



UNIVERSIDAD DE

VALLADOLID

1

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

## TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN

EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

# **Application of machine learning techniques to optical communication systems and networks**

Autor:

**D. Francisco Javier Mata Gómez**

Tutores:

**Dr. Ignacio de Miguel Jiménez**

**Dr. Ramón José Durán Barroso**

Valladolid, September 14<sup>th</sup>, 2017

---

**TÍTULO:**            **Application of machine learning techniques to optical communication systems and networks**

**AUTOR:**           **D. Francisco Javier Mata Gómez**

**TUTORES:**       **Dr. D. Ignacio de Miguel Jiménez**  
                          **Dr. D. Ramón José Durán Barroso**

**DEPARTAMENTO:** **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

---

**TRIBUNAL**

---

**PRESIDENTE:**   **Dr. D.**

**VOCAL:**         **Dr. D.**

**SECRETARIO**    **Dr. D.**

---

**FECHA:**           **15 de Septiembre de 2017**

**CALIFICACIÓN:**

---

**Resumen**

El presente TFM realiza una revisión de la aplicación de técnicas de aprendizaje automático en los sistemas y redes de comunicaciones ópticas. Además, estudia y compara las características de diversos métodos de aprendizaje automático, tales como: máquinas de vectores soporte, regresión logística, árboles de decisión y bosques aleatorios, para predecir la calidad de la transmisión al emplear circuitos ópticos en redes de comunicaciones ópticas con encaminamiento por longitud de onda. Los modelos desarrollados en el TFM obtienen mejores prestaciones que propuestas anteriores, fundamentalmente en términos de tiempo de cálculo, posibilitando su utilización en modo on-line incluso en redes altamente dinámicas, amén de ser más sencillos.

**Palabras clave**

Técnicas de aprendizaje automático, máquinas de vectores soporte, regresión logística, árboles de decisión, bosques aleatorios, calidad de transmisión, validación cruzada, circuitos ópticos.

## **Abstract**

This TFM reviews the application of machine learning techniques in optical communication systems and networks. In addition, it studies and compares the characteristics of various machine learning methods, such as: support vector machines, logistic regression, decision trees and random forests, to predict the quality of transmission when using optical circuits in wavelength routed optical communication networks. The models developed in this TFM offer better performance than previous proposals, mainly in terms of computing time, making possible its use in online mode even in highly dynamic networks, in addition to being simpler.

## **Keywords**

Machine learning techniques, support vector machines, logistic regression, decision trees, random forests, quality of transmission, cross validation, optical circuits.

# Table of contents

Resumen.....	2
Palabras clave.....	2
Abstract .....	3
Keywords.....	3
1. Introduction .....	5
1.1. State of the Art: Machine Learning Techniques in Optical Communication Systems and Networks .....	5
1.2. Problem Statement .....	11
2. Theory .....	13
2.1. Support Vector Machine (SVM) .....	13
2.2. Logistic Regression .....	14
2.3. Classification and Regression Trees (CART) .....	15
2.4. Random Forests.....	16
3. Methodology.....	17
3.1. Training Phase .....	18
3.1.1. Cross-Validation .....	18
3.1.2. Training Scenarios .....	19
3.2. Testing Phase.....	21
3.2.1. Testing Scenarios.....	21
4. Simulation Scenarios and Results.....	23
4.1. <i>MATLAB</i> Scenario: Replication of the Reference Article.....	23
4.2. <i>R</i> Scenarios .....	28
4.2.1. Decision Maker + Machine Learning Model.....	28
4.2.2. Removing the Decision Maker .....	38
5. Conclusions .....	50
6. References.....	53
A. Resumen en Español .....	56

# 1. Introduction

Over the last decade, machine learning techniques have spread rapidly throughout a remarkable number of massive and complex data-intensive fields such as astronomy, medicine, economics, commerce, security, biology, etc. These techniques are capable to perform relevant tasks by generalizing from examples, i.e. they can learn programs from data. It seems obvious, therefore, that the expansion of this discipline has not happened by chance. The fact that we are living an epoch in which colossal amount of data have been being constantly generated at unprecedented and ever increasing scales, allows machine learning algorithms to be fed with extensive amounts of examples, thereby enabling their effective training. On the other hand, this collection of data sets is so immense and complex that it is difficult to deal with it with other conventional methods. Machine learning algorithms provide, in this context, possible solutions to mine the information hidden in the data and can automatically adapt and/or optimize to their environment. In addition they make it possible in a cost-effective form by avoiding unfeasible manual programming.

## 1.1. State of the Art: Machine Learning Techniques in Optical Communication Systems and Networks

The improvement of the performance of telecommunication networks by the application of machine learnings techniques is an area under extensive research. Optical communication networks and systems have not been kept apart, but have adopted this discipline to efficiently solve setbacks derived from the exponential growth in both capabilities and complexity these networks are experiencing during the last years. The aim of this section is to introduce some of the currently considered approaches to increase the performance of optical transport networks by the use of learning mechanisms, providing a brief overview of the current research within this area.

Firstly, at the device level, machine learning techniques allow to statistically model different components of optical networks by including the underlying physics. In all these cases where a deterministic approach results in an impractical computational heavy load, learning mechanisms are becoming a promising and accurate performance improvement tool. Let's take as an example laser characterization. With the advent of advanced modulation formats aiming to increase the spectral efficiency, ranging from 16 quadrature amplitude modulation (16 QAM) to 64 QAM and beyond, the need for robust carrier frequency and phase synchronization becomes crucial. At this point, a precise characterization of the

amplitude and phase noise of the lasers is essential. Conventional time-domain approaches perform coherent detection in combination with digital signal processing (DSP) to cope with this issue [1], [2], but as higher order modulation formats are implemented, the accuracy of the phase noise estimation is compromised in the presence of moderate measurement noise. Authors in [3] present a framework of Bayesian filtering in combination with expectation maximization to accurately characterize laser amplitude and phase noise that outperforms these conventional approaches. Results demonstrate an accurate estimation of the phase noise even in presence of large measurement noise.

Erbium doped fiber amplifiers (EDFAs) are another optical network component on which these novel algorithms are being extensively applied. EDFAs are one of the key elements in which optical transport networks rely, due to their ability to extend the reach of the transmitted optical signal by performing a regeneration of dense wavelength division multiplexing (DWDM) channels in the optical domain. Notwithstanding, EDFA systems pose particular challenges in order to optimize the performance of a given link, playing a crucial role in the quality assurance of transmission. Machine learning techniques offer efficient solutions to a wide range of challenges inherent to the operation of these devices within fiber optic transmission.

Specifically, in [4] the authors define a regression problem with supervised machine learning, kernel based, to statistically model the channel dependence of power excursions in multi-span EDFA networks. The model thus constructed with historical data - past channel states and standard deviations of the output power levels -, provides the system with accurate recommendations on channel add/drop strategies to minimize the power disparity among channels derived from the combination of the automatic gain control (AGC) process and the wavelength dependent gain of each channel. The main advantages of the machine learning engine proposed in this study in relation to previously proposed solutions [5], [6] - based on deterministic models of the gain profile - is the suitability of this model for live-network non-disruptive equipment and its generalization capability to EDFA networks of different designs. Furthermore, with the arrival of flexgrid networks in which dynamic defragmentation is crucial to improve the spectral efficiency, the previous study is extended in [7] to cope with the power excursion problem in dynamically changing spectral configurations. Machine learning methods - a ridge regression model to determine the magnitude of the impact of a given sub-channel and a logistic regression to specify whether the contribution will result on an increase or decrease to the discrepancy among post-EDFA powers - are therefore used to characterize the channel dependence of power excursions for different defragmentation methods of flexgrid superchannels. The machine learning models are then incorporated into an algorithm that determines the adaptive power adjustments, reducing the problem to a single final step adjustment based on the wavelength assignments of channels. Results show a mitigation of post-EDFA power

discrepancy among channels by over 62%. Finally, a novel method to autonomously adjust the operating point of amplifiers in an EDFA cascade by using a multilayer perceptron neural network is presented in [8]. The aim of this adjustment is to optimize the performance of the link by minimizing both the noise figure and the ripple of the frequency response of the transmission system while ensuring predefined input and output power levels. The incorporation of this restriction in the power levels to the adjustment process is precisely the main contribution of the use of machine learning techniques with respect to previous studies that also sought to optimize the operating point of EDFAs in a cascade for a dynamic optical link [9]. As a matter of fact, the study presents an error lower than 0.1 dB in their attempt to maintain input and output power levels around 3 dBm while defining the gain of 6 amplifiers returning a link with a noise of figure and a frequency response ripple equal to 30.06 dB and 5.26 dB, respectively.

Another widely extended trend examines the suitability of the application of machine learning techniques in monitoring and mitigating various degrading effects affecting the performance of optical links, such as chromatic dispersion (CD), optical signal to noise ratio (OSNR), polarization mode dispersion (PMD) or differential group delay (DGD). Tendency that is moving towards the introduction also of the inherent nonlinear nature of the transmission within the optical fiber, whose influence is becoming increasingly dramatic due to the use of higher transmission rates and more advanced modulation formats, within increasingly complex architectures and heterogeneous and dynamic transmission systems.

A first approach related to the previous tendency deals with optical performance monitoring (OPM) – the estimation and acquisition of physical parameters of transmitted optical signals [10] – indispensable in ensuring robust and efficient network operation by allowing a diagnose of the network in order to take actions against malfunctions as repairing damages, driving compensators/equalizers or rerouting traffic around non-optimal links. As an example, the authors in [11] present an extensive study of the application of artificial neural networks in OPM. The study ranges from the simultaneous identification of linear impairments (OSNR, CD and PMD), to the addition of the accumulated nonlinearity to the previous impairments. Solid results are demonstrated for 40 Gb/s return-to-zero on-off keying (OOK) and RZ differential quadrature phase shift keying (RZ-DQPSK) systems. However, techniques applied in this work and similar ones [12], [13], require prior knowledge about the type of signal (bit-rate and modulation format) which is not feasible in practical because the complexity of the nodes would increase beyond their limit. In that sense, a novel technique for simultaneous lineal impairments identification (OSNR, CD and DGD) that is independent from bitrate and modulation format (providing that this information belongs to a known set) is proposed in [14]. The study is performed using principal component analysis-based pattern recognition on asynchronous delay-tap plots. Results show a considerable accuracy in the simultaneous

monitoring of linear impairments, nonetheless, the mean estimation errors of this technique are obviously increased by introducing non-linear effects.

Another recent work facing the limited scalability of the studies previously mentioned, which are based on the prior knowledge of a determined set of signals is presented in [15], where a deep neural network (DNN), trained with raw data asynchronously sampled by a coherent receiver in a 16 GBd dual-polarization QPSK, is proposed for OSNR monitoring. Results show that OSNR is accurately estimated within the range of 7.5 to 31 dB. Yet, this DNN needs to be configured with at least 5 layers and needs to be trained with 400000 samples to achieve accurate results, requiring long training time. One step beyond, an OSNR estimator and a modulation format classifier for systems employing more advanced modulation formats (up to 64 QAM) and direct detection are introduced in [16]. The former employs a neural network, while the latter uses a support vector machine (SVM), both in order to learn a continuous mapping function between input features extracted from the power eyediagram after the photodetector and the reference OSNR and modulation format, respectively. Results show that predictions from the modulation format classifier achieve an average classification accuracy of 94%, while the estimator obtains a total mean OSNR estimation error of 0.7 dB (worst case of 3.5 dB). However, results are demonstrated considering only white Gaussian noise, while ignoring for the moment linear and nonlinear optical fiber impairments.

Following the same trend, but focusing on the mitigation of nonlinearities on the transmission over optical fibers, there has been also an extensive research during the last years. Among these nonlinearities, nonlinear phase noise (NLPN) is one of the prominent factors. So far this issue has been treated with electronic methods relying on the deterministic information of the fixed fiber link, like maximum likelihood estimation [17] and digital back propagation [18] which may be computationally too heavy for practical implementation. Currently, machine learnings techniques are being incorporated to digital signal processing to mitigate nonlinearities in a more efficient way, allowing more accurate symbol detection. As an example, state-space models in combination with Bayesian filtering and expectation maximization are presented in [19] with the aim of taking into account the underlying physics of the channel and optical elements in the formulation of signal processing algorithms. As a result, an overall system improvement is achieved, including cross-polarization mitigation, carrier synchronization and optimal symbol detection. However, expectation maximization depends on the parameters of the transmission link and consequently it is not applicable to dynamic optical networks. Furthermore, authors in [20] propose a machine learning algorithm to mitigate NLPN affecting M-ary phase-shift keying (M-PSK) based coherent optical transmission systems. Specifically, the algorithm introduced is a nonlinear SVM classifier that learns the link properties from the training data without any prior information. This classifier is able to generate nonlinear decision boundaries that allows to bypass the errors induced by nonlinear impairments in



the constellations of M-PSK signals, resulting in improvements both in the maximum transmission distance (by 480 km) and launch power dynamic range (by 3.3 dB) for 8-PSK. Notwithstanding, SVM is only a binary classifier, so to deal with higher order modulation formats, many SVMs would be necessary.

With the objective of solving drawbacks derived from both previously mentioned studies, a k-nearest neighbors detector is described and demonstrated in [21]. This algorithm only needs a small set of labeled data in order to learn the link properties and generate the nonlinear decision boundaries but, unlike SVM, it performs a multi-class classification and, therefore, it is capable of classifying multiple kinds of data simultaneously. In this way, maximum transmission distance and nonlinear tolerance improvements are demonstrated in a 16 QAM coherent transmission system. Following the same line of study, a similar proposal is presented in [22]. It consists of a non-symmetric demodulation technique for receivers equipped with DSP based on clustering (using k-means algorithm), which allows to mitigate the effect of time-varying impairments such as IQ imbalance, bias drift and phase noise. The main advantages offered by the incorporation of this machine learning technique are its high computational efficiency and its transparency with respect to the nonlinearity source. The experimental results demonstrate a significant reduction of the symbol error rate in a 16QAM Nyquist system at 16 Gbaud in 250km links thanks to the creation of non-symmetric decision boundaries. Finally, a recent study [23] extends the previous proposals by introducing this type of techniques in more advanced systems, with greater spectral efficiency, such as coherent optical orthogonal frequency division multiplexing (CO-OFDM) systems. Moreover, the low computational load and execution time of this approach allows its practical implementation as opposed to previous studies [24] that doesn't allow its operation in real time. Specifically, the proposed algorithm is a non-linear equalizer SVM of reduced classifier complexity using the Newton-method (N-SVM). Results show an effective handling of inter-subcarrier nonlinear crosstalk effects in 40 Gb/s 16 QAM CO-OFDM systems and an increase of the launched optical power.

Last but not least, it is worth highlighting another different approach related to the use of learning mechanisms in improving the performance of optical networks: the application of cognition to the operation of optical networks. Cognitive dynamic optical networks [25] are capable by definition to identify current conditions, to modify their configuration according to these observations and, what is more interesting regarding the scope of this state of the art, to consider past experiences in decision-making. These functionalities make them an excellent candidate to face the increased levels of heterogeneity, both in types of services and transmission technologies, which have been mentioned throughout this section.

There are several examples of cognition worth mentioning at this point, most of them developed in the framework of the EU FP7 Cognitive Heterogeneous

Reconfigurable Optical Network (CHRON) project. In the interest of this state of the art, they will be grouped into two categories: reconfiguration of virtual topologies and quality of transmission estimation.

The first category refers to the set of lightpaths established in a given optical network, which is not designed precisely randomly but trying to optimize different parameters or performance criteria such as reducing energy consumption, network congestion, end-to-end delay or blocking probability or trying to ensure quality of transmission (QoT), etc. This virtual topology doesn't have to be statically configured but it could be dynamically reconfigured in order to better adapt to evolving traffic demands. For that purpose, a multiobjective genetic algorithm to design virtual topologies with the aim of reducing both the energy consumption and the network congestion is presented in [26]. Two different learning mechanisms, the possibility to remember both solutions successfully used in the past and connections with low QoT, are incorporated to the genetic algorithm resulting in an increase of the number of possible solutions and also in their performance. Furthermore, techno-economic studies of the introduction of cognitive techniques in virtual topology design exhibit also significant savings in terms of the total cost of ownership compared to conventional methods. As a matter of fact, savings up to 20% and 25% in capital and operational expenditures via a genetic algorithm-based method are demonstrated in [27]. One step forward, authors in [28] use monitored data to produce estimations that can help to anticipate changes in the traffic and proactively reconfigure the virtual network topology. Finally, outside CHRON project, an algorithm to identify/locate failures in the virtual network topology that can lead to an unacceptable quality of service is proposed in [29]. It consists on a previous experimental characterization of several failure causes that are categorized by a clustering algorithm and then used to train a Bayesian network (BN). This trained BN is used to localize and identify the most probable cause of failure impacting a given service. Additionally, a different virtual network topology reconfiguration approach is introduced in [30]. It performs reconfigurations based on the traffic volume and direction predicted by an artificial neural network proposed for every origin-destination pair. Periodically, collected monitoring data are transformed into modelled data and the artificial neural networks are used to predict the next-period traffic. Results show savings both in capital and operational expenditures.

