition of the minimal square error function with an additional term called "regulation". Therefore, from the equation (3) the next equation is obtained:

$$S(\omega) = \beta E_D + \alpha E_W; E_W = \sum_{j=1}^m \omega_j^2 \quad (7)$$

Where β y α are the regulation parameters and E_W is the penalty term, which penalizes big values for the weights, being m the number of weights of the system. In this context, the weights of the network are considered to be random variables. Then statistical techniques can be applied to calculate distribution parameters such as, for instance, variations. According to the Bayes' law, the distribution of probability can be written as [24]:

$$P(\omega|\alpha, \beta, D) = \frac{P(D|\omega, \beta)P(\omega|\alpha)}{P(D|\alpha, \beta)} \quad (8)$$

Where $P(\omega|\alpha)$ is the prior probability of the weights, which represents how reliable the weights are before gathering the data; $P(D|\omega, \beta)$ is the likelihood function, which represents the error probability; and $P(D|\alpha, \beta)$ is the normalization factor, named evidence in this model [25].

The optimal weights for the model can be obtained in the training phase by maximizing the posterior probability. It has the same results as minimizing the regulated objective function (7) [26]. If the weight and the distribution of the probability of the data are Gaussian, the prior probability of the weights can be written as follows [27]:

$$P(\omega|\alpha) = \frac{1}{z_W(\alpha)} \exp(-\alpha E_W) \quad (9)$$

Likewise, the error probability can be expressed as:

$$P(D|\omega, \beta) = \frac{1}{z_D(\beta)} \exp(-\beta E_D) \quad (10)$$

Finally, the distribution of posterior probability can be calculated as:

$$P(\omega|\alpha, \beta, D) = \frac{1}{z_S(\alpha, \beta)} \exp(-S(\omega)) \quad (11)$$

By using the Bayes' law, the optimal values of the α and β regulation parameters can be inferred from the data.

$$P(\alpha, \beta|D) = \frac{P(D|\alpha, \beta)P(\alpha, \beta)}{P(D)} \quad (12)$$

Where $P(\alpha, \beta)$ is the prior probability for the α and β regulation parameters, $P(D|\alpha, \beta)$ is the likelihood term, called the evidence for α and β [25]. Thereby, the minimization of $S(\omega)$ has the same results as the maximization of $P(\omega|\alpha, \beta, D)$, which depends on the α and β parameters. The optimal values of α and β are obtained from the next equation [27]:

$$\alpha = \frac{\gamma}{2E_W} \quad (13)$$

$$\beta = \frac{(n-\gamma)}{2E_D} \quad (14)$$

γ is obtained from the next equation:

$$\gamma = \sum_{j=1}^m m - \alpha \cdot \text{trace}^{-1}(H) \quad (15)$$

Where γ is the number of effective parameters, i.e. how some of the NN parameters are effectively used to reduce the error function, m is the number of parameters and H is the Hessian matrix of the objective function $S(\omega)$.

In the Bayesian context, the iterative process to find the optimization of the weights ($S(\omega)$) and the optimal values of α and β in (15) (γ) has to be, according to [26]:

1. Initialize the weights and the values of α and β .
2. Move a step forward in the LM algorithm to find the weights that minimize the objective function $S(\omega)$ (7).
3. Calculate the effective number of parameters γ and new values for α and β .
4. Repeat steps number 2 and 3 until it converges.

With this parameters, the number of neurons in the hidden layer and the number of hidden layers, the model has been adjusted and the multimedia traffic classification results have been improved, as Table I shows. All the scores have been improved by adjusting the model, except for accuracy. The recall of the model has been improved up to near a 4%. This is an important fact for the problem of this work.

TABLE I
OPTIMIZATION OF THE RNN CLASSIFICATION MODEL

	PRECISION	RECALL	ACCURACY	F1	PROXIMITY
RNN	91,289	86,645	95,038	88,907	97,847
RNN $_{\alpha, \beta, H}$	94.032	90.067	91.511	90.465	97.117

In conclusion, the BRNNs are more robust than the standard networks with backpropagation, and the can reduce or even eliminate the necessity of an extended cross-validation. The Bayesian regulation is a mathematic process that converts a non-linear regression into a statistical problem. The advantage of BRNN networks is that they provide solution to a series of problems such as: the model choice, the robustness of the model, the choice of the validation set, the size of the validation effort and the optimization of the network architecture. Moreover, it is hard to overtraining them, because the evidence procedures presents an objective Bayesian criteria to stop the training. It is hard to overfitting them too, because RNN calculates and trains in with a set of effective parameters or weights in the network, disabling those that are not relevant.

IV. CLASSIFICATION AND MODELLING OF NETWORK AND VIDEO FOR QoE REGULATION

Once the type of traffic classification model, which indicates when the traffic is critical, has been defined, some subpatterns that represents different network conditions and video characteristics are needed. In order to do that, the samples are classified according to the MOS, which is defined by the class label. Then, each subgroup is divided into clusters that are represented by a centroid or subpattern.

A. EXTRACTION OF THE NETWORK AND VIDEO SUBPATTERNS

The multimedia traffic subpatterns help to recognize network conditions and possible solutions for critical traffic situations by the variation of the characteristics of the video sent. The first step to extract these multimedia traffic subpatterns was to study the correlation between the video characteristics during the transmission and the video quality perceived for the end user. This is analyzed from the results presented in Fig. 6. It shows several network and video characteristics and their relationship with MOS. They are: frame rate, minimum frame rate, width, height, bitrate and stream size and bandwidth. Their values have been obtained from the dataset used in this work. Each value corresponds to the mean and the standard deviation of every characteristic with the same MOS value defined by the class label. As Fig. 6 shows, there is some connection between the characteristics of the video sent and the MOS. In Fig. 6 a) this relation can be observed with regard to the minimum frame rate. It can be observed that, with the current dataset, the bigger the MOS, the bigger the minimum frame rate. However, this trend does not occur with the frame