The last topic to be mentioned in this section regards quality of transmission estimation, particularly relevant in impairment-aware optical network design and operation. As starting point, a quality of transmission estimator, QTool, has already been proposed in [31]. It computes the associated Q-factors of a set of lightpaths, given a reference topology. These Q-factors are indicators of the QoT as they are closely related to the bit error rate (BER) but, although the estimates of the QTool are relatively accurate, its use is impractical in scenarios where time constraints are important due to its high computing time. Several approaches propose cognitive

techniques to solve this serious drawback. As an example, authors in [32] present a QoT estimator capable of exploiting previous experience and thus, provide with fast and correct decisions on whether a lightpath fulfils QoT requirements or not. It is based on case-based reasoning (CBR) [33], an artificial intelligence mechanism which offers solutions to new problems by retrieving the most similar cases faced in the past whether by reusing them or after adapting them. Cases are retrieved from a knowledge base (KB) that can be updated to include new experiences. The estimator proposed is of a hybrid character, with an initial step in which only the length of the lightpath is taken into account and, if necessary, a second step is performed using the CBR system. The results show a percentage greater than 99% in correct classification of lightpaths into high/low QoT and more important, three orders of magnitude faster than the QTool mentioned above. Furthermore, this study is experimentally demonstrated in a WDM 80 Gb/s PDM-QPSK testbed [34], where, even with a very small KB, successful classifications are achieved in a percentage between 79% and 98.7%. Notwithstanding, the update of the KB previously described with new experiences (learning process) may have a great impact on the retrieval time as the size of the KB is increased unrestrainedly, putting at risk the benefits of storing the knowledge. For this reason, the study in [32] is enhanced with the introduction of learning and forgetting techniques in a maintenance phase of the KB [35]. As a result, cases that do not improve the performance of the system are eliminated, which entails a significant reduction of the KB size and consequently a reduction of the computing time without neglecting the percentage of successful assessments, which is similar to the previous situation. A last recent work, outside CHRON project, which also deals with the QoT estimation is presented in [36]. Unlike previously mentioned studies, authors apply a machine learning based classifier, specifically a random forest classifier, to predict the probability that the BER of a candidate lightpath will not exceed a given threshold.

## 1.2. Problem Statement

The starting point is the article [35] that has been exposed at the end of the section dedicated to the state of the art. The problem posed in it is a binary type classification in which lightpaths are classified into two categories: high QoT and low QoT. This kind of problems can be solved with machine learning techniques in a very efficient way and for this reason it is worth studying the benefits offered by these novel techniques in the scenario proposed in the article.

It should not be forgotten that the prediction time for this scenario is a critical design point, especially when operating in on-line mode in highly dynamic networks. Machine learning techniques can be very efficient precisely for this requirement and could offer smaller prediction times with comparable or even better accuracy rates that might allow their practical use.

In addition, the article proposes a decision maker that is designed to address those lightpaths that are outside a certain uncertainty area, whose threshold lengths are artificially constructed by merely observing data, at guess. This point may be clearly improved by machine learning techniques, since they could learn by themselves to recognize those areas and their possible exceptions, making the use of this decision maker completely unnecessary.

Finally, a network operator might be interested in implementing a system that predicts QoT by minimizing the existence of false positives (although this would lead to an increase in false negatives), since they can seriously impair the quality of service. This degree of design cannot be easily managed by the system proposed in the article of reference, while machine learning techniques do offer the possibility of optimizing specificity versus sensitivity.

## 2. Theory

As the main objective of the master degree is to introduce the student to the research process, special emphasis has been placed on the sections corresponding to the state of the art and its references, the methodology, the exposure of the results of the investigation and the conclusions.

This section, which includes theoretical aspects about machine learning techniques, does not pretend, however, to go deeply into the rigorous and detailed explanation of the models used, rather it intends to briefly outline their main characteristics.

All methods used in this thesis perform supervised learning, i.e., they are trained with labelled data.

### 2.1. Support Vector Machine (SVM)

Support Vector Machine (SVM) is mainly a classifier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

The optimization problem posed by this model for the training phase is the following:

$$\begin{aligned} \text{Minimize}_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha - 1^T \alpha \\ \text{subject to} \quad & y^T \alpha = 0 \\ & C \geq \alpha \geq 0 \end{aligned}$$

where  $\alpha$  is the so-called support vector,  $C$  acts as a regularization parameter (more regularization involves a lower  $C$ ) and  $Q$  is a  $N$  by  $N$  matrix:

$$Q_{nm} = y_n y_m (z_n^T z_m)$$

where  $z$  represents the nonlinear transformation of  $x$

The inner product ( $z^T z'$ ) may be calculated without explicitly transforming  $x$  and  $x'$  to  $Z$  space with the help of a kernel function.

Particularly, in this thesis a Gaussian kernel or Radial Basis Function (RBF) kernel is used:

$$K(x, x') = e^{-\gamma \|x - x'\|^2} \quad (\gamma > 0)$$

As a result, two different parameters may be optimized during the training process.  $C$ , as explained above, acts as a regularization parameter avoiding overfitting of the model and  $\gamma$  (gamma, as mentioned later in the methodology section) determines the “width” of the Gaussian centred in the support vectors.

## 2.2. Logistic Regression

Logistic regression is a statistical method for analysing a dataset in which there are one or more independent variables or predictors that determine a dependent variable, an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

The goal of logistic regression is to find the best fitting model to describe the relationship between the outcome and the predictors. This is performed by minimizing the following cost function:

$$\underset{w}{\text{Minimize}} J(w) = \frac{1}{N} \sum_{n=1}^N \ln(1 + e^{-y_n w^T x_n})$$

where  $y_n$  are the outcomes,  $x_n$  represents the predictors, and  $w$  are the weights.

Two different variables can be optimized during the training phase for logistic regression in  $R$ :

- Alpha (also known as mixing percentage in  $R$  programming environment): used to establish the kind of regularization performed during the training phase. For  $\alpha=0$ , ridge regression is used as regularization method trying to avoid overfitting by penalizing large coefficients through the L2 Norm. For  $\alpha = 1$ , lasso regression is used as regularization method, in this case trying to avoid overfitting by penalizing large coefficients through the L1 Norm.
- Lambda: the regularization parameter (lambda = 0 implies no regularization).

## 2.3. Classification and Regression Trees (CART)

CART refer to Decision Tree algorithms that can be used for classification or regression predictive modelling problems. A learned binary tree is actually a partitioning of the input space. New data is filtered through the tree and lands in one of the partitions and the output value for that partition is the prediction made by the model.

In this way, creating a CART model involves selecting input variables and split points on those variables until a suitable tree is constructed. The selection of which input variable to use and the specific split or cut-point is chosen by applying a recursive binary splitting (a numerical procedure where all the values are lined up and different split points are tried and tested using a cost function, selecting the split with the best cost).

For classification the Gini index function is used. This function provides an indication of how mixed the training data assigned to each node is:

$$G = 1 - \sum_j p_j^2$$

The recursive binary splitting procedure needs to know when to stop splitting as it works its way down the tree with the training data. One way to do it is to use a minimum count on the number of training instances assigned to each leaf node. If the count is less than some minimum then the split is not accepted and the node is taken as a final leaf node.

The count of training members is tuned to the dataset. It defines how specific to the training data the tree will be (if it is too specific, it will entail overfitting problems).

The stopping criterion is important as it strongly influences the performance of a given tree. Pruning can be used to further lift performance after learning phase is completed.

The complexity parameter (cp in R) is defined as the number of splits in the tree. This parameter can be optimized by R in the training phase. Simpler trees are preferred, since they are easy to understand and they are less likely to suffer from overfitting.

## 2.4. Random Forests

Random Forests are another learning method for classification and regression. This algorithm operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes, for classification problems, or mean prediction, for regression tasks, of the individual trees. Its principal advantage with respect to CART is that it is more efficient in avoiding overfitting.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a certain training set with the corresponding outcomes, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples. After training, predictions for unseen samples can be made by averaging the predictions from all the individual regression trees on these unseen samples. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias.

However, Random forests slightly differ from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the sampled trees, causing them to become correlated.

The number of randomly selected predictors (`mtry` in R), variable optimizable during the training phase, refers to the previously mentioned random subset of features.



# 3. Methodology

The main objective of the present study is to estimate the Quality of Transmission (QoT) in the context of wavelength routing optical communication networks (WRON) with the help of the machine learning techniques described previously and to exploit the advantages that these algorithms might offer. The starting point will be the work presented in [35], which describes and demonstrates the benefits of the employment of a cognitive QoT estimator of a hybrid nature. This estimator basically consists of two elements: a decision maker that only takes into account the length of the lightpath and a Case-Based Reasoning (CBR) module that classifies a given element according to the most similar case stored in its Knowledge Base (KB). The former is in charge of classifying all those lightpaths whose length is above or below the limits that identify a given uncertainty area in which, in principle, the QoT cannot be firmly predicted, while the latter will try to classify indeed those inside the uncertainty area.

With this background, the approach applied in this research can be summarized in the following steps:

1. Replicate the previous work recreating the exact conditions in which it was performed and replacing the CBR module with a machine learning based one, a Support Vector Machine (SVM).
2. Prove, in these conditions, that the percentage of accuracy in QoT classification achieved is similar or better for this new technique and that it is capable of performing this classification in a shorter computation time per lightpath.
3. Recreate the study again in a different programming environment (R), “friendlier” in terms of machine learning techniques comparisons (using the caret package [37]).
4. Compare SVM performance with the performance of other machine learning algorithms (logistic regression, Classification and Regression Trees and Random Forests), within the new programming environment, with the aim to identify the one offering the best performance.
5. Design a new scenario in which the decision maker (built *a priori*, by merely observing the data) is removed and the machine learning based modules learn to classify lightpaths QoT regardless of whether or not they belong to that artificial uncertainty area.

To better characterize the previous process, two different phases are described below: training phase and testing phase.

## 3.1. Training Phase

As an indispensable requirement to be able to carry out the training, it is necessary to provide the machine learning algorithm with a set of data. For the sake of a fair comparison, the set of data used is the same as the one used in the reference article, which corresponds to the different lightpaths established by running previous off-line simulations, with different loads, after solving the routing and wavelength assignment (RWA) problem, for the networks under study. The cases that compose this set of data consist of the description of each lightpath (i.e., a set of attributes) and its associated Q-factor (we are faced, as explained in previous sections, with supervised learning). The description of the lightpath contains the source and destination nodes, the set of links it traverses (represented by the percentage of its individual contribution to the total length of the lightpath), the corresponding wavelength, its total length, the sum of co-propagating lightpaths per link and the standard deviation of that number. In addition, the associated Q-factor was calculated off-line by means of the Q-Tool [31]. Finally, in order to classify the different cases into high and low QoT a threshold is set to 16.9 dB (which corresponds to a BER of  $10^{-12}$ ), as in the reference article.

### 3.1.1. Cross-Validation

It is important to emphasize that the proposed training phase uses cross-validation in order to characterize the model more precisely and thus facilitate its comparison with others. Specifically, 10-fold cross-validation is used, so that training data is divided into 10 folds and 10 iterations are performed. At each iteration a fold is removed and used to evaluate the model obtained after training with the remaining ones. In this way an average performance (e.g., accuracy) with greater statistical significance can be obtained. The creation of the random folds is controlled by a seed, ensuring a fair comparison between models.

In addition, during the cross-validation process, the best tuning of parameters for each type of automatic learning algorithm is also sought. The model is evaluated, in terms of a certain performance variable, for each of the different values (or combination of values) of these tunable parameters. At the end of the cross-validation process, the average performance (eg, accuracy) of each of the 10 iterations using the best tuning is obtained. The result is an optimized model with statistical significance. The following table (Table I) shows the different types of tunable parameters used in the optimization process, which were explained in the section on theoretical concepts.

ALGORITHM	TUNABLE PARAMETER	RANGE	NUMBER OF ELEMENTS
SVM	C	$[2^{-5}, 2^{15}]$	11
	gamma	$[2^{-15}, 2^3]$	10
LOGISTIC REGRESSION	alpha	Chosen by R	10
	lambda	$[0.1, 1]$	10
CART	complexity parameter	Chosen by R	8
RANDOM FORESTS	mtry	Number of predictors	8

*Table 1. Tunable parameters for the Cross-Validation optimization process*

Caret package [37] has been used to carry out the cross validation process in the proposed scenarios in R. In particular, method “svmRadial” is applied for SVM training (i.e., using a (Gaussian) radial basis function kernel, or RBF kernel), method “glmnet” and family “binomial” for logistic regression, method “rpart” for CART and method “rf” for Random Forests.

Finally, it is important to note that the performance of the proposed models may be evaluated during the cross-validation process not only in terms of accuracy. In fact, depending on the scenarios described below, the area under the ROC curve (AUC) is also taken into account as an optimization parameter. The ROC curve allows to visualize the trade-offs between sensitivity and specificity in binary classifiers. These metrics characterize the model ability of identifying true positives (TP), or true negatives (TN) as follows:

$$Sensitivity = \frac{TP}{(TP + FN)}$$

$$Specificity = \frac{TN}{(FP + TN)}$$

where FN refers to false negatives and FP refers to false positives. An AUC of 1 would represent the ideal scenario in which every positive and negative instance is correctly identified.

### 3.1.2. Training Scenarios

When it comes to training it will be necessary to consider the type of scenario to emulate. In every situation, a testing data set is randomly selected and removed

from the data base so as to avoid “data snooping”, i.e. completely isolating it from the training phase.

For the case where the reference article (points 1 and 2 of the summarized methodology) is replicated, the set of training data consists only of randomly selected cases within the uncertainty area (emulating the CBR module by the employment of the SVM). The training process is repeated 100 times (i.e., with 100 different sets) to obtain results with great statistical significance. All this is also repeated for a different number of training cases.

This study is performed in MATLAB, in order to establish a fair comparison with the article. As output variables, besides the already trained and optimized model, the training time, which involves the whole process of cross-validation and the so-called “Accuracy In”, which refers to the percentage of lightpaths correctly classified when evaluating the cases used for training, are calculated.

For the next scenario (points 3 and 4), the joint action of the decision maker and the machine learning module is simulated for different algorithms, in a different programming environment (R). In this case, the set of training data consists again of randomly selected cases within the uncertainty area, but the models are trained only once, performing the cross-validation with a significant difference: not only the accuracy is considered as the optimization parameter but also an alternative scenario is proposed considering the AUC, which allows to optimize the trade-offs between sensitivity and specificity in these binary classifiers. A seed is used to ensure that the models are trained with the same random data sets in honour of a fair comparison of the algorithms (SVM, CART, logistic regression and Random Forests). As output variables of the training phase, the training time is again obtained and also the so-called “Accuracy-CV” (accuracy obtained in the cross-validation process) or the AUC-CV, depending on the scenario.

The last scenario (point 5) is trained with cases belonging both to the uncertainty area and outside of it. The models are trained once and the same seed as before is considered for fair comparisons. At the end, training data sets are the same as in previous scenario with the addition of cases not belonging to the uncertainty area and randomly selected, in such a number that the same proportion of cases within the uncertainty area and outside of it is kept as it appears in the whole data base. In addition, both the accuracy and the AUC are used as optimization parameters for the cross-validation process and same output variables as before are calculated.

## 3.2. Testing Phase

For all the scenarios proposed above, except for the last one, low and high thresholds of the uncertainty area need to be established. Logically, they are considered to be the same as in the reference article (Table II).

NETWORK	NUMBER OF WAVELENGTHS	LOW LENGTH LIMIT (KM)	HIGH LENGTH LIMIT (KM)
DT	32	975	1875
	64	975	2050
GÉANT2	32	1250	4125
	64	1175	4225

Table II. Low and High Thresholds of the Uncertainty Area

### 3.2.1. Testing Scenarios

For the testing phase, all the lightpaths may be considered for every scenario regardless whether they belong or not to the uncertainty area. As mentioned in the training phase, a testing dataset is randomly selected and removed from the database before training cases are selected.

For the situation where the reference article (points 1 and 2 of the summarized methodology) is replicated, the test data set is randomly selected and then separated into two different sets: one corresponding to the uncertainty area, which is used to evaluate the performance of the optimized SVM model and another one containing the remaining cases, which feeds the decision maker (classifying a lightpath as low quality if its length is above the upper threshold of the uncertainty area or as high quality if it is below the lower limit). As in the corresponding training scenario, the process is repeated 100 times, generating statistically significant results. The study is carried out on the same machine as in the reference article (a Debian GNU / Linux 6.0 machine) and, obviously, implemented in MATLAB, in order to establish a fair comparison both in terms of accuracy and computing time per lightpath. As output variables, the so-called "Accuracy Out", which refers to the percentage of testing lightpaths correctly classified by the whole system (decision maker and SVM model) and the computing time per lightpath (the time it takes the decision maker and the SVM model to classify their corresponding lightpaths divided by the total number of cases to be classified) are calculated.