rate, which increases with MOS until a MOS = 3 but then there are no more considerable variations. This is also presented in [1] where MOS has a high correlation with the minimum frame rate (0.4732), unlike with the frame rate (0.1528). The correlation value of Pearson exists in the interval $[-1, 1]$. A value close to 0 indicates a low correlation but a value close to 1 or -1 indicates a high correlation. Fig. 6 b) depicts a similar relation but declining, the biggest the MOS, the lowest the video resolution. This trend is clear and, as it is shown in [1], it coincides with a high correlation between those values (Height = -0.2902 and Width = -0.3108). As regards bitrate and stream size, this tendency is not as clear, although it is considerable (Fig. 6 c)). This fact demonstrates the correlation between both measurements and it will help to the development of the management system. The last measurement, the bandwidth, Fig. 6 d), represents a network parameter with a really low correlation of 0.1552. Therefore, this parameter will not be considered by the system as a decisive measurement for improving the critical status of the network.

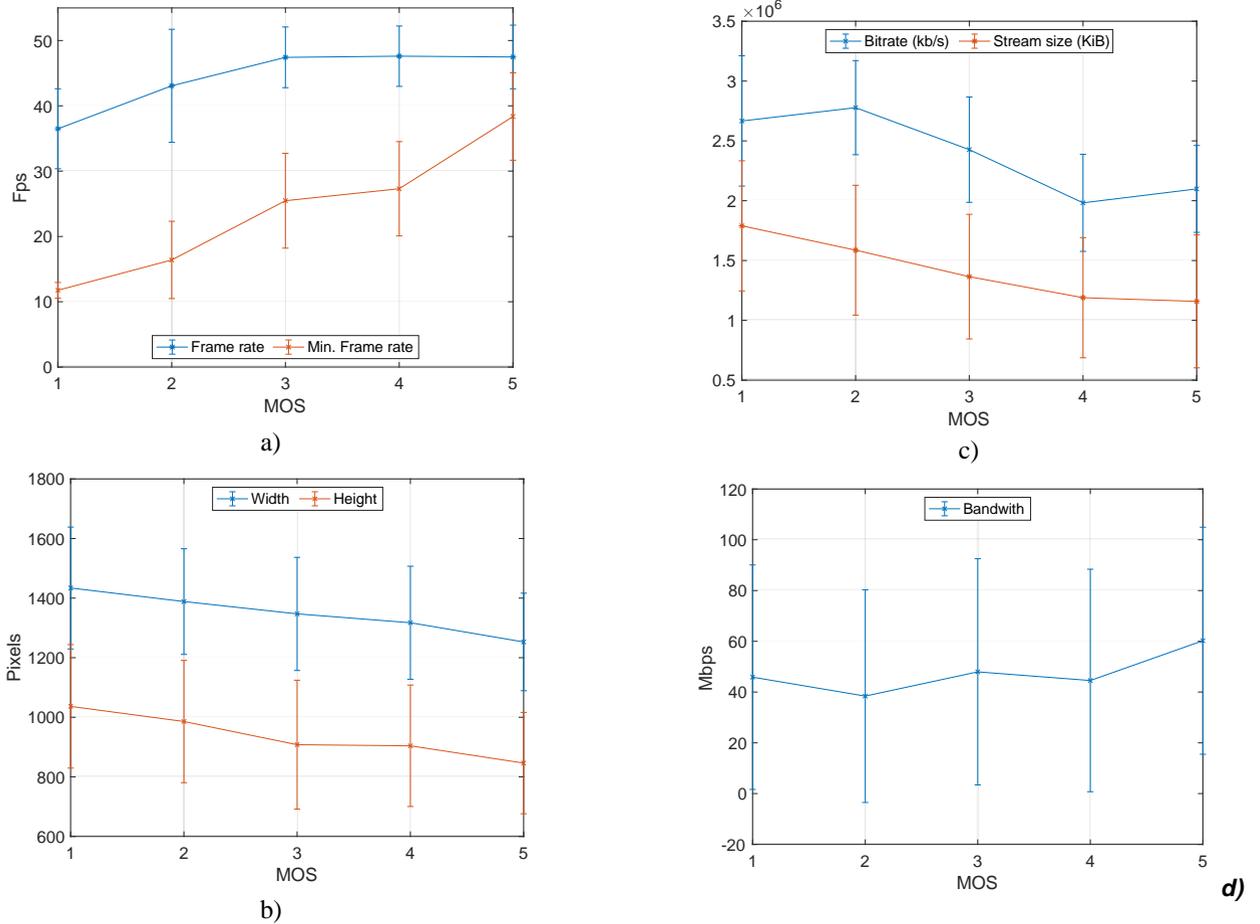


FIGURE 6. Mean and standard deviation of several characteristics with regard to MOS, like a) frame rate and minimum frame rate, b) width and height, c) bitrate and stream size and d) bandwidth.

After discussing the existent correlation between video and network characteristics with regard MOS, a system for extracting subpatterns will be developed. The main idea is to create clusters which can make relations between both, different network conditions and video characteristics, and the quality perceived by the end user. Therefore, for each MOS value, a set of subpatterns that represent that value will be obtained. The number of subpatterns or clusters for each MOS value is unknown. In order to extract them, validation techniques will be used to select the adequate number.

The algorithm used to carry out the clustering has been K-means. The metric distance used for the calculation of the centroids is the Euclidean distance. The whole dataset X has been divided into subset $X_M \in X$, where $M = \{1, 2, 3, 4, 5\}$. This defines subset with regard the MOS values. Then, given an array of measurements $x_m = \{x_{m_1}, x_{m_2}, \dots, x_{m_n}\} \in X_M$, the k sets are calculated using the K-means algorithm, minimizing the sum of squares for each set of centroids $C_M = \{C_{M1}, C_{M2}, \dots, C_{Mk}\}$, so that:

$$\underset{C_M}{\operatorname{argmin}} \sum_{i=1}^k \sum_{x_m \in C_{Mi}} \|x_j - \mu_i\|^2 \quad (16)$$

Where μ_i is the mean of points in C_{Mi} and $n = 14$, being n the number of characteristics per sample. The result is C_{Mk} centroids for X_M data sets.

Once the algorithm used for the clustering has been defined, the optimal number of clusters will be calculated. The validation method used to define the number of clusters is Silhouette. The statistic of Silhouette works with distances. This method traces the silhouettes using the function of distance between points specified in the distance metric, which in this case is the Euclidian distance. In Fig. 7, the results obtained for the subsets defined for MOS = 4 and MOS = 5 after applying the Silhouette method are depicted. The optimal number of clusters is 4 for the subset of data defined by MOS = 4 and 5 for the one defined by MOS = 5. To begin with, negative values represent errors in the selection of a cluster. However, there are no negative values in Fig. 7. That means there are no values assigned to a cluster where they actually do not belong. Another relevant fact, which determines the validation of the clustering, is that the values obtained must be higher than 0.8. In both graphs, the majority of values are higher than 0.8. That means that most of the area under the value of maximum probability equal to 1 is covered. Once defined the number of clusters and once the centroids have been extracted, the working of the system is defined in the next subsection.

B. SUBPATTERN-BASED QoE REGULATION MODEL

Once the subpatterns are defined, the system has to be able to, given a set of network parameters, obtain video characteristics, or vice versa, to solve the situations where the traffic is critical. In order to achieve this, the next process is performed.

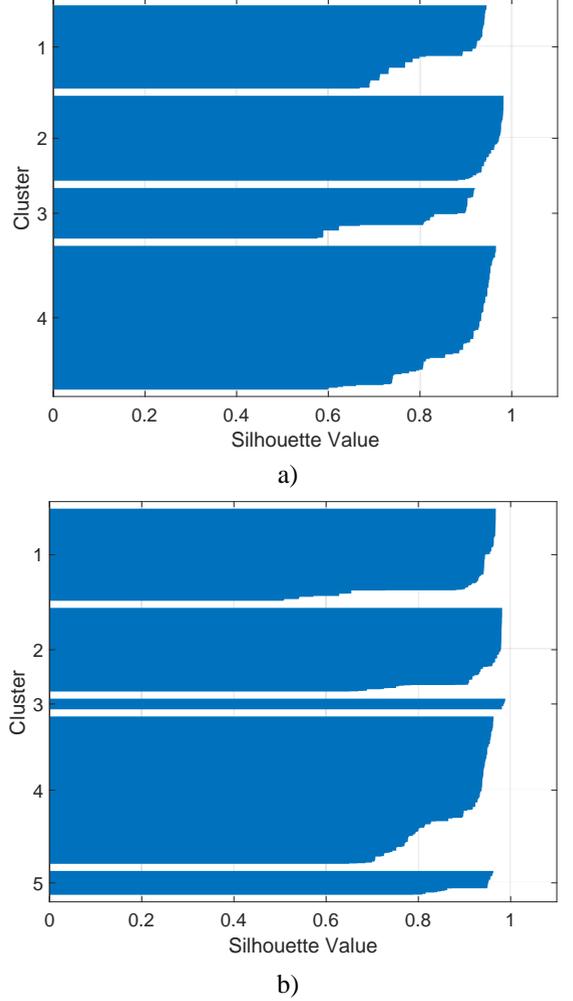


FIGURE 7. Subpatterns for a) MOS = 4 and b) MOS = 5.

Firstly, the minimum Euclidean distance between the network parameters of the input and every subpatterns or centroids is calculated. The network Euclidean distance between the input sample $x = \{x_1, x_2, \dots, x_n\}$ and the centroid $c = \{c_1, c_2, \dots, c_n\}$ is:

$$\begin{aligned} d((x_1, x_2, \dots, x_r), (c_1, c_2, \dots, c_r)) &= d((c_1, c_2, \dots, c_r), (x_1, x_2, \dots, x_r)) \\ &= \sqrt{(c_1 - x_1)^2 + (c_2 - x_2)^2 + \dots + (c_r - x_r)^2} \\ &= \sqrt{\sum_{i=1}^r (c_i - x_i)^2} \end{aligned} \quad (17)$$

Where r is the number of network parameters for each sample and whose value is 5. The set of centroids that define a non-critical traffic, i.e. with MOS = 5 and MOS = 4, is defined as:

$$C = \{(c_{t1}, c_{t2}, \dots, c_{tn}) \mid C = C_4 \cup C_5 \wedge t = K_4 + K_5, \text{ where } K_M \text{ is the total number of centroids of } C_M\}.$$

Therefore, the minimum network distance to each subpattern is defined as:

$$\underset{c_{j,i}}{\operatorname{argmin}} \left(\sqrt{\sum_{i=1}^r (c_{j,i} - x_i)^2} \right), j \in [1, t] \quad (18)$$

Equation (18) returns the subpattern or centroid \mathbf{j} whose network conditions are similar to the input sample. From this result, the video characteristics of that subpattern are extracted. Then, the video characteristics will change until the critical network situation will be solved. The Pearson's lineal correlation coefficient between the objective QoE and each characteristic determines the probability that the system chooses one characteristic or other. The higher the probability of a characteristic is, the higher priority of changing that characteristic has. The probability function of $c_{j,i}$ is defined as:

$$f(c_{j,i}) = f(x_i) = \frac{\sum_{s=1}^w (x_{s,i} - \bar{x})(y_s - \bar{y})}{\sqrt{\sum_{s=1}^w (x_{s,i} - \bar{x})^2} \sqrt{\sum_{s=1}^w (y_s - \bar{y})^2}} \quad (19)$$

Where W is the total number of samples of the dataset \mathbf{X} , \mathbf{Y} is the set of class label for each sample and i the characteristic index.