The next scenario (points 3 and 4) proposes a similar analysis to the previous one, but in a different programming environment (R). In this case, ten different test data set are randomly selected to evaluate the whole system performance, dividing these data in two different sets as previously described. A seed ensures these test datasets are the same for every trained model (SVM, CART, logistic regression and Random Forests). Same output variables as in the previous scenario are obtained with the addition of the AUC (in this case, area under the ROC curve derived from the result of the classification of testing lightpaths corresponding to the uncertainty area carried out by the different trained models).

Finally, in the last scenario (point 5), the evaluation of the machine learning module takes place with the whole test data sets (no elements are separated in order to be classified by a decision maker). A seed ensures test datasets are the same for every algorithm as in the previous scenario. The Accuracy Out, the AUC and the computing time per lightpath are obtained as output variables.

## 4. Simulation Scenarios and Results

To evaluate the performance of the different machine learning algorithms proposed, simulations have been carried out in two different networks (depending on the scenario, both may be compared or just one is studied, as explained along this chapter): a long haul network, the 14-node Deutsche Telekom (DT) network [31], and an ultra-long haul network, the 34-node GÉANT2 network [38]. 32 and 64 wavelengths per link have been considered for both networks with the same assumptions regarding spans or transceivers as in the article of reference [35].

### 4.1. *MATLAB* Scenario: Replication of the Reference Article

Figure 1 shows the successful classification of lightpaths carried out by the joint action of the decision maker (classifying lightpaths outside the uncertainty area) and the SVM module (classifying lightpaths inside the uncertainty area) for DT network, comparing it with the different implementations of the CBR proposed in the reference article. The number of training cases is set from 500 to 5000 in line with the number of cases of the Knowledge Base (KB) of the CBR. Like in the study of reference, the number of testing cases is set to 6000.

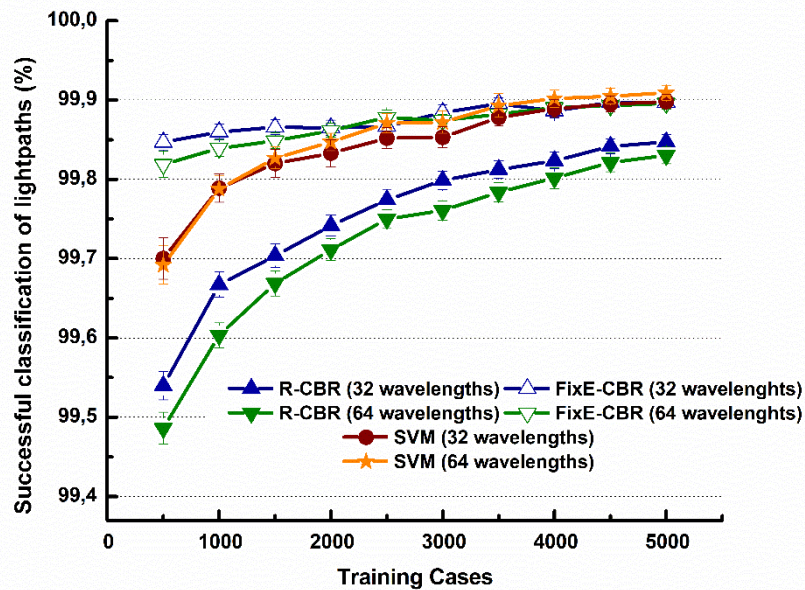


Figure 1. Successful classification of QoT when comparing R-CBR, FixE-CBR and SVM methods for DT network

R-CBR refers to the situation where the KB of the CBR is completely static, and so it is not updated with new cases nor optimized (cases are classified by retrieving the most similar case in the KB), while FixE-CBR refers to the situation where an optimization phase for the KB is performed by the employment of learning and forgetting techniques with the aim to store only the most significant cases [35]. It is worth mentioning that, unlike in this article, FixE-CBR method is represented taking into account the initial size of the KB in order to offer a clearer visualization of the results, although its final size after the optimization phase would be smaller. Results have been obtained after repeating the corresponding training and testing phases 100 times, as mentioned in the methodology chapter. Average results are represented together with 95% confidence intervals (although in some cases the size of the symbols are bigger than the confidence interval).

As shown in the figure, SVM method is able to improve the performance of the R-CBR method regardless the number of training cases. When compared to the optimized CBR, SVM seems to offer a lower performance for smaller sizes of the training set, but the comparison is not completely fair. FixE-CBR KB is randomly populated with cases belonging to the uncertainty area at a first stage, but this method performs an off-line KB optimization process on a second stage by classifying 6000 new lightpaths belonging to the uncertainty area (the optimization process is run after every 500 classifications). As a result a number of redundant cases are removed and a number of significant new ones are stored. Once the optimization process has finished, the performance is analysed by classifying a new set of 6000 lightpaths. In consequence, CBR is somehow “trained” during the optimization process with a significantly higher additional number of cases than the ones used for SVM training. The proportional difference in the number of cases between the ones involved in the optimization process of the FixE-CBR method and the SVM module is reduced as the training dataset size is increased. As a result, SVM achieves a comparable performance to FixE-CBR with 2500 training cases and outperforms it from 3500 onwards. As shown in the figure, SVM configuration achieves up to more than 99.9% of successful classifications of lightpaths.

At this point, the reader may be wondering whether this behaviour is similar for the GÉANT2 network or if it suffers from scalability problems. The truth is that, unlike in the reference article, simulations were run for GÉANT2 network for training cases ranging from 500 to 5000 (5000 to 50000 in the study of reference). The reason for this restriction can be noticed in Figure 2, where the training time (including all the cross-validation optimization process) for the SVM algorithm is presented for both DT and GÉANT2 networks. Average results are shown as in the previous figure together with the 95% confidence level.

As depicted in the figure, training time is increased exponentially with the size of the training dataset. Furthermore, as GÉANT2 network contains a significantly greater number of links and nodes, the number of predictors used in



the SVM training phase increases dramatically, and so does the training time (becoming prohibitive for a 50000 training dataset size, not to mention the situation in which the training phase is repeated 100 times).

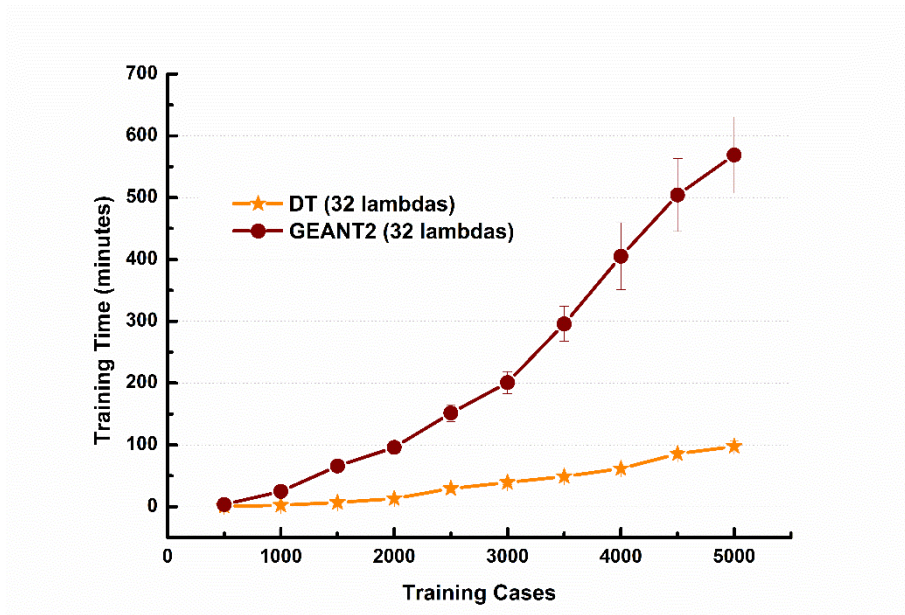


Figure 2. Time consumed in order to train the SVM module for DT and GÉANT2 networks

Notwithstanding, the performance of the SVM module for the GÉANT2 network can be compared with the CBR based modules in Figure 3, where the percentage of successful classification of lightpaths in the uncertainty area is presented. In this case, the training dataset ranges from 500 to 5000 cases. This figure compares R-CBR and SVM performance for both networks, together with the performance obtained by betting for the most likelihood category (emulating a decision maker within the uncertainty area).

As depicted in Figure 3, for a 5000 size of the training dataset for the SVM module, the percentage of successful classifications reaches up to 98.73% for DT network and 91.79% for GÉANT2 network, outperforming R-CBR in more than one percentage point (97.5% and 90.56% for R-CBR, respectively). While for the DT network a saturation point seems to be being reached with 5000 training cases, for the GÉANT2 network the improvement offered by SVM over R-CBR increases as the number of cases grows. Since the decision maker is implemented in the same way in this study and in the reference article, there is no reason to suspect that in the scenario in which GÉANT2 is trained with up to 50,000 cases, SVM would not behave better than CBR (although training time would be prohibitive). Finally, the fact that GÉANT2 is offering a significant lower percentage of successful classifications is related to both the significantly bigger size of the network and the more complex structure of the data as explained in [35] (only 8% of the cases belong to the uncertainty area and thus are not solved by the simple decision maker for DT network, while this percentage rises up to more than 50% for GÉANT2 network).

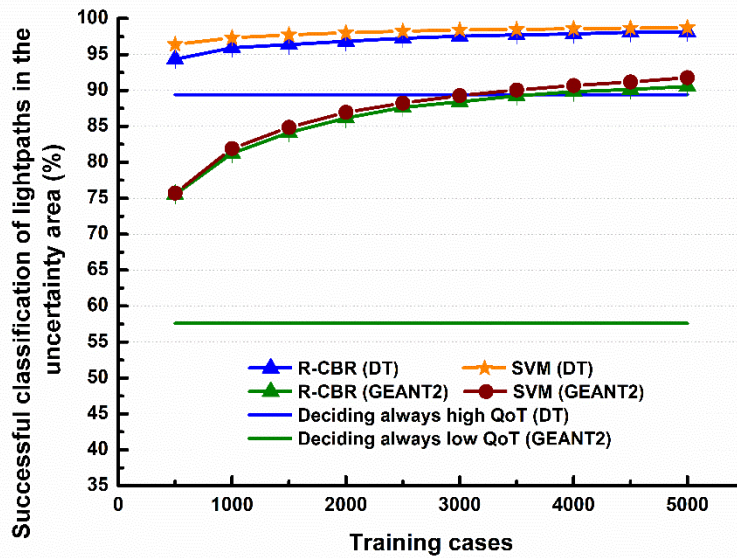


Figure 3. Successful classifications of R-CBR and SVM in the uncertainty area for DT and GÉANT2 networks with 32 wavelengths and successful classifications in the uncertainty area betting for the most likelihood category.

So far, SVM has proved to be more accurate in classifying lightpaths in low and high QoT categories, but since in the dynamic operation of a network not only the QoT of each new lightpath to be established must be assessed, but also that of co-propagating ones, a low computing time per lightpath is required, especially in highly dynamic networks.

Figure 4 displays the computing time to assess the QoT per lightpath for the DT network for 32 wavelengths using both CBR methods, SVM and the Q-Tool. As shown in Figure 4, SVM outperforms both CBR methods. In particular, the computing time for SVM is around 40 times faster than that of the R-CBR for 5000 training cases. It can be observed that the difference between R-CBR and SVM computing times is increased with the size of the training dataset. The reason is that, while SVM classifies lightpaths using a mapping function and thus the computing time remains almost constant through the training dataset sizes range (being more influenced by the decision maker), the computing time for CBR is related to the size of the KB and the needed time to retrieve the most similar case from it. For FixE-CBR the KB is reduced in the optimization process and thus the computing time is lower than that for R-CBR. Notwithstanding, SVM outperforms also the optimized CBR, being approximately 5 times faster for 5000 training cases.

In fact, SVM method trained with 5000 cases is faster than both CBR based methods regardless the number of cases in the KB. In this way, a system performing SVM classification after training it with 5000 cases would be faster than a CBR based one with an initial (or permanent) KB size of 500 cases and would significantly

outperform the accuracy in classification of lightpaths in the uncertainty area as it was shown in Figure 3 (98.73% vs. 94%).

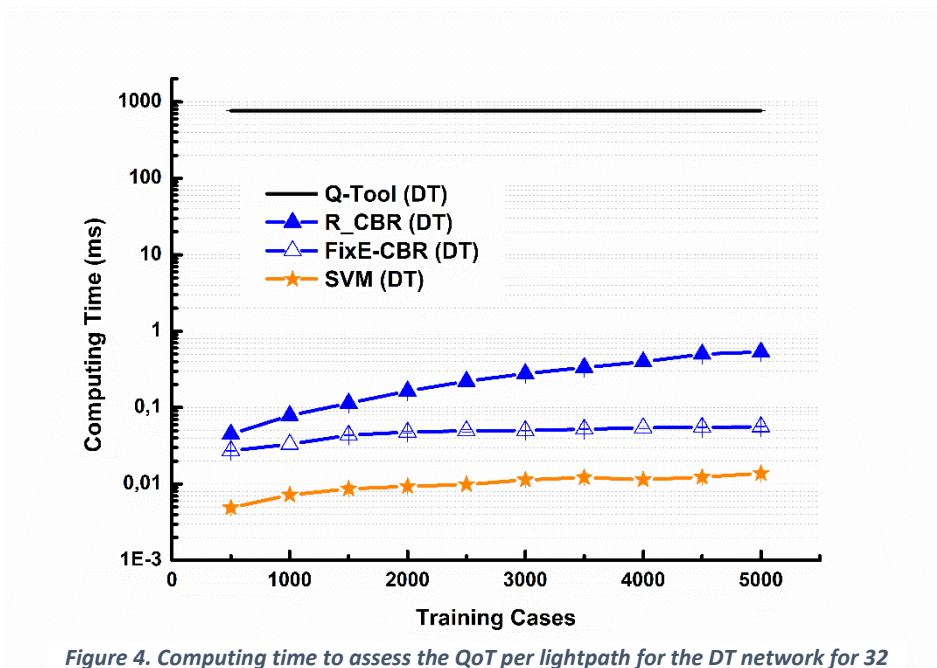


Figure 4. Computing time to assess the QoT per lightpath for the DT network for 32 wavelengths using the Q-Tool, R-CBR, FixE-CBR and SVM methods.

Lastly, Figure 5 presents the generalization error of the SVM models trained in this section for DT network. This measure of error is crucial when it comes to qualifying a model based on machine learning as better or worse, since it characterizes its predictability. The generalization error is defined as the difference in performance (percentage of success in classification, in this case) that is obtained

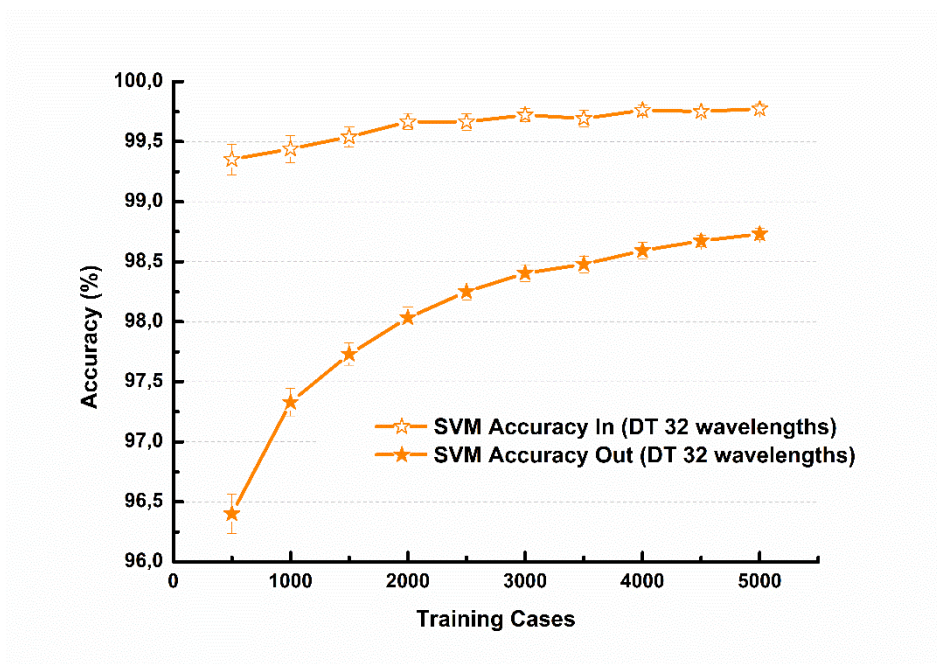


Figure 5. Generalization error for SVM models for DT network

when testing the optimized model with the own training data or "Accuracy In" and that obtained when testing the model with the test data belonging to the uncertainty area or "Accuracy Out". It is foreseeable to find a lower Accuracy Out value, as long as the test data is not used in any way in the training phase, since the model is optimized for the training data. However, as shown in the figure, as the training cases are increased, the generalization error decreases. Logically, the more cases are "learnt" by the model, the more accurate will be the decision boundary when classifying.