The probability of selecting the characteristic i of the centroid \mathbf{j} from the centroids set \mathbf{C} is defined by $P(c_{j,i})$. The order of the index depends on the maximum probability value that maximizes the value of the function that is defined as:

$$i = \underset{i \in I}{\operatorname{argmax}} f(x_i, i) \quad (20)$$

Where $I = i_5, i_6, \dots, i_{14}$ is the set of indices to be estimated and which correspond to the video characteristic indices. If the estimated index did not solve the network critical state problem, the next index with the maximum probability would be searched.

V. ROBUST MULTIMEDIA TRAFFIC MANAGEMENT SYSTEM

After configuring the multimedia traffic classification model and the QoE regulation model, a robust multimedia traffic management has been developed. The system analyzes and classifies the multimedia traffic, detects if it is critical and, if so, finds the video parameter that, by being changed, can solve the problem. The system is considered as robust because it learns from the cases that are not solved by applying the method discussed in section IV B, creating new subpatterns. If the problem is not solve, the network conditions and the video characteristics are saved to feedback the system. In this section, the architecture of the network is presented and the interaction between the modules defined is discussed. Then, the algorithm and its implementation are detailed. Finally, the test bench is discussed.

A. NETWORK ARCHITECTURE

The classification model, as a system, can be applied in every kind of network because the performance of the system does not depend on the kind of network, but only on the network parameters (jitter, delay, packet loss and bandwidth) monitored on the nodes." However, in this paper, we use SDN as a tool to provide the system with data. Consequently, we extract statistics directly from the network nodes. Despite the independency of the system towards the network architecture, our proposed network presents the architecture displayed on Fig. 8, as an example. The blue lines are the corresponding ones to the datapath, which is used to communicate the data through the network nodes. However, the connections painted in green belong to the SDN Control to Data-Plane Interface (CDPI), which physically implements the interaction between network nodes and the SDN controller. The final nodes are IoT nodes, which are the producers of video traffic. However, the core of the network, starting from the gateway, are nodes of a SDN network. Therefore, there is a SDN controller which manages the network. SDN controller receives the statistics gathered by the network nodes. These statistics are network parameters: delay, jitter, loss rate and bandwidth. From these parameters, the management system determines if a great QoE degradation can happen. OpenFlow protocol is used to implement the communication between the nodes and the SDN controller. Furthermore, these statistics are needed by the system to perform several tasks. For instance, classifying the critical traffic is one of the task that needs the data from the network in real time. Since this interaction between the SDN controller and the system must be implemented through the northbound API, we define the global logical architecture of the network as is depicted in Fig. 9. It depicts how the communication between modules is implemented. Firstly, the communication between the SDN controller and the network nodes is performed through the southbound API. The northbound API is the channel between the SDN controller and the AI module. In this interaction, not only the network statistics and data are sent to the AI module. The AI module sends back the necessary actions to be performed in the network to ensure the QoS. The video streaming application receives orders from the SDN controller. This allows the SDN controller to execute the orders of the AI module. Depending on the results of the system, the SDN controller will change the video characteristics. This is performed by using the interface of the video streaming application. The software features are presented in the test bench subsection.

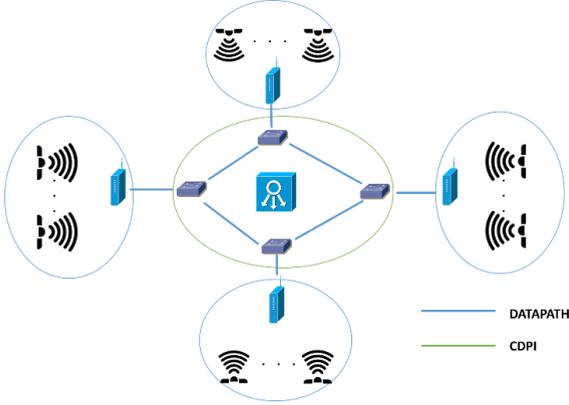


FIGURE 8. Network architecture example

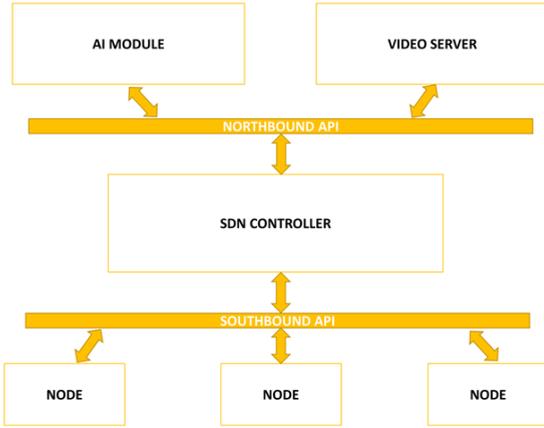


FIGURE 9. SDN Architecture

B. SIMPLE MANAGEMENT ALGORITHM

Fig. 10 shows the algorithm that describes in detail how the management system works. Every 2s, which corresponds to a GOP, the system gets as input the QoS and video parameters. According to those parameters, the classifying model estimates if the traffic is critical or if it is not. If it is not critical, the previous process will be repeated. However, if the traffic is critical, the process to solve this critical situation will start. Every traffic subpatterns that can recover a MOS higher than 3 to exit from the critical situation are analyzed. Those subpatterns are obtained from the clustering, explained in section IV, applied to the multimedia traffic patterns.

When the subpatterns have been obtained, the system calculates which parameters of the transmitted video can be modified to improve the QoE. The system uses as input parameters the QoS parameters and the different traffic subpatterns extracted from the clustering. Then, the shortest distance to every subpattern according to the network parameters is calculated. Thereby, the system looks for the subpattern, or centroid, that represents the closest or most similar network conditions to the current ones. Once the subpattern has been obtained, the video characteristics of that subpattern are used as the new video transmission parameters. The selection of the video characteristic is performed

regarding to the model explained in section IV. Specifically we will use (20). If the critical traffic situation is not solved, the next characteristic, according to the model, is chosen. If applying this process the traffic critical situation is not solved, then we would apply the robust method of management. In this way, instead of changing only one different characteristic, each of them will be added or combined to the transmitted video. If this procedure does not work, we would keep this sample in order to improve the system with new intelligent methods that we will study in future works. This process is repeated until the problem is solved or there are no more characteristics to choose. As the subpatterns are defined by both, network parameters and video characteristics, any of them could be the input parameter or the parameter to be adjusted. This feature of the system will be discussed further in future works.

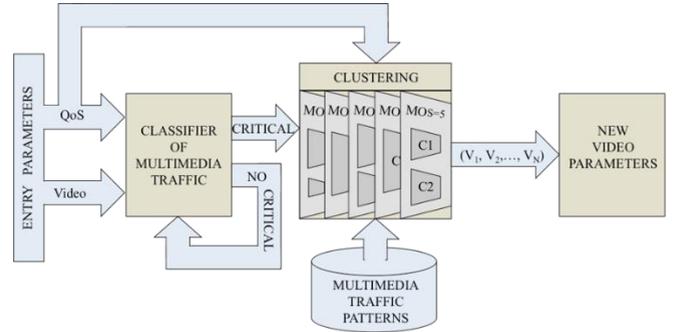


FIGURE 10. Management system algorithm.

C. ROBUST MANAGEMENT SYSTEM IMPLEMENTATION

In this subsection, the implementation of the robust management system is described. Algorithm 1 details the process. For each input cycle (GOP of 2s), the program reads the average network and video parameters and an array of indices is initialized. A loop starts until the array is not empty. In this loop, the system checks the multimedia traffic using the BRNN, trained and validated in the learning process. If the traffic is not critical, the array is cleared and the program restarts. Nevertheless, if the traffic is critical, firstly, the index of the subpattern with the lowest Euclidean distance, by using equation (18), between the network parameters of the subpatterns matrix (SUBPATTERNS) and the network parameters of the input sample (X). Then, the new video characteristics (NEWVIDEO_{param}) are obtained from the subpatterns matrix of that index. After that, the index of the characteristic with the highest correlation from NEWVIDEO_{param} is calculated, by using equation (20), and, then, it is removed from the array of indices. Finally, this characteristic is changed in the video that is being transmitted. If, after changing the characteristic, the traffic becomes non-critical, the program will leave the while loop, clearing the array of indices I and it will restart. If the critical situation is not solved, the next characteristic with the highest correlation

will be selected. This is repeated until the program finds a characteristic that solves the critical situation or until the array of indices gets empty. If the problem is not solved, a robust solution is taken. This solution consists of combining different characteristics attending to equation (20) until a solution is found. This process is performed if there has not been any other similar case performed with this robust method. The robust solution is determined by the Euclidean distance, using equation (18). If there is no other similar case and the system finds a solution, it calculates the distance to the centroids of the clusters and a new subpattern, along with its solution, it is saved (ROBSUBPATTERNS). If the system do not find a solution, the case is saved as a sample to study to improve the system (UNKNOWN). The intelligent system to improve it with this feedback is not studied in this work, but can be discussed in future works. In this last scenario, the program saves the sample so that the network can learn from it (UNKNOWN).

Algorithm 1

```

1:  X1 = QoSparam
2:  X2 = VIDEOparam
3:  I = ALL_INDEX
4:  indexsubpat = min_dist(SUBPATTERNS(4:5), X1)
5:  NEWVIDEOparam = SUBPATTERNS(indexsubpa, 6:14)
6:  while size(I) not NULL && SOLVED = false
7:    TRAFFIC = net(X1, X2)
8:    if TRAFFIC = CRITIC
9:      if NOT ROBUSTMETHOD
10:     indexcarac = maxcorr(NEWVIDEOparam, I)
11:     X2(indexcarac) = NEWVIDEOparam(indexcarac)
12:   else
13:     ROB_INDEXcarac = robust_maxcorr(NEWVIDEOparam, I)
14:     X2(ROB_INDEXcarac) = NEWVIDEOparam(ROB_INDEXcarac)
15:   end if
16:   I = delatecarac(I, indexcarac)
17:   if size(I) is NULL
18:     TRAFFIC = net(X1, X2)
19:     if TRAFFIC not CRITIC
20:       Go to 6
21:     end if
22:   end if
23: else
24:   SOLVED = true
25:   if ROBMETHOD
26:     ROBSUBPATTERNS(f++) = NEW ROBSUBPATTERN
27:   end if
28: end while
29: if SOLVED = false
30:   if NOT ROBMETHOD
31:     indexsubpat_robust = min_dist(ROBSUBPATTERNS(:,1:5), X1)
32:     if indexsubpat_robust = NULL

```

```

33:     ROBMETHOD = true; I = ALL_INDEX;
34:     X2 = VIDEOparam
35:     Go to 6
36:   else
37:     NEWVIDEOparam = ROBSUBPATTERNS(indexsubpa, 6:14)
38:     X2 = NEWVIDEOparam
39:   end if
40:   else if SOLVED = false && ROBMETHOD
41:     UNKNOWN(fi++) = [X1 X2]
42:   end if

```

D. TEST BENCH

In this subsection, both the hardware and the software used are specified. In Fig. 11, the hardware of the SDN nodes is shown.

Regarding the software, the SDN controller has been created from scratch (programming by ourselves, not using one of the available SDN controllers like OpenDaylight) so that we could perform customized researches such as [28]. This allows a better control over the operation and guarantees the inter-operability between the software modules described by the network architecture. As video server software, VLC has been chosen due to its capability to modify the video streaming dynamically. Therefore, the SDN controller can modify the video characteristics. The version used is VLC 3.0.7.1 for both, the video server and the client computers.

Node	Hardware
SDN Controller Host	Intel Core i7-5500U 2.40GHz 8GB RAM
SDN Nodes	Aruba 2930F 24G 4SFP Switch
Client Computers	Intel Core i5 processor 8 GB RAM
Video Server	Intel Core i7 processor 16 GB RAM

FIGURE 11. Hardware used in the test bench.