## 4.2. *R* Scenarios

Once SVM was proved to be more accurate and faster when it comes to classifying lightpaths into low and high QoT categories than CBR in the exact same conditions, the study is transferred to a different programming environment, R, where the caret package may be very helpful for training, optimizing, characterizing, evaluating and comparing models. The aim is to check if other machine learning algorithms may outperform SVM for this specific problem.

All *R* scenarios randomly select 10 completely different test datasets of 6000 lightpaths regardless their total length. A seed ensures all models and scenarios are tested with the same 10 datasets, which are removed from the database before collecting the training set.

### 4.2.1. *Decision Maker + Machine Learning Model*

The scenario in which a machine learning model is combined with a decision maker in order to carry out the QoT classifications is replicated again. In this case, only the DT network with 32 wavelengths is studied, as the analysis for 64 wavelengths does not yield results of significant interest compared to it. Also, GÉANT2 network entails prohibitive training times for SVM as previously shown. Four different machine learning algorithms are compared: SVM, CART, logistic regression and Random Forests, and two different optimizations are performed during the cross-validation process: accuracy and AUC. As mentioned in the methodology section, testing cases will be separated depending on their total length, feeding the machine learning algorithm or the decision maker whether they belong to the uncertainty area or not. Training phase involves a unique training of every model with different sizes of the training set, which comprises cases belonging to the uncertainty area.

#### 4.2.1.1. *Accuracy Optimization*

In this section, results when the optimization process of the cross-validation is carried out looking for the highest possible value of Accuracy-CV are exposed.

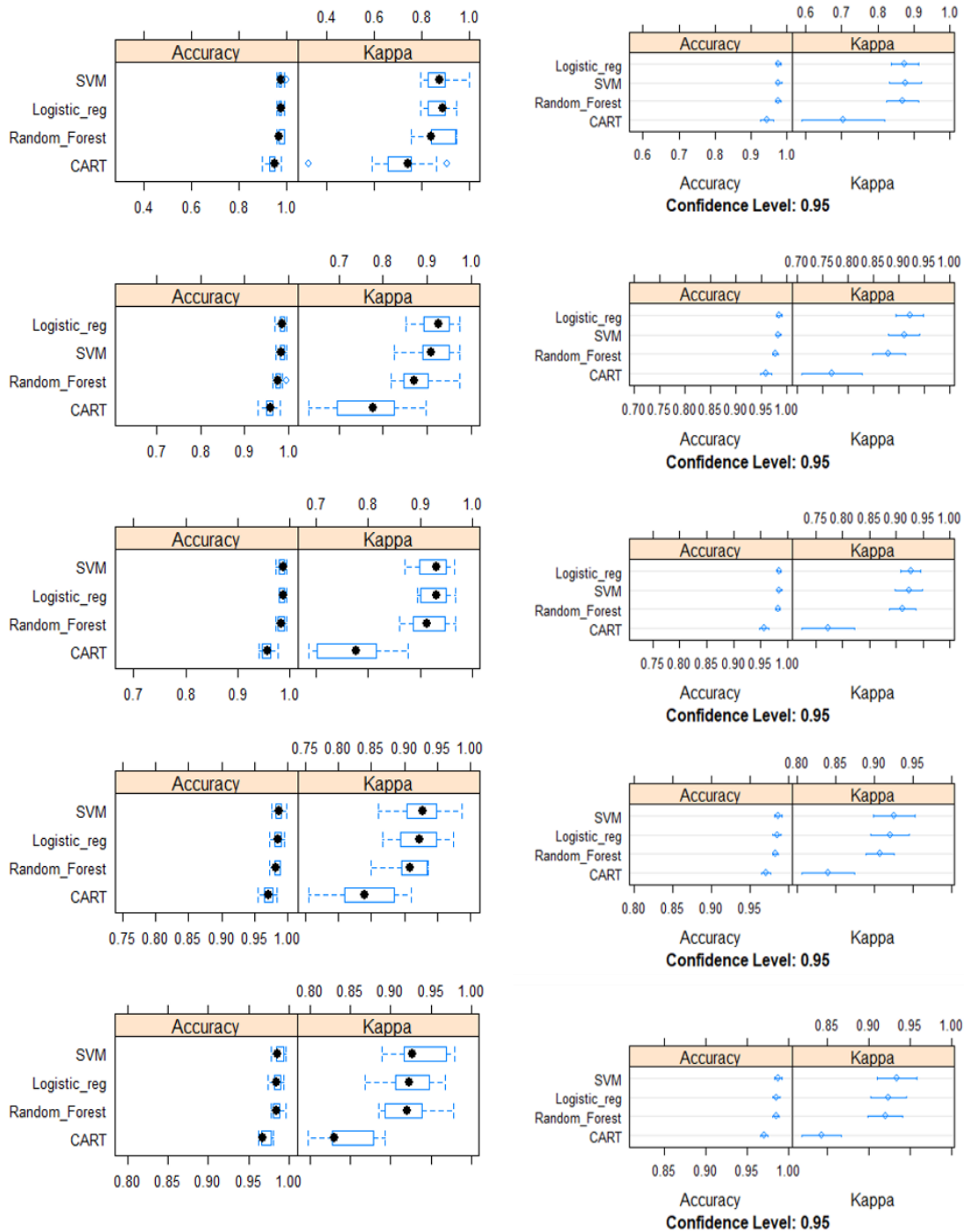


Figure 6 Performance in terms of Accuracy and Kappa of the cross-validation process for DT network (32 lambdas)

Figure 6 shows the performance offered by the four different machine learning algorithms in terms of Accuracy and Kappa derived from the training phase. On the left column, the boxplot (including the median, first quartile and third quartile) is depicted, while the average value together with the 95% confidence intervals are depicted on the right. Rows represent the size of the training data set, ranging from 1000 to 5000 from top to bottom. In addition, in every cell, the algorithms are already ordered from best to worst, in terms of median (left) and mean (right). Accuracy corresponds to the “Accuracy CV” already mentioned along

this study. Kappa is a metric that compares an observed accuracy (the number of instances that were classified correctly) with an expected accuracy (the accuracy that any random classifier would be expected to achieve) as in the following formula:

$$Kappa = \frac{(observed\ accuracy - expected\ accuracy)}{(1 - expected\ accuracy)}$$

That way, the smaller the expected accuracy for the same observed accuracy the greater the Kappa. Models with a greater Kappa are therefore more predictable, better in terms of the performance they might offer after the testing phase (classification of new instances).

Summing up, Figure 6 shows an expected evolution of every model towards higher performance as the training dataset grows. In order to select the best one among them, a compact confidence interval with a higher mean value together with a higher median value and a compact boxplot moved to the right are the desirable conditions. It is not easy to select the best candidate, three of them, SVM, Random Forest and logistic regression, seem to offer quite similar performance both in terms of accuracy and kappa. At most CART could be discarded, although it is offering indeed a very high performance.

Figure 7 shows the average Accuracy CV together with the 95% confidence intervals, employing a finer granularity, for each model. As previously discussed, every model seems to offer a high performance in terms of accuracy and only one can be discarded as the ideal candidate (CART) since the remaining three achieve a very similar average accuracy of around 98,5%.

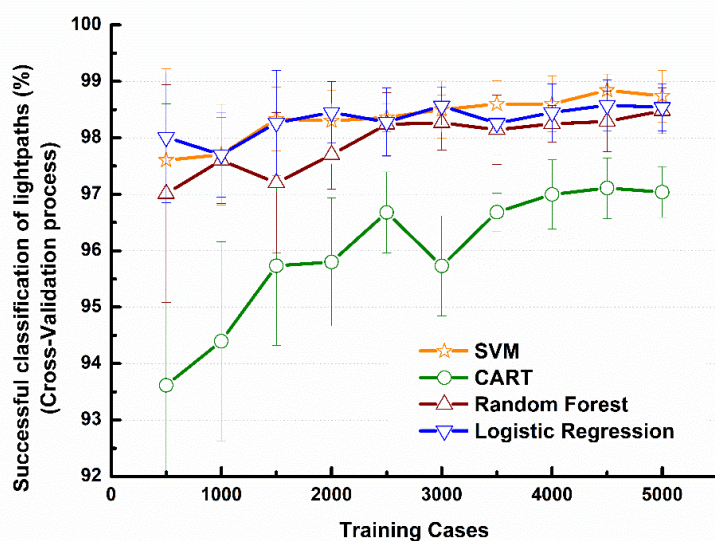


Figure 7. Percentage of successful classification of lightpaths during the cross-validation process for SVM, CART, Random Forest and logistic regression

There are other variables that can be considered to characterize these models. As an example, Figure 8 represents the training time for each model depending on the size of the training data set.

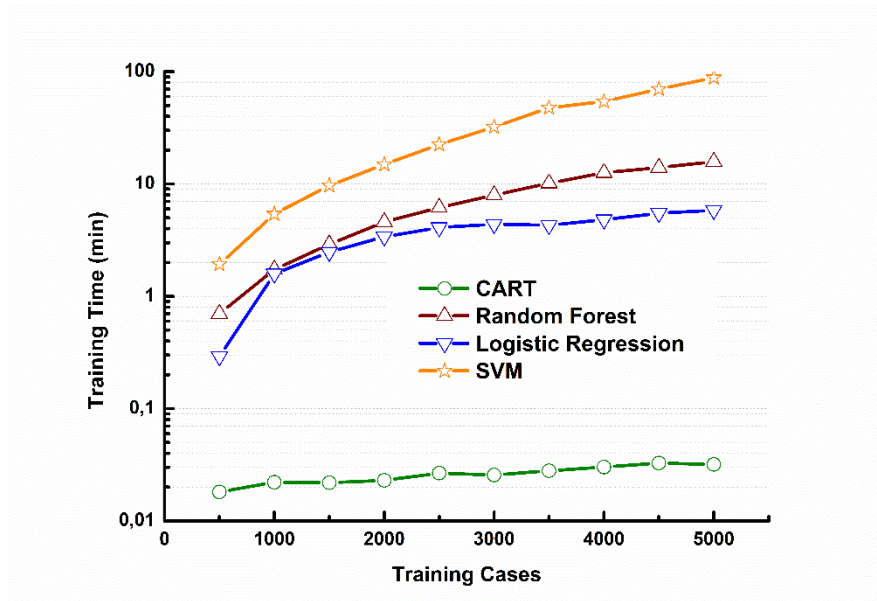


Figure 8. Training time in minutes for CART, SVM, logistic regression and Random Forest models

As shown in Figure 8, although CART was discarded as the best candidate due to its lower accuracy-cv, it is by far the fastest in the training phase. As a matter of fact, it takes from 2 up to 4 magnitude orders more to train the other ones when compared to CART models. On the other hand, SVM is the slowest, being one magnitude order slower than Random Forest and logistic regression and what is more, SVM is more penalized by the size of the training dataset, resulting in greater differences with respect to the others as the number of training cases grows. Summing up, Random Forest and logistic regression seem to be taking the lead.

So far, only the training phase has been taken into account. The following figures show the results after the testing phase.

Figure 9 depicts the accuracy out (the accuracy obtained after classifying the testing cases inside the uncertainty area employing the machine learning model together with the accuracy obtained by the decision maker). Not many differences are observed when comparing this figure with Figure 7. It is only worth mentioning that the global accuracy is higher (thanks to the decision maker, which was artificially designed to classify correctly 99.99% of lighpaths outside the uncertainty area) and that the levels of accuracy achieved are comparable to the ones obtained for the MATLAB scenario for the three best candidates: SVM, Random Forests and logistic regression.

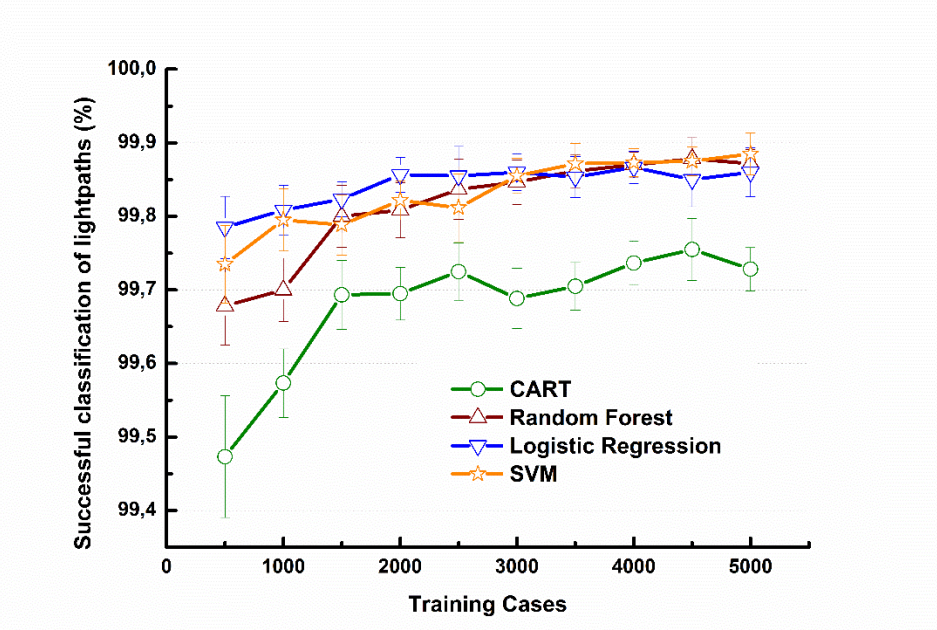


Figure 9. Successful classification of lightpaths of the whole system (combination of the decision maker with CART, Random Forests, logistic regression and SVM )

Another metric that is worth studying once the testing phase is concluded is the AUC, defined in this scenario for the classification of testing lightpaths within the uncertainty area performed by the machine learning model. Figure 10 shows the average AUC together with the 95% confidence intervals for CART, Random Forest, logistic regression and SVM. All models present an AUC close to 1, especially for Random Forests, SVM and logistic regression, in that precise order, meaning that they perform classifications very accurately, identifying correctly both categories.

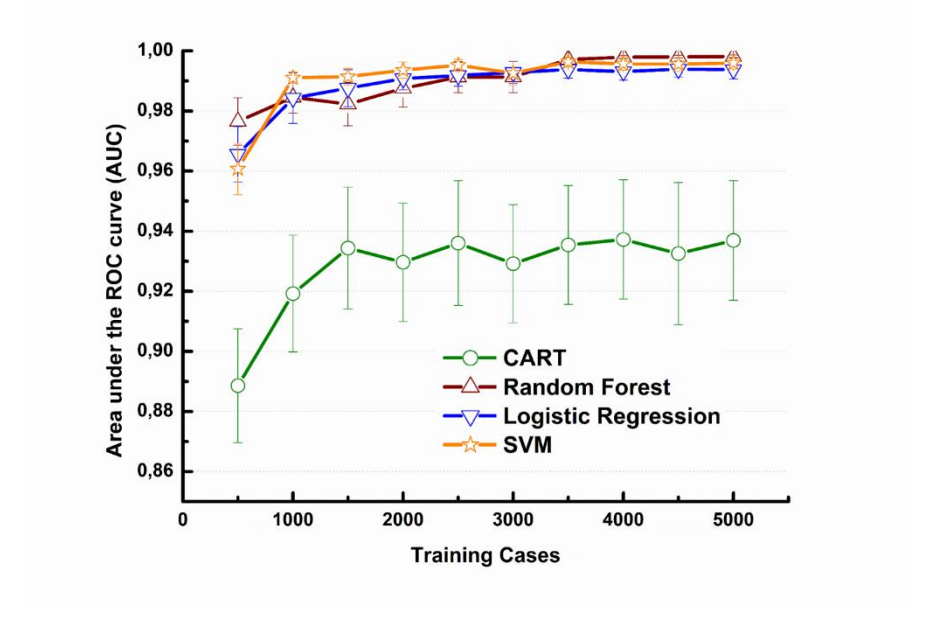


Figure 10. Area under the ROC curve for testing cases for CART, Random Forest, logistic regression and SVM



Finally, the computing time per lightpath is analysed in Figure 11 (taking into account the decision maker and the corresponding machine learning model). As shown in the figure, all models are faster than SVM (up to 1 magnitude order), which involves that they are also faster than the CBR scheme proposed in the reference article. As in previous figures, Random Forests and logistic regression are presenting quite similar results.

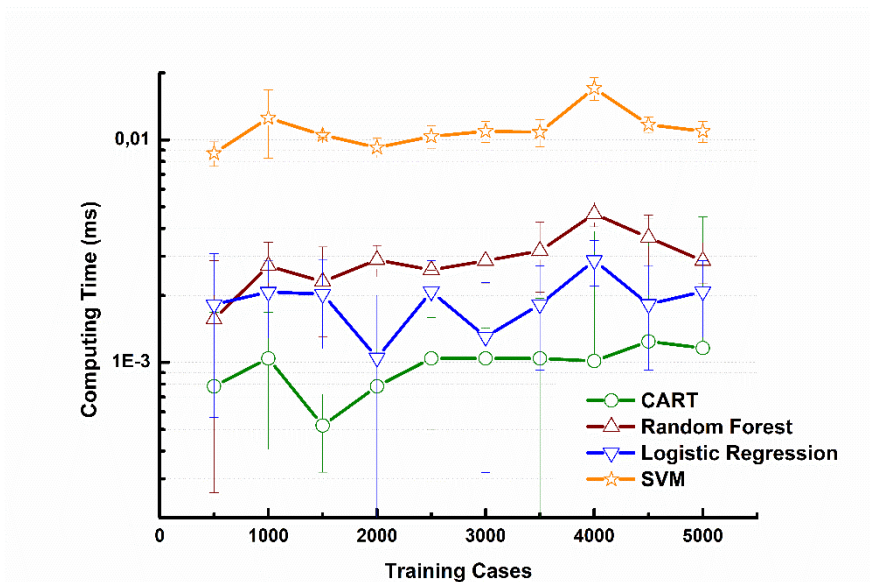


Figure 11. Computing time per lightpath (ms) taking into account the combined classification of the decision maker and the different machine learning models

So far, only CART model can be discarded as an ideal candidate: despite offering the best results in terms of time metrics, it provides significantly lower performance metrics than the rest. Among the other models, metrics proposed are very similar, with a slight advantage for SVM in accuracy and Random Forests in AUC. However, significantly higher training and testing times allow SVM to be discarded from among them. Finally, in order to choose between logistic regression and Random Forests, the best decision seems to opt for the simpler model, which entails shorter time metrics, that is, to opt for logistic regression. Notwithstanding, the following scenario may help to refine the best candidate search.

#### 4.2.1.2. AUC Optimization

In this section, results when the optimization process of the cross-validation is carried out looking for the highest possible value of AUC-CV are exposed. The reason for analysing this metric is justified by the interest that a network operator might have in a system that predicts the lightpath QoS minimizing the number of false positives, even though this could lead to an increase in the number of false negatives. In this way, lightpaths that would result in a Q-factor value below a certain threshold would not be established, reinforcing quality of service (QoS). Therefore, higher values of specificity would be desirables in the design of the proposed machine learning models.

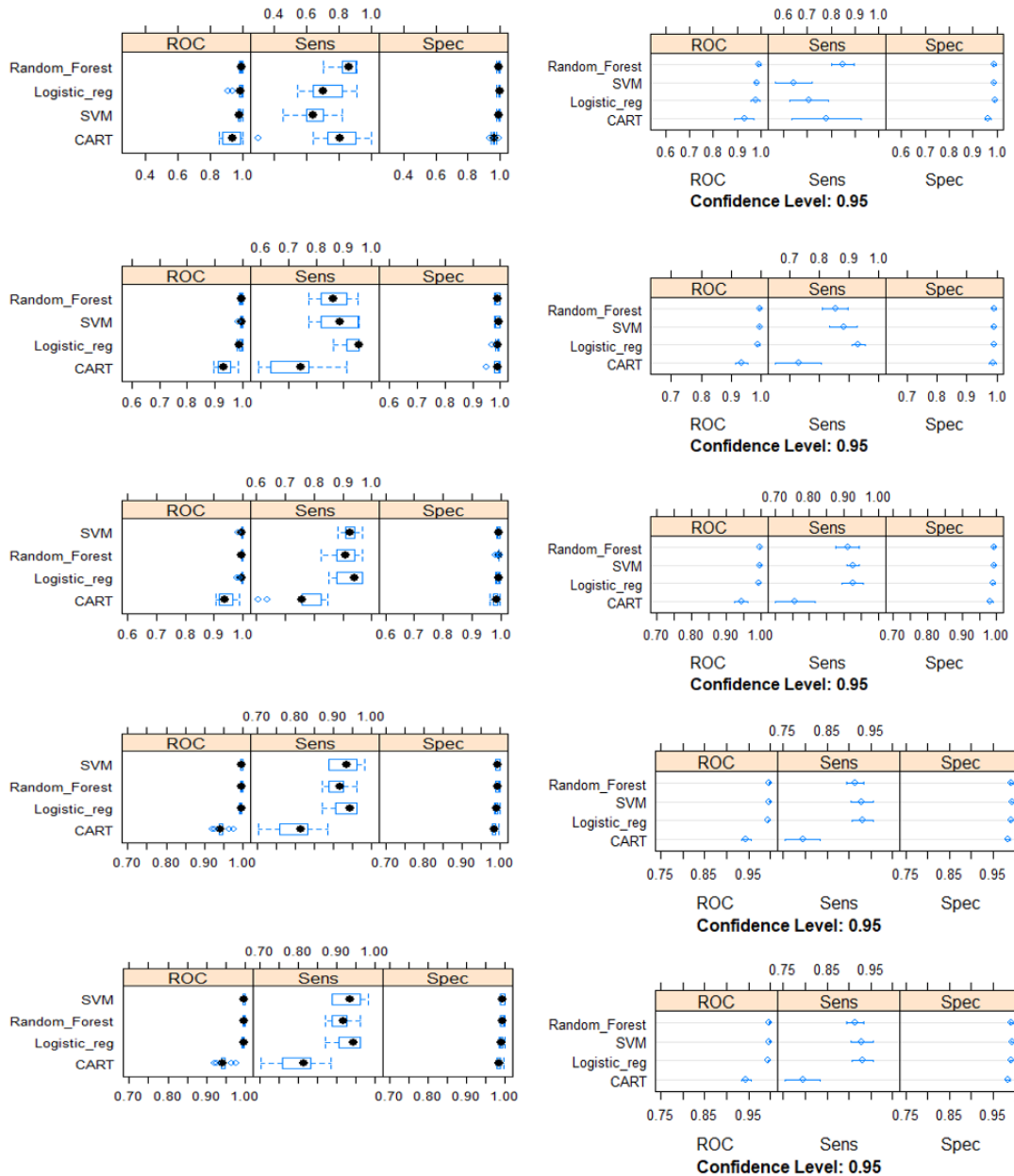


Figure 12. Performance in terms of AUC (named ROC), sensitivity (Sens) and specificity (Spec) of the cross-validation process for different machine learning algorithms

Same considerations as in Figure 6 for Figure 12 regarding rows (i.e., training cases ranging from 1000 to 5000), columns (boxplots vs. confidence intervals) and machine learning algorithms. On this occasion, performance metrics derived from the cross-validation process are AUC (named ROC in the figure), sensitivity (Sens) and specificity (Spec). As mentioned before, a high AUC is desirable and also the specificity is considered an essential metric for model design (i.e., specificity needs to be as close to 1 as possible). As shown in Figure 12, after the cross-validation process with the aim of improving the AUC, all models provide extremely high performance metrics. CART is clearly a step below the rest, but the remaining ones provide extremely compact boxplots and confidence levels, with median and mean values of AUC and Spec extremely close to 1.

Figure 13 shows also the AUC-CV with a finer granularity. Conclusions derived from this figure are the same as in Figure 12, i.e., CART is clearly providing a lower AUC metric than the rest, yet its performance is high enough.

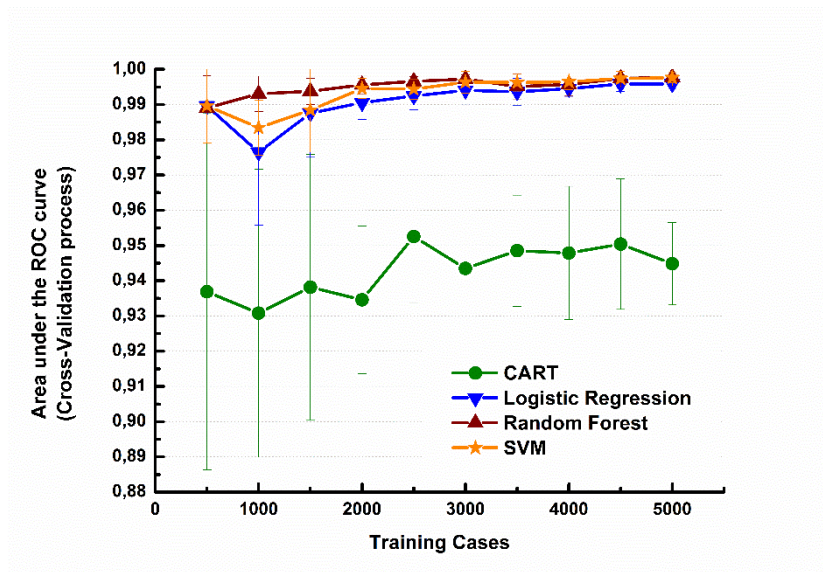


Figure 13. Area under the ROC curve for the cross-validation process for CART, logistic regression, Random Forests and SVM models

With the aim to shed some more light on the search for the ideal candidate, the following figure (Figure 14) proposes the analysis of the training time for all the models under study. Results are, logically, similar to the ones in Figure 8. As in the previous scenario analysis (optimization of the accuracy metric) CART stands out for its speed when it comes to train, up to 4 magnitude orders faster than the slowest, SVM.

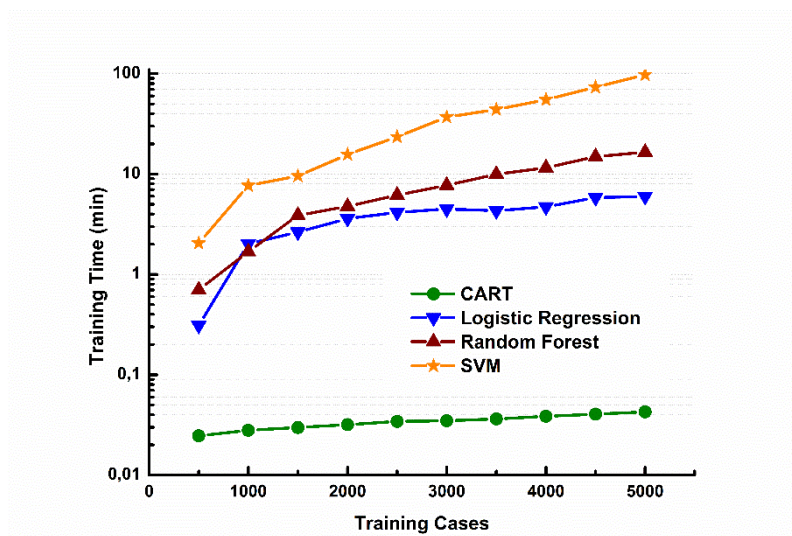


Figure 14. Training time after optimizing AUC for CART, logistic regression, Random Forests and SVM

On the other hand, SVM is more penalized by the size of the training dataset, resulting in greater differences with respect to the others as the number of training cases grows. Summing up, Random Forest and logistic regression trade-offs between time and performance metrics seem optimal.

So far, only the training phase has been taken into account. The following figures show the results after the testing phase.

Figure 15 depicts the accuracy out for all the models. Not many differences are observed when comparing this figure with the corresponding (Figure 9) of the previous scenario. Again, it is worth mentioning that the levels of accuracy achieved are comparable to the ones obtained for the MATLAB scenario for the three best candidates: SVM, Random Forests and logistic regression

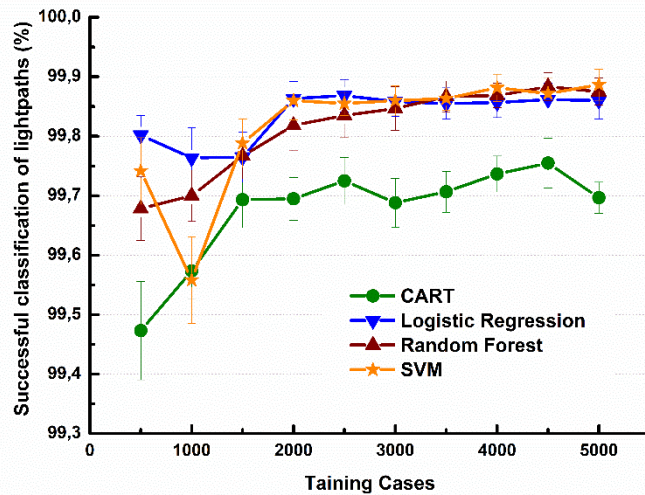


Figure 15. Successful classification of lightpaths (decision maker + machine learning models) for CART, logistic regression, Random Forests and SVM

Figure 16 shows the average AUC together with the 95% confidence intervals for all the models under study. Everyone presents an AUC close to 1, especially for Random Forests, SVM and logistic regression, in that precise order, meaning that they perform classifications very accurately, identifying correctly both categories. Results are very similar to the ones obtained in the previous scenario with one small difference, AUC out, as expected, is slightly higher for SVM than the scenario where the accuracy was optimized.

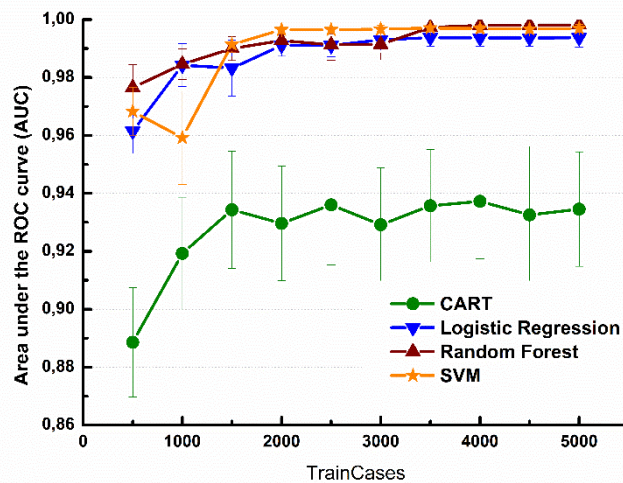


Figure 17. AUC after the testing phase for CART, logistic regression, Random Forests and SVM

Finally, Figure 17 presents the computing time per lightpath during the classification into high and low QoT categories for every model (in combination with the decision maker). Same conclusions can be derived as in the previous scenario, as the computing time depends on the complexity of every model, which may not really differ when optimizing during the cross-validation phase with a performance metric or another. It is worth only remembering that SVM turns up to be the slowest, but at the same time it was faster in MATLAB scenario than every CBR implementation of the reference article.

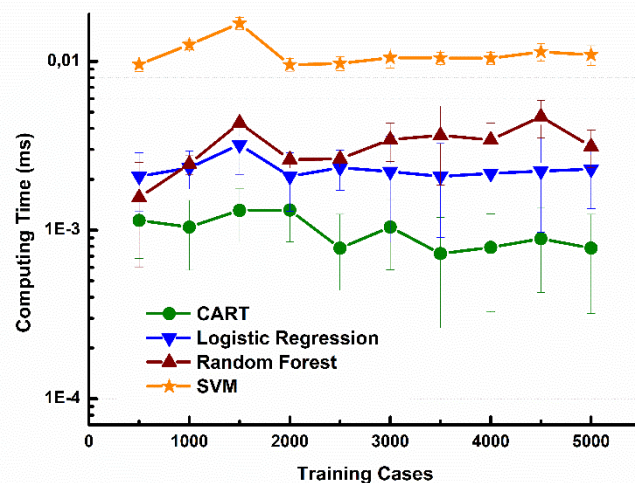


Figure 16. Computing time per lightpath (ms) during the classification for CART, logistic regression, Random Forests and SVM in combination with the decision maker.

As every model turned to be highly accurate in this binary classification model, the analysis of the optimization with different performance metrics (accuracy and AUC) has not yield significant differences. Notwithstanding, the additional analysis of the AUC performance metric allowed to extract important

information of the expected response of the designed models, like the associated specificity.

As stated above, logistic regression and Random Forests are clearly the best candidates. Among them it is still not clear which one to choose, as Random Forests provides slightly higher performance metrics (accuracy and AUC) but logistic regression is faster.

#### 4.2.2. *Removing the Decision Maker*

So far the study has considered an artificial decision maker to classify cases outside an uncertainty area, whose thresholds are chosen by investigating the data priori, with the aim to reduce the overall computing time per lightpath. In particular for DT network, only 8% of the cases are solved whether by the CBR or the machine learning module. The truth is that machine learning algorithms might be used to face the entire problem, avoiding the artificial creation of thresholds at a guess. For this reason, two new scenarios (one optimizing the accuracy performance and another one optimizing the AUC) are proposed removing the decision maker (i.e., the artificial thresholds). For this reason, the training dataset will contain lightpaths of any length.

For the sake of a fair comparison with the previous  $R$  scenarios, the training datasets contain the same cases belonging to the uncertainty area as before and new cases outside this area are added in such a number that the proportion of lightpaths inside and outside the uncertainty area in the whole database is kept (i.e., the number of lightpaths in the training dataset shared with the previous scenarios will represent only 8% of the total training dataset, while the remaining 92% will be randomly selected among the lightpaths outside the uncertainty area). In addition, test datasets are the same as the ones used in the previous  $R$  scenarios.

It is worth mentioning, as stated in the methodology section, that once SVM was proved to outperform the CBR systems proposed in the reference article in the MATLAB scenario and later was also proved itself to be outperformed by Random Forests and logistic regression in the  $R$  scenarios (above all due to time metrics), it is discarded for the remaining of the study, mainly because of its extensive training time.