In the next subsection, the results of the test are discussed.

E. WORKING TEST

The last study performed in this work is a working test of the developed system. The test samples are provided to the system and the system check for each sample if the traffic is critical or it is not. If the traffic is critical, the system tries to solve this situation by changing the video parameters as it has been described in this section. Two different experiments have been performed, in the first one all the centroids or subpatterns are used and in the other one some of them were removed to analyze the consequences. In order to improve the robustness of the system, the network and video parameters are saved to feedback the system, only if the problem has not been solved.

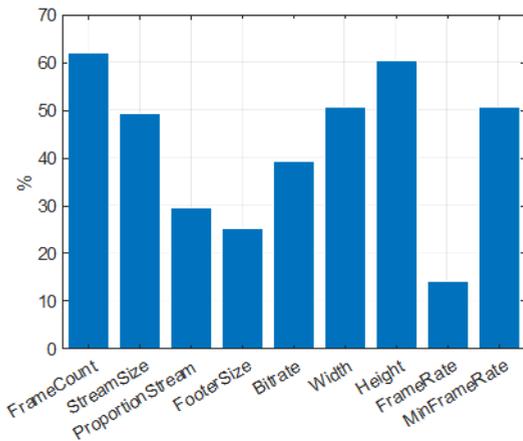
1) FIRST TEST

In the first experiment, the performance of the system when all the multimedia subpatterns are used is evaluated. If the system detects a multimedia traffic critical situation, the network status at that moment will be the input parameters. Then, with those parameters, the system will search for the

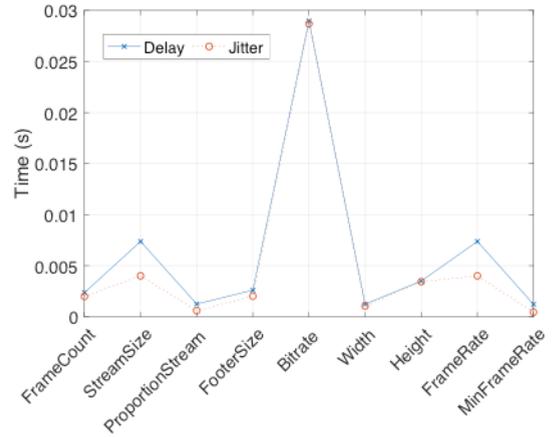
subpattern that matches the conditions given. The Euclidean distance is the method which is used to make this comparison. The lowest distance determines the subpattern to be selected. Once the subpattern has been selected, the video characteristics will be modified to solve the critical situation of the network. In this first experiment, every video characteristics were checked to observe if the critical situation changed. Fig. 12 a) shows the results of the test. If the system is allowed to select any centroid, it has a 100% success rate. As it is shown in the figure, the characteristics that more critical traffic situations solve are: Frame count and height, with about 60%. Then, with about a 50% stream size, width and min. frame rate. The characteristic with the worst results to solve critical traffic situations is the frame rate, with only a 15% of all the cases.

A statistical study has been performed from the obtained results. The data presented in Fig. 12 shows that the system uses bitrate as characteristic to be changed when the delay is high (over 0.03s), frame rate and stream size with delays values over 0.0073s and frame count, proportion stream, footer size, width and minimum frame rate when the delay is low (0.003-0.0012s). The results as regards jitter are similar. However, the differences between low and very low jitter values are not as big as with regard to delay. These values are 0.004s for a low jitter and 0.002s for a very low jitter.

The loss rate can be divided into three categories: high loss rate (4%), medium loss rate (0.5%-0.2%) and low loss rate (0.08%). In order to solve a critical situation with a high loss rate, the characteristics used are the stream size, the bitrate and the frame rate. These characteristics are related to the video bandwidth consumption. In scenarios with network conditions with loss rate of 0.2%-0.5%, the video characteristics changed are: frame count, footer size and height. Finally, when the loss rate is lower than 0.008% these parameters are: proportion stream, width and min. frame rate.



a)



b)

FIGURE 12. Video characteristic selection performed by the system according to the NN model (based on QoE estimation) to solve the critical situation with all the subpatterns a). In b) the results of the statistical study relationship jitter and delay with every video characteristic selection.

2) SECOND TEST

Like in the last scenario, a statistical study has been performed from the obtained results. Nevertheless, if the system can only select a specific number of centroids and the others are discarded, the problem is not always solved. This happened in two cases. In order to solve those two cases, the robust method of modifying two different characteristics together, depending on the subpattern, was used in this study. Fig. 13 a) shows the test result for every video characteristic checked to observe if the critical situation changed. As the figure shows, and unlike the last case (12 a)), the characteristic that solves more critical traffic situations is the footer size with about 55% of cases. The next characteristic is the minimum frame rate, with about a 50%. Stream size, bitrate and height are able to solve about a 40% of critical traffic situations. The characteristic with the worst results, and like in the last case (12 a)), is the frame rate with less than the 15% of the cases. It can be observed that the frame count in the last case and the footer size in this one, solve an important quantity of cases, despite being characteristics with low correlation. This will be an interesting aspect to be addressed in future works regarding the improvement of the characteristics selection by using the management system. As it is depicted in Fig. 13 b), when there is a high delay, the parameters changed are the bitrate and the frame rate of the video. When there is a low delay, the other parameters are used. The delay is considered high when it is higher than 0.03s and low when it is lower than 0.007s. When the delay is around 0.07s, the parameter changed is the stream size. The frame count, footer size and height are varied when the delay is over 0.002s. Finally, when the delay is 0.0012s, the parameters used are proportion stream, width and minimum frame rate. The behavior regarding jitter is similar, although, for the stream size parameter, the value is decreased to 0.004s from the 0.007s with regard to delay. The loss rates

can be classified into high loss rate (4%), significant loss rate (1%) and low loss rate (0.2%-0.8%). In order to solve a critical situation with loss rates greater than 4%, the parameters changed are bitrate and frame rate. The stream size is used with losses of 1% and the other parameters for losses of 0.08-0.2%.