##### 4.2.2.1. *Accuracy Optimization*

As stated above, in this scenario the models are trained with the aim to optimize the accuracy performance. Figure 18 shows the results of this optimization of the accuracy performance after the cross-validation process.

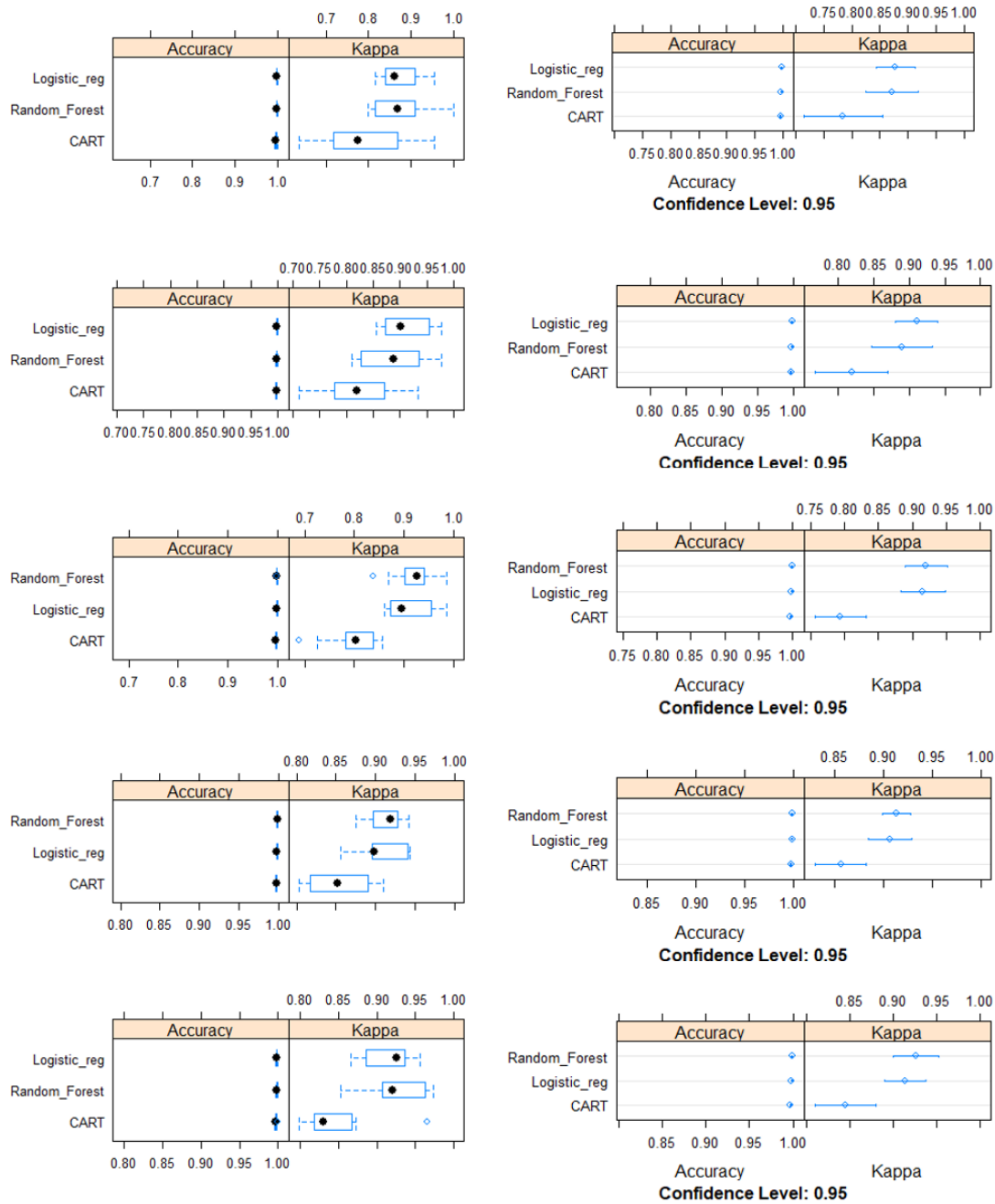


Figure 18. Performance metrics (Accuracy and Kappa) after the cross-validation process for logistic regression, Random Forests and CART

Same considerations as in Figure 6 for Figure 18 regarding rows (i.e., training cases, in this case ranging from 12500 to 62500), columns (boxplots vs. confidence intervals) and machine learning algorithms (once SVM is discarded). In this case, the accuracy CV is extremely closer to 100% for every model than as presented in Figure 6, due to the fact that a higher number of training cases are introduced to replace the decision maker. In terms of accuracy the three models are almost indistinguishable. On the other hand, kappa allows to categorize CART as the one expected to provide with the worst behaviour among the three of them, while

Random Forests may be expected to provide with a slightly higher performance than logistic regression.

Figure 19, which presents the accuracy CV for the three models, helps to visualize these facts. While Figure 7 showed an accuracy CV below 97% for CART and around 98.5% for logistic regression and Random Forests, Figure 19, depicts an increase of this metric up to 99.75% for CART and up to around 99.85% for the remaining models, with Random Forest slightly outperforming logistic regression.

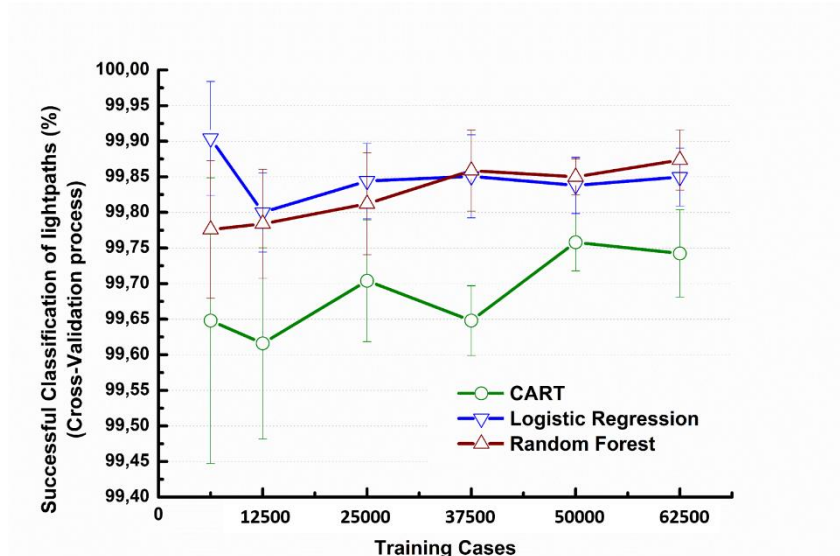


Figure 19. Successful classification of lightpaths during the cross-validation process for CART, logistic regression and Random Forests

As shown in Figure 20, training times are dramatically increased with respect to the training times of the previous scenarios. This result is expected since, as stated before, this metric grows exponentially with the increase of the number of training cases and now training cases are increased by a 1250%! CART is still extraordinary

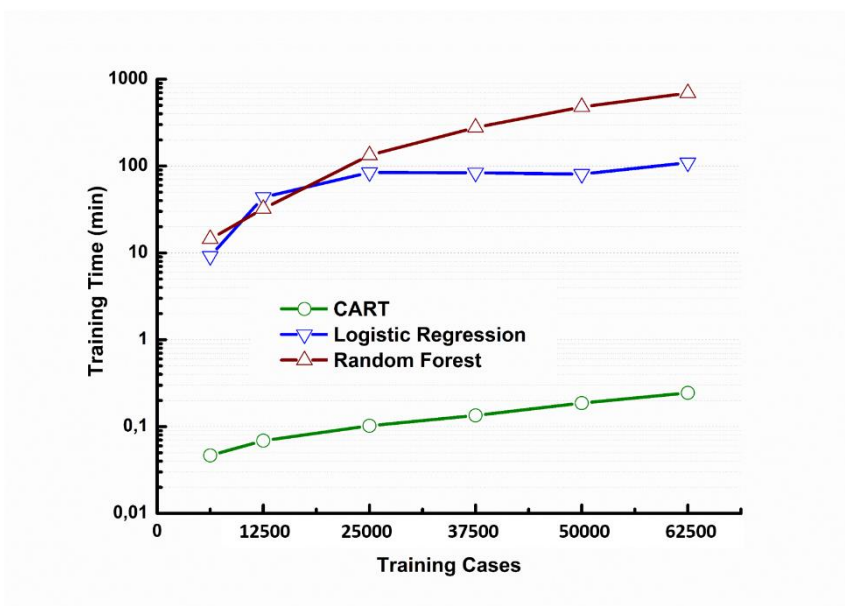


Figure 20. Training time for CART, logistic regression and Random Forests



fast, while Random Forests and logistic regression remain comparable up to 25000 training cases. For greater training data sets the difference in training time rises up to almost one magnitude order slower for Random Forests.

Figure 21 presents the accuracy out (as a reminder, now all instances within the testing datasets are classified by the machine learning models regardless the total length of the lightpaths) for CART, logistic regression and Random Forests after the classification is carried out. Results are perfectly comparable with the ones obtained with the hybrid system (decision maker + machine learning models) in Figure 9. Random Forests maintains its accuracy level in 99.87%, logistic regression accuracy is slightly decreased from 99.86% to 99.82% in this new scenario and CART maintains also its accuracy level in 99.73%.

In the absence of analysing the AUC and computing time per lightpath, it seems that Random Forests and logistic regression are valuable alternatives to the hybrid systems seen so far.

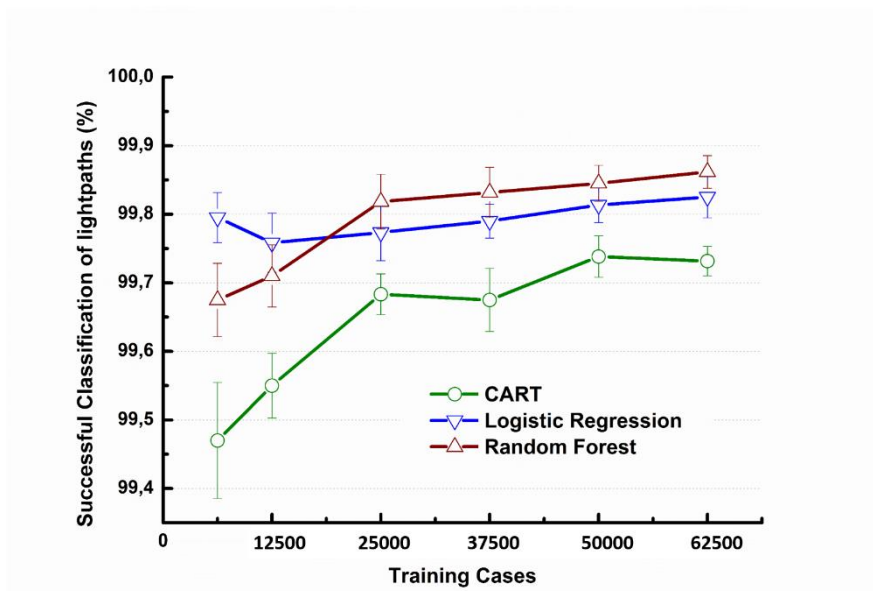


Figure 21. Successful classification of lightpaths after the testing phase for CART, logistic regression and Random Forests

Figure 22 depicts the AUC out for the three models under study. It seems that a saturation point is reached with a training dataset size of 37500, At this point, Random Forests and logistic regression provides an almost perfect trade-offs between sensitivity and specificity (AUC extremely close to 1), up to 0.9998 and 0.9977 for Random Forests and logistic regression, respectively. CART, although providing very high performance levels is clearly outperformed by the others.

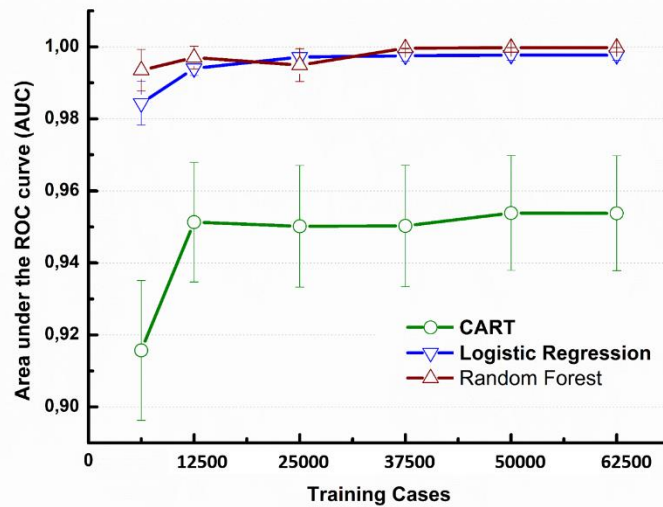


Figure 22. Area under the ROC curve after the testing phase for CART, logistic regression and Random Forests

Finally, Figure 23 allows to analyse the computing time per lightpath during the classification. As stated along this section, in this case, this computing time refers only to the classification performed by the machine learning models themselves since the decision maker is removed. There are some interesting conclusions that can be extracted comparing this figure with the corresponding one in the hybrid scenario in *R*, Figure 11.

For the convenience of the reader, results shown in Figure 11 for 5000 training cases, equivalent to 62500 cases in this new scenario (5000 cases inside the so-called uncertainty area and 57500 outside it), were 0.01, 0.003, 0.002, 0.001 milliseconds for SVM, Random Forests, logistic regression and CART, respectively. As shown in Figure 23, computing time per lightpath has been increased for every model up to 0.025, 0.017, and 0.004 milliseconds for Random Forests, logistic regression, and CART, respectively. This increase was expected since the simple decision maker was extremely fast and it carried out 92% of the classifications, but Random Forests and logistic regression computing time per lightpath values are still comparable to the computing time per lightpath for SVM in the hybrid one (2.5 and 1.7 times slower, respectively). Moreover, their accuracy performance is similar to the one offered by SVM (99.89%) in that scenario (Figure 11), since Random Forests and logistic regression achieved a successful classification of lightpaths of 99.87% and 99.82% in Figure 21.

Summing up, SVM in the MATLAB hybrid system proved to be up to 40 times faster than R-CBR and up to 5 times faster than FixE-CBR, all of it with a higher percentage of successful classification of lightpaths. Random Forests and logistic regression achieved comparable percentages of successful classification of lightpaths in the scenario where the decision maker was removed with respect to SVM in the *R* hybrid scenario and also comparable computing times per lightpath. In

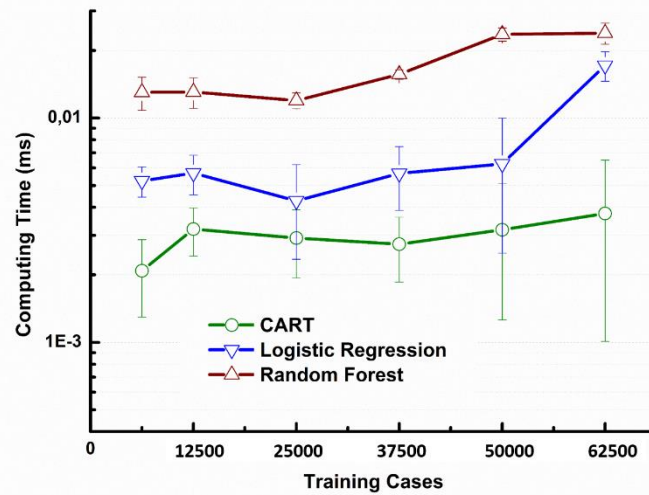


Figure 23. Computing time per lighpath during the classification for CART, logistic regression and Random Forests

conclusion it can be stated that Random Forests and logistic regression can assume the binary classification problem proposed in the article of reference by themselves (i.e., without the decision maker) outperforming the accuracy metrics and in a shorter computing time per lighpath.

#### 4.2.2.2. AUC Optimization

This scenario, as stated before, comprises the same study as the previous one (removing the decision maker) but carrying out an optimization of the AUC during the cross-validation process. Results are quite similar to the scenario that optimizes the accuracy. For this reason and with the intention of not extending this section much more, the main differences will be briefly discussed.

As with the rest of *R* scenarios, Figure 24 depicts the performance metrics derived from the optimization of the AUC: ROC (which corresponds with the AUC-CV), Sens (sensitivity) and Spec (specificity). In terms of specificity the models are indistinguishable, offering an excellent performance. The AUC is also extremely close to the ideal value of 1 for Random Forests and logistic regression, slightly higher for the former as it can be better visualized in Figure 25. CART is again clearly below the level of the other models, yet providing also a significantly good performance.

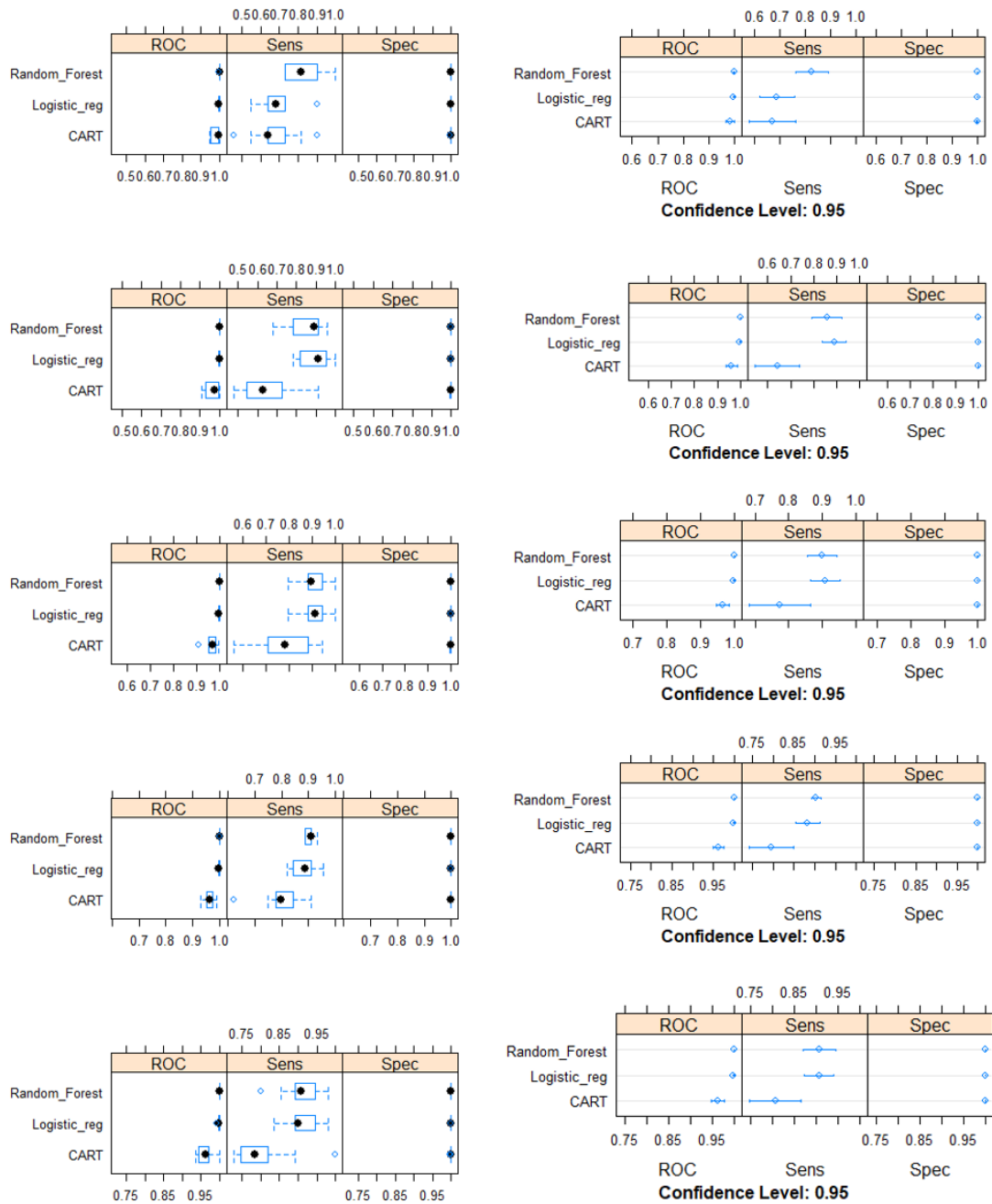


Figure 24. Performance of AUC (ROC), sensitivity (Sens) and specificity (Spec) for the cross-validation process

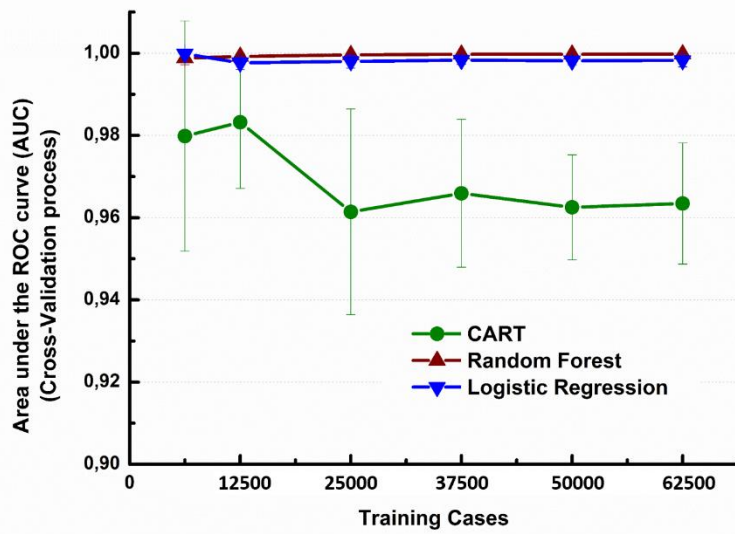


Figure 25. Area under the ROC curve derived from the cross-validation process for CART, Random Forests and logistic regression

As expected, training times represented in Figure 26, are basically the same analysed previously in Figure 20, with CART turning to be extremely fast compared to the others and Random Forests increasing its training time faster than logistic regression as the number of training cases rise up.

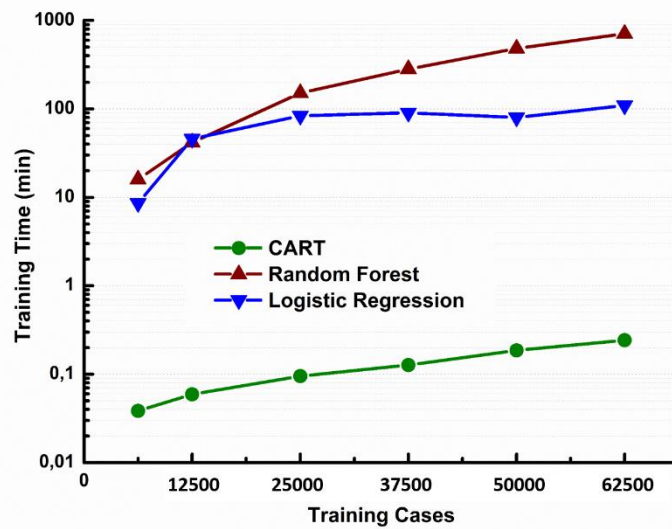


Figure 26. Training time for CART, Random Forests and logistic regression

Figure 27 shows the accuracy out for this scenario. Again, differences with the corresponding figure (Figure 21) are insignificant, with a minimum reduction (0.04 percentage points in worst case) of the accuracy out, as the optimization is carried out this time for AUC, instead of for accuracy.

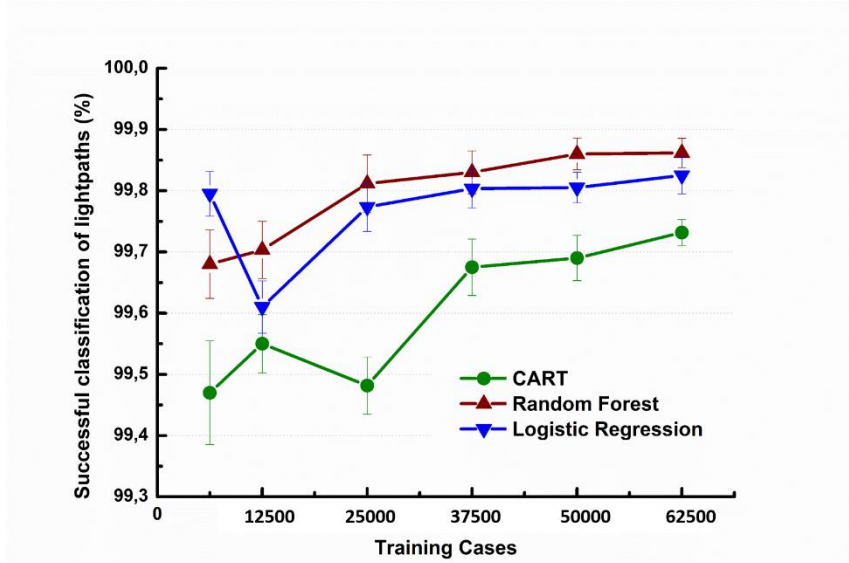


Figure 27. Percentage of successful classification of lightpaths after the testing phase for CART, Random Forests and logistic regression

As shown in Figure 28, the AUC out (after the testing phase) is again extremely similar to the corresponding figure (Figure 22). The only difference is that saturation points are reached sooner. As an example, Random Forests AUC out is always above 0.9978. This small difference is in line with the fact that the cross-validation stage optimized the AUC

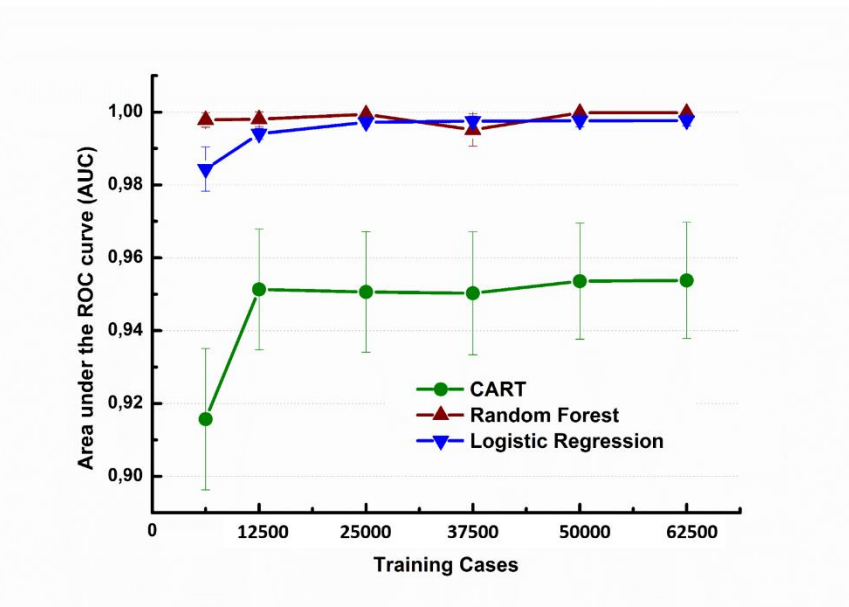


Figure 28. Area under the ROC curve after the testing phase for CART, Random Forests and logistic regression

Finally, Figure 29 depicts the computing time per lightpath for this scenario. This metric is again around the same levels as in the corresponding figure of the previous scenario (Figure 23)

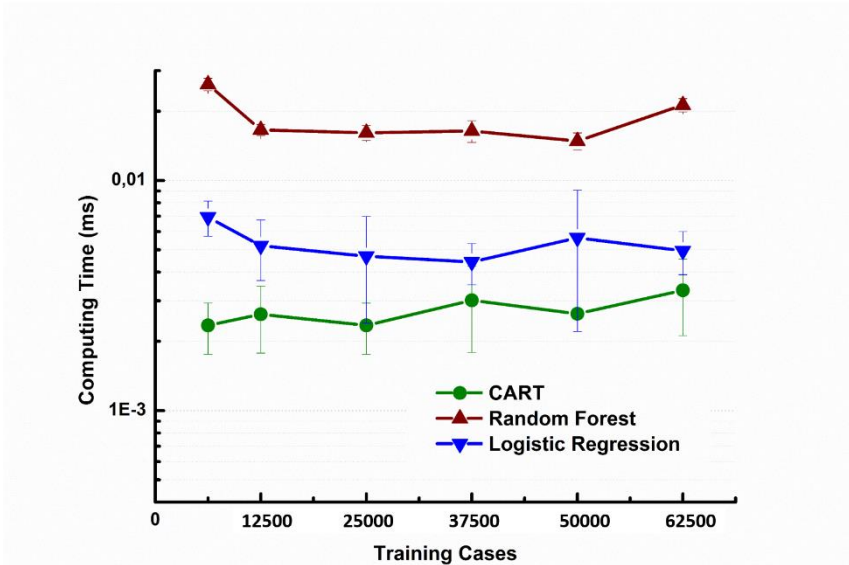


Figure 29. Computing time per lightpath during the classification phase for CART, Random Forests and logistic regression

#### 4.2.2.3. Selection of the Ideal Candidate

Machine learning algorithms studied in this thesis have proven to be very efficient in solving the classification problem proposed in the reference article. So far, SVM has been discarded as the ideal candidate due to its necessary extremely high training time, while CART has been discarded because its performance metrics are below the desired levels.

Moreover, the models proposed are able to classify lightpaths regardless whether they belong or not to the artificial uncertainty area, avoiding the need for a decision maker and the artificial creation of thresholds at a guess. In this ideal scenario, CART may provide with an interesting representation of how the learning process is performed. Figure 30 shows a chart representing schematically the drawing of the CART tree after the training phase.

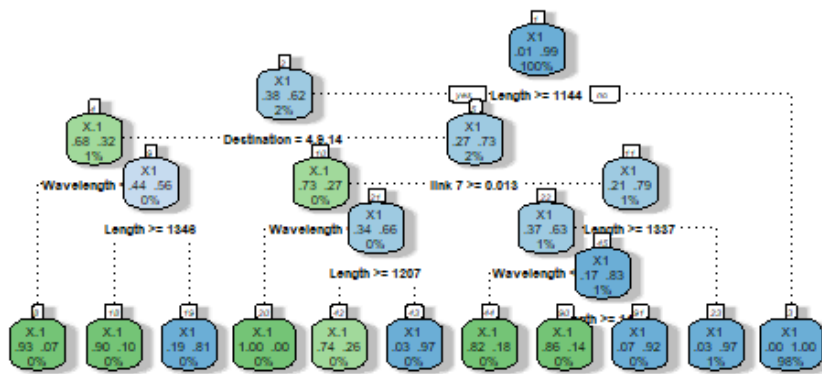


Figure 30. Drawing of the CART tree after the training process for the scenario without decision maker and optimization of the AUC

Before analysing the drawing of the tree, it is worth mentioning that the CART proposed neither performs centering and scaling of the data nor dummy variables transformation in order to make the tree more easily interpretable.

X1 represents high QoT category, while X.1 represents low QoT. A node is labelled with the majority class that is introduced in it. Below the label of each node it can be found the proportion of cases of low QoT (to the left) and of upper QoT (to right-handed) with respect to the total of cases introduced in that node. Finally, the last number of the node represents the percentage of cases over the total that enter that node. After each node there is a bifurcation (except in the lowest level). If the condition is fulfilled the process takes the left path, if it is not fulfilled it takes the one on the right.

So far, Random Forests and logistic regression have proven to be the best candidates but it is hard to say which one should be chosen. There is another degree of design that can be applied at this point and may help to make this decision. As in the classification process the probability of belonging to one or another class is calculated for each instance, the decision threshold can be modified to favour the sensitivity or specificity of the model as a function of the requirements. In our case, it can be reasoned that a network operator would be more interested in avoiding false positives that could affect QoS, becoming the existence of false negatives much less critical. Therefore, specificity becomes clearly the most significant metric.

Taking this into account, the study carried out in the last scenario (the decision maker removed and optimization of the AUC) is taken as the starting point of the final decision. The models created in that scenario for Random Forests and logistic regression for 62500 training cases are tested with the 10 datasets at the same time (i.e., testing one time with 60000 test cases, instead of testing with 6000 cases and repeating the process 10 times as in previous sub-sections). Figure 31 and Figure 32 show the ROC curves after this classification process is carried out, for logistic regression and Random Forests, respectively.

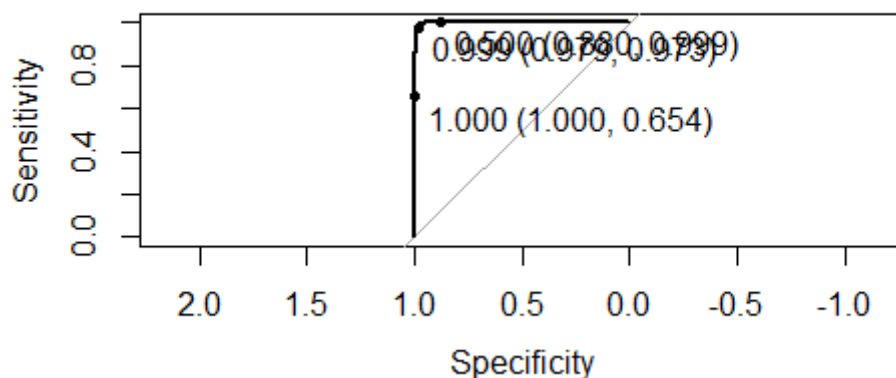


Figure 31. ROC curve for logistic regression



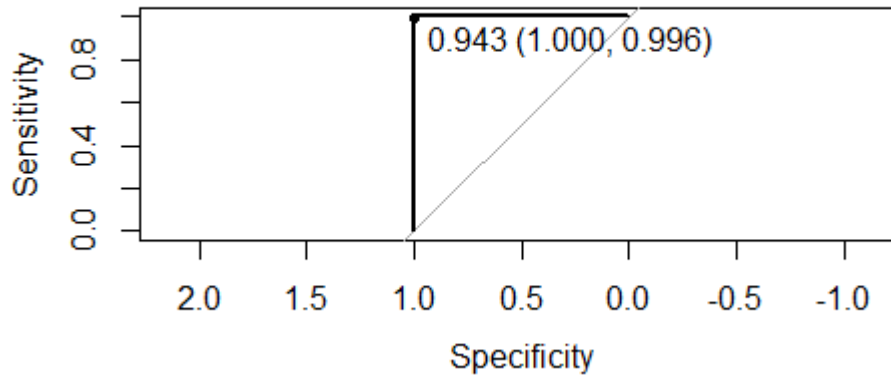


Figure 32. ROC curve for Random Forests

Some thresholds are proposed in both figures so as to increase the specificity (this process penalizes the sensitivity). The threshold is shown at first place, while the specificity and sensitivity are the values inside parenthesis, in that order.

From Figure 31, it is easy to conclude that increasing the threshold entails a penalization in the sensitivity metric. As an example, a threshold of 0.999 allow to increase specificity up to 0.979 and sensitivity would be reduced up to 0.973. If the requirements were even more restrictive, a threshold of 0.9999999985 (shown as 1.000 in the figure) would be necessary to reduce the number of false positives to 0 (i.e., increase specificity to 1), but this threshold would entail a reduction of the sensitivity up to 0.654 and, as a result of it, accuracy would be dropped to an undesirable 0.669.

Random Forests proves to be easier to optimize in this last design. Figure 32 shows a threshold of 0.943, that involves a perfect specificity of 1 with a sensitivity of 0.996. As a result, accuracy out is reduced only from 99.86% to 99.56%. For this reason, Random Forests is finally chosen as the ideal candidate for this classification problem.

## 5. Conclusions

The starting point of this thesis was the work presented in [35], which described and demonstrated the benefits of the employment of a cognitive QoT estimator of a hybrid nature in WRON networks to classify lightpaths into high and low QoT categories. This estimator basically consisted of two elements: a decision maker that only took into account the length of the lightpath and a Case-Based Reasoning (CBR) module that classified a given element according to the most similar case stored in its Knowledge Base (KB). The former was in charge of classifying all those lightpaths whose length was above or below the limits that identified a given uncertainty area in which, in principle, the QoT could not be firmly predicted, while the latter tried to classify indeed those inside the uncertainty area

A machine learning approach was proposed to solve this binary classification problem the 14-node Deutsch Telekom (DT) network. With this aim, a SVM module was proposed to replace the CBR trying to classify lightpaths inside the uncertainty area. SVM proved to be a more efficient solution to this problem in a scenario recreating the exact conditions in which the reference study was performed. As a matter of fact, SVM achieved, in combination with the decision maker, a percentage of successful classifications up to 99.91% with 5000 training cases, while R-CBR and the optimized CBR achieved up to a 99.8% and 99.89% in the same conditions. In addition, SVM proved to be faster than the proposed system in the reference article, fact of significant relevance since a low computing time per lightpath is a critical requirement, especially in highly dynamic networks. SVM computed lightpaths around 40 times faster than R-CBR and 5 times faster than FixE-CBR in the optimal 5000 training cases situation. Finally, in the scenario in which all lightpaths that were used to train and to test the different proposals belonged to the uncertainty area, SVM showed again a higher percentage of successful classifications of lightpaths, not only for DT network, but also for the ultra-long haul GÉANT network, achieving a 98.73% and 91.79% of accuracy, respectively, vs a 97.5% and 90.56% achieved by the R-CBR. Notwithstanding, the time that took to train the SVM model was too high and grew exponentially as the number of training cases was increased. As an example, for 5000 cases for DT network, it took more than 100 minutes to complete the training. For that reason, alternative machine learning algorithms were investigated

Once SVM proved to outperform both CBR configurations proposed in the reference article, the study was moved to a different programming environment (R), “friendlier” in terms of machine learning techniques comparisons (using the caret package). There, the investigation performed in the reference article was recreated again with new machine learning algorithms replacing the CBR module. Specifically, besides SVM, Random Forests, logistic regression and CART were proposed as alternative candidates. In addition, the models were optimized during the training

phase looking for the best performance in terms of two different metrics: accuracy and Area Under the ROC Curve (AUC). Although the investigation derived from the utilization of one metric or the other did not yield any interesting conclusion since results were virtually identical, it allowed to deeply characterize and understand the trained models. However, the comparison between models did yield significant conclusions. CART proved to be the model providing with the lowest performance for the best case (5000 training cases) in terms of accuracy (99.72%) and AUC (0.9369). On the other hand, logistic regression (accuracy of 99.86% and AUC of 0.9937), Random Forests (accuracy of 99.87% and AUC of 0.998) and SVM (accuracy of 99.885% and AUC of 0.9959) showed very close performance metrics.

In order to select the best candidate, computing time per lightpath was also taken into account. CART turned to be the fastest (0.00078 ms), while Random Forests (0.0028 ms) and logistic regression (0.0017 ms) showed a very similar behaviour. On the other hand, SVM was clearly the slowest with a computing time per lightpath of 0.01 ms. At this point, it was easy to conclude that Random Forests and logistic regression were even better candidates than SVM to replace the CBR module, moreover if the training time was taken into account as SVM was approximately 10 times slower than Random Forests and logistic regression.

Another step beyond, the artificial decision maker using length thresholds imposed at a guess was removed and, a new scenario in which the machine learning algorithms were in charge of classifying the lightpaths by themselves, was proposed. SVM had been already discarded before, the other three were investigated in this new scenario.

As it happened before, the comparison of the models optimized by the accuracy or by the AUC in this new scenario was little productive in terms of conclusions on differences in benefits. However, the comparison between the models themselves yield interesting conclusions.

In terms of performance metrics, the accuracy was maintained in this scenario with respect the previous one and the AUC was even increased. As a matter of fact, CART provided, in the best case (62500 training cases), with an accuracy of 99.73% and AUC of 0.954, logistic regression showed an accuracy of 99.82% and AUC of 0.997 and Random Forests achieved a 99.86% of accuracy and 0.9998% of AUC.

In terms of time metrics, training time was dramatically increased for all models since the number of training cases was also increased by a 1250% in order to cope with enough cases outside the uncertainty area. In this situation, CART continued to be extraordinary faster than the rest. At the same time logistic regression could be trained, in the most critical case, around 8 times faster than Random Forests. Regarding computing time per lightpath, differences were less

significant, CART consumed 0.0037 ms, while logistic regression and Random Forests consumed 0.017 and 0.024 ms, respectively.

From the results obtained in this new scenario, it was concluded that Random Forests and logistic regression could be perfectly considered as alternative candidates to replace the combination of the decision maker and the CBR modules proposed in the reference article. This statement is supported by the fact that these models obtained comparable accuracy values and a closer to 1 area under the ROC curve with respect to SVM in combination with the decision maker, all of it with comparable computing times per lightpath.

The last remaining question was to decide between Random Forests and logistic regression, since the former offered better AUC and accuracy values, while the latter offered better test and training times. For these reasons, a new degree of freedom was used when designing these types of models. Such optimization was directly related to the interest that a network operator might have in using models that had a high specificity, thus reducing the number of false positives that could have a negative impact on QoS.

By studying the ROC curve for logistic regression and Random Forests and proposing new decision thresholds, it was found that Random Forests could easily be optimized to provide a perfect specificity, reducing the number of false positives to 0 and without unduly compromising the final accuracy, which was slightly reduced from 99.86% to 99.56%. Since the logistic regression demanded much higher thresholds to achieve the optimum specificity, which in turn compromised in a more than considerable amount the final success, it was concluded that the best candidate to solve the problem raised was Random Forests.

As future work it is proposed the extension of this analysis to ultra-long haul networks, like GÉANT2, in order to identify possible scalability issues. In addition, a new scenario in which the Q-factor may be completely estimated by machine learning algorithms, instead of classifying the quality of transmission into two categories, is proposed. For this matter, linear regression and neural networks may be an interesting starting point.

## 6. References

- [1] T. Duthel, G. Clarici, C. R. S. Fludger, J. C. Geyer, C. Schulien and S. Wiese, "Laser Linewidth Estimation by Means of Coherent Detection," in *IEEE Photonics Technology Letters*, vol. 21, no. 20, pp. 1568-1570, Oct.15, 2009.
- [2] K. Kikuchi, "Characterization of semiconductor-laser phase noise and estimation of bit-error rate performance with low-speed offline digital coherent receivers," *Opt. Express* 20, 5291-5302 (2012).
- [3] D. Zibar et al., "Application of Machine Learning Techniques for Amplitude and Phase Noise Characterization", *J. Lightwave Technol.*, vol. 33, pp. 1333, 2015.
- [4] Y. Huang *et al.*, "A Machine Learning Approach for Dynamic Optical Channel Add/Drop Strategies that Minimize EDFA Power Excursions," *ECOC 2016; 42nd European Conference on Optical Communication*, Dusseldorf, Germany, 2016, pp. 1-3.
- [5] A. S. Ahsan, C. Browning, M. S. Wang, K. Bergman, D. C. Kilper and L. P. Barry, "Excursion-free dynamic wavelength switching in amplified optical networks," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 9, pp. 898-905, Sept. 2015.
- [6] D. A. Mongardien, S. Borne, C. Martinelli, C. Simonneau and D. Bayart, "Managing Channels Add/Drop in Flexible Networks Based on Hybrid Raman/Erbium Amplified Spans," *2006 European Conference on Optical Communications*, Cannes, France, 2006, pp. 1-2.
- [7] Y. Huang, P. B. Cho, P. Samadi and K. Bergman, "Dynamic power pre-adjustments with machine learning that mitigate EDFA excursions during defragmentation," *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, Los Angeles, CA, 2017, pp. 1-3.
- [8] E. d. A. Barboza, C. J. A. Bastos-Filho, J. F. Martins-Filho, U. C. de Moura and J. R. F. de Oliveira, "Self-adaptive erbium-doped fiber amplifiers using machine learning," *2013 SBMO/IEEE MTT-S International Microwave & Optoelectronics Conference (IMOC)*, Rio de Janeiro, 2013, pp. 1-5.
- [9] J.R.O. et al., "Demonstration of edfa cognitive gain control via gmpls for mixed modulation formats in heterogeneous optical networks, " in *Optical Fiber Communication Conference*. Optical Society of America, 2013.
- [10] Z. Dong, F. N. Khan, Q. Sui, K. Zhong, C. Lu and A. P. T. Lau, "Optical Performance Monitoring: A Review of Current and Future Technologies," in *Journal of Lightwave Technology*, vol. 34, no. 2, pp. 525-543, Jan.15, 15 2016.
- [11] X. Wu, J. A. Jargon, R. A. Skoog, L. Paraschis and A. E. Willner, "Applications of Artificial Neural Networks in Optical Performance Monitoring," in *Journal of Lightwave Technology*, vol. 27, no. 16, pp. 3580-3589, Aug.15, 2009.
- [12] Z. Li, Z. Jian, L. Cheng, Y. Yang, C. Lu, A. P. T. Lau, C. Yu, H. Y. Tam, and P. K. A. Wai, "Signed chromatic dispersion monitoring of 100 Gbit/s CS-RZ DQPSK signal by evaluating the asymmetry ratio of delay tap sampling," *Opt. Express*, vol. 18, no. 3, pp. 3149-3157, Feb. 2010.

- [13] F. N. Khan, A. P. T. Lau, Z. Li, C. Lu, and P. K. A. Wai, "OSNR monitoring for RZ-DQPSK systems using half-symbol delay tap sampling technique," *IEEE Photon. Technol. Lett.*, vol. 22, no. 11, pp. 823–825, June 2010.
- [14] M. C. Tan, F. N. Khan, W. H. Al-Arashi, Y. Zhou and A. P. T. Lau, "Simultaneous optical performance monitoring and modulation format/bit-rate identification using principal component analysis," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 6, no. 5, pp. 441-448, May 2014.
- [15] T. Tanimura, T. Hoshida, J. C. Rasmussen, M. Suzuki and H. Morikawa, "OSNR monitoring by deep neural networks trained with asynchronously sampled data," *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, Niigata, Japan, 2016, pp. 1-3.
- [16] J. Thrane, J. Wass, M. Piels, J. C. M. Diniz, R. Jones and D. Zibar, "Machine Learning Techniques for Optical Performance Monitoring From Directly Detected PDM-QAM Signals," in *Journal of Lightwave Technology*, vol. 35, no. 4, pp. 868-875, Feb.15, 15 2017.
- [17] A. P. T. Lau and J. M. Kahn, "Signal Design and Detection in Presence of Nonlinear Phase Noise," in *Journal of Lightwave Technology*, vol. 25, no. 10, pp. 3008-3016, Oct. 2007.
- [18] A. Napoli *et al.*, "Reduced Complexity Digital Back-Propagation Methods for Optical Communication Systems," in *Journal of Lightwave Technology*, vol. 32, no. 7, pp. 1351-1362, April 1, 2014.
- [19] D. Zibar, M. Piels, R. Jones and C. G. Schaeffer, "Machine learning techniques in optical communication," *2015 European Conference on Optical Communication (ECOC)*, Valencia, 2015, pp. 1-3.
- [20] D. Wang *et al.*, "Nonlinear decision boundary created by a machine learning-based classifier to mitigate nonlinear phase noise," *2015 European Conference on Optical Communication (ECOC)*, Valencia, 2015, pp. 1-3.
- [21] Danshi Wang *et al.*, "KNN-based detector for coherent optical systems in presence of nonlinear phase noise," *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, Niigata, Japan, 2016, pp. 1-3.
- [22] J. J. G. Torres, A. Chiuchiarelli, V. A. Thomas, S. E. Ralph, A. M. C. Soto and N. G. González, "Adaptive nonsymmetrical demodulation based on machine learning to mitigate time-varying impairments," *2016 IEEE Avionics and Vehicle Fiber-Optics and Photonics Conference (AVFOP)*, Long Beach, CA, 2016, pp. 289-290.
- [23] E. Giacomidis *et al.*, "Nonlinear inter-subcarrier intermixing reduction in coherent optical OFDM using fast machine learning equalization," *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, Los Angeles, CA, 2017, pp. 1-3.
- [24] E. Giacomidis *et al.*, "Nonlinear Blind Equalization for 16-QAM Coherent Optical OFDM using Support Vector Machines," *ECOC 2016; 42nd European Conference on Optical Communication*, Dusseldorf, Germany, 2016, pp. 1-3.
- [25] I. de Miguel *et al.*, "Cognitive dynamic optical networks [invited]," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A107-A118, Oct. 2013.
- [26] N. Fernández *et al.*, "Survivable and impairment-aware virtual topologies for reconfigurable optical networks: A cognitive approach," *2012 IV International*

*Congress on Ultra Modern Telecommunications and Control Systems*, St. Petersburg, 2012, pp. 793-799.

- [27] N. Fernández *et al.*, "Techno-economic advantages of cognitive virtual topology design," *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, London, 2013, pp. 1-3.
- [28] N. Fernández *et al.*, "Virtual topology reconfiguration in optical networks by means of cognition: Evaluation and experimental validation [invited]," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A162-A173, Jan. 2015.
- [29] M. Ruiz *et al.*, "Service-triggered Failure Identification/Localization Through Monitoring of Multiple Parameters," *ECOC 2016; 42nd European Conference on Optical Communication*, Dusseldorf, Germany, 2016, pp. 1-3.
- [30] F. Morales, M. Ruiz, L. Gifre, L. M. Contreras, V. Lopez and L. Velasco, "Virtual network topology adaptability based on data analytics for traffic prediction," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. A35-A45, Jan. 2017.
- [31] S. Azodolmolky *et al.*, "Experimental Demonstration of an Impairment Aware Network Planning and Operation Tool for Transparent/Translucent Optical Networks," in *Journal of Lightwave Technology*, vol. 29, no. 4, pp. 439-448, Feb.15, 2011.
- [32] T. Jiménez *et al.*, "A cognitive system for fast Quality of Transmission estimation in core optical networks," *OFC/NFOEC*, Los Angeles, CA, 2012, pp. 1-3.
- [33] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" *Artificial Intelligence Communications* 7, 1, 39-59 (1994).
- [34] A. Caballero *et al.*, "Experimental demonstration of a cognitive quality of transmission estimator for optical communication systems," *2012 38th European Conference and Exhibition on Optical Communications*, Amsterdam, 2012, pp. 1-3.
- [35] T. Jimenez *et al.*, "A Cognitive Quality of Transmission Estimator for Core Optical Networks," in *Journal of Lightwave Technology*, vol. 31, no. 6, pp. 942-951, March15, 2013.
- [36] L. Barletta, A. Giusti, C. Rottondi and M. Tornatore, "QoT estimation for unestablished lighpaths using machine learning," *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, Los Angeles, CA, 2017, pp. 1-3.
- [37] The caret package, Sep. 2017 [Online]. Available: <http://topepo.github.io/caret/index.html>.
- [