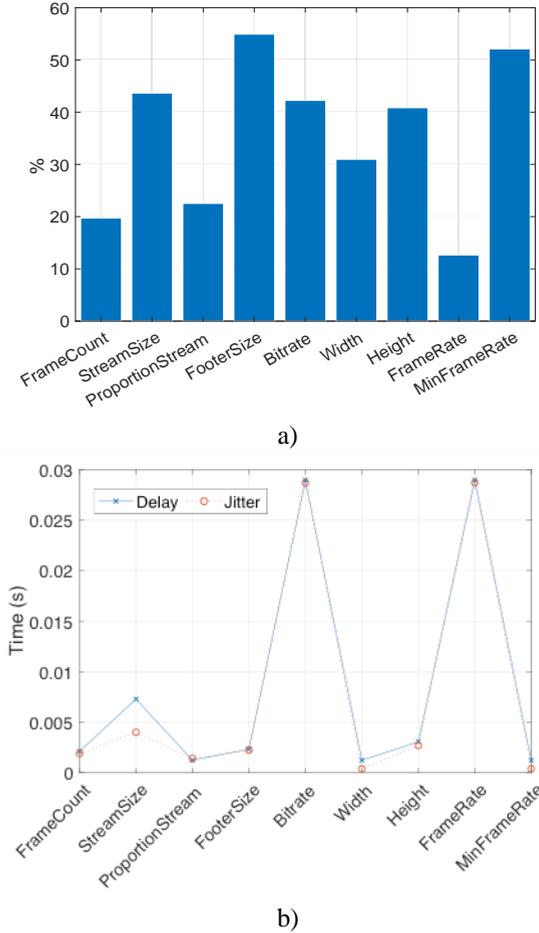


FIGURE 13. Video characteristic selection performed by the system according to the original model (subjective QoE) to solve the critical situation without all the subpatterns a). In b) the results of the statistical study relationship jitter and delay with every video characteristic selection.

VI. CONCLUSION AND FUTURE WORK

In this work we have developed a multimedia traffic management system to classify the type of traffic and, in case of critical situation, this tool is able to reverse the situation by changing the parameters of the video that is being transmitted. For the development of this system, first we have analyzed different classification and regression models for the QoE estimation. The models that better fit the proposed estimation problem are the classification CSVM and the regression RNN. We have performed a test to analyze them in more detail and we have concluded that, although both of them have very similar precision values, RNN adjusts better to the problem. The expected result is very close to the one we get with this

method. In the following study we have analyzed and evaluated the same models from the previous study case to check how well they fit the problem of multimedia traffic classification. Again, CSVM and RNN were the models providing the best results. RGP, CDT and CKNN are also good models. They provided fair results in both studies but not so good as CSVM and RNN. Based on the results obtained, we decided that the model based on RNN fits the problem better. Next step was to define the model and the architecture based on BRNN. We also made some adjustments in order to get better statistical classification results, improving the results in the measure of Recall by 4%. Thus, after a deep study and development of a QoE estimation system, we have been able to improve the results of multimedia traffic classification obtained in the study presented in [1].

The system consists of a traffic classification module and a module to reverse critical situations of the network. Once detected a critical state of the network, this module looks for network conditions that are similar to different conditions classified in subpatterns. When the subpattern is found, it will give us a series of video features that will be applied to the video that is being transmitted, in order to reverse the critical traffic situation. The selection of these characteristics is based on the maximum correlation with respect to the objective QoE.

Once we have selected the multimedia traffic classification model and the QoE regulation model based on subpatterns, we implement the management system. The system works in the following way: first it captures samples every 2s (based on GOPs), then it establishes if it corresponds to a network critical situation. If critical situation occurs, then the system checks which subpattern has more similar network conditions. This subpattern will give us the characteristics that we should apply to the video that is being transmitted until the critical situation of the network reverts, as explained in Section IV. A system implementation, from the network architecture, using the SDN technology, to the algorithm has been proposed in Section V. After the tests we made with the system, we have verified that minimum frame rate, resolution and stream size are highly efficient when the system is reverting a critical situation. We also have verified that with all subpatterns, all critical network situations have been solved. If we eliminate subpatterns, this fact would not occur.

In future works we will improve the proposed system. The selection of the characteristics will be made by machine learning and statistical methods of conditional probability. In this way, we will try to obtain the value of maximum probability given by the conditional probability of a given subpattern, and obtain the probability that a specific characteristic is selected as a solution to the problem. We have been considered the idea of interacting with the end user to improve the robustness of the system, using their feedback. Thereby, the model could learn continuously from new cases and improve the estimation results. We will also consider that, instead of waiting for the first warning of critical network situation to activate the system, we will analyze when this

activation could be more effective. In addition, we will analyze how the system works in different environments such as IoT, smart cities, and 5G and how the system can learn in these different scenarios. Another task for future work is to implement the system in SDN nodes to create a routing algorithm [29] and analyze how it works in a real environment. Finally, we would like to optimize the system, improve the execution time, to be used in real networks, and using resource allocation systems [30]. And we also plan to make the system reversible, that is, since the subpatterns are defined based on network parameters and video characteristics, it will allow us to enter the video features as input, and as output, return those network parameters that could revert critical network situations.

ACKNOWLEDGMENT

This work has been partially supported by the “Ministerio de Educación, Cultura y Deporte”, through the “Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2015)”, grant number FPU15/06837 and by the “Ministerio de Economía y Competitividad” in the “Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento” within the project under Grant TIN2017-84802-C2-1-P.

REFERENCES

- [1] A. Cánovas et al., “Multimedia Data Flow Traffic Classification Using Intelligent Models Based on Traffic Patterns”, *IEEE Network*, vol. 32, no. 6, pp. 100-107, 2018.
- [2] Z. Kotevski and P. Mitrevski, “Performance Assessment of Metrics for Video Quality Estimation,” *Proc. Conf. Info., Commun. Energy Systems and Technologies*, pp. 693–96, June 23–26, 2010, Ohrid, Macedonia.
- [3] H. Zhang et al., “DeepQoE: A Unified Framework for Learning to Predict Video QoE”, *IEEE International Conference on Multimedia and Expo (ICME)*, 23-27 July 2018, San Diego, USA.
- [4] M. Amiri et al., “Game-Aware and SDN-Assisted Bandwidth Allocation for Data Center Networks”, *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 10-12 April 2018, Miami, USA.
- [5] V. Vasilev et al., “Predicting QoE Factors with Machine Learning”, *IEEE International Conference on Communications (ICC)*, 20-24 May 2018, Kansas, USA.
- [6] A. Floris et al., “Managing the Quality of Experience in the Multimedia Internet of Things: A Layered-Based Approach”, *IEEE International Conference on Communication Workshop (ICCW)*, 8–12 June 2015, London, UK.
- [7] M. López-Martínez et al., “Deep Learning Model for Multimedia Quality of Experience Prediction Based on Network Flow Packets”, *IEEE Communications Magazine*, vol. 56, no. 9, pp. 110-117, 2018.
- [8] K. Seshadrinathan et al., “Study of Subjective and Objective Quality Assessment of Video,” *IEEE Trans. Image Processing*, vol. 19, no. 6, pp. 1427–41, 2010.
- [9] X. Huang et al., “Deep Reinforcement Learning for Multimedia Traffic Control in Software Defined Networking”, *IEEE Network*, vol. 32, no. 6, pp. 35-41, 2018.
- [10] T. T. T. Nguyen and G. Armitage, “A Survey of Techniques for Internet Traffic Classification Using Machine Learning,” *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp.56–76, 2008.
- [11] M. Soysal and E. G. Schmidt, “Machine Learning Algorithms for Accurate Flow-Based Network Traffic Classification: Evaluation and Comparison,” *Performance Evaluation*, vol. 67, no. 6, pp. 451–67, 2010.
- [12] A. W. Moore and D. Zuev, “Internet Traffic Classification Using Bayesian Analysis Techniques,” *Proc. 2005 ACM SIGMETRICS Int’l. Conf. Measurement and Modeling of Computer Systems*, Banff, Alberta, Canada, June 6–10, pp. 5–60, 2005.
- [13] A. Tongaonkar et al., “Towards Self-Adaptive Network Traffic Classification,” *Computer Communications*, vol. 56, no. 1, pp. 35–46, 2015.
- [14] M.S. Mushtaq, “Empirical study based on machine learning approach to assess the QoS/QoE correlation”, *17th European Conference on Networks and Optical Communications*, 2012, Spain.
- [15] R. Shalala, “Video QoE Prediction Based on User Profile”, *International Conference on Computing, Networking and Communications (ICNC 2018)*, Maui, USA.
- [16] L. Huixian et al., “A Novel Traffic Classification Algorithm using Machine Learning”, *IEEE International Conference on Broadband Network & Multimedia Technology*, 2009, Beijing, China.
- [17] S. Zaner, “Practical Machine Learning Based Multimedia Traffic Classification for Distributed QoS Management”, *IEEE 36th Conference on Local Computer Networks*, 2011, Bonn, Germany.
- [18] Kayri, Murat, “Predictive Abilities of Bayesian Regularization and Levenberg–Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data”, *Mathematical and Computational Applications*, vol. 21, pp. 1-11, 2016.
- [19] Hirschen K, Schafer M, “Bayesian regularization neural networks for optimizing fluid flow processes”, *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 7, pp. 481–500, 2006.
- [20] Martin HT, Menhaj MB, “Training feed forward networks with the Marquardt algorithm”, *IEEE Trans Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [21] Ozgur Kisi, Erdal Uncuoğlu, “Comparison of three back-propagation training algorithms for two case studies”, *Indian Journal of Engineering and Materials Sciences*, vol. 12, no. 5, 2005.

- [22] S. Mohanty, M. K. Jha, A. Kumar, K. P. Sudheer, "Artificial neural network modeling for groundwater level forecasting in a river island of eastern India", *Water Resources Management*, vol. 24, no. 9, pp. 1845–1865, 2010.
- [23] H. P. Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems", Department of Civil and Environmental Engineering, Duke University, pp. 1–17, 2013.
- [24] Yi Lin, "Support Vector Machines and the Bayes Rule in Classification", *Data Mining and Knowledge Discovery* vol. 6, no. 3, pp. 259-275, 2002.
- [25] Steven N Goodman, "Introduction to Bayesian methods I: measuring the strength of evidence", *Clinical Trials*, vol. 2, no. 4, pp. 282–290, 2005.
- [26] Foresee and Hagan, 1997 F.D. Foresee, "M.T. HaganGauss–Newton approximation to Bayesian learning", *Proceeding of the 1997 International Joint Conference on Neural Networks*, pp. 1930-1935, Houston, 1997.
- [27] Durden F., Winkler D., "Bayesian Regularization of Neural Networks", *Artificial Neural Networks, Methods in Molecular Biology™*, vol. 458, pp. 23-42, 2008.
- [28] A. Rego, A. Canovas, J. M. Jiménez and J. Lloret, "An Intelligent System for Video Surveillance in IoT Environments," *IEEE Access*, vol. 6, pp. 31580-31598, 2018.
- [29] Albert Rego, Sandra Sendra, Jose M. Jimenez, Jaime Lloret, "Dynamic metric OSPF-based routing protocol for Software Defined Networks", *Cluster Computing*, 2018.
- [30] A Cánovas, M Taha, J Lloret, J Tomás, Smart resource allocation for improving QoE in IP Multimedia Subsystems, *Journal of Network and Computer Applications* 104, 107-116. 2018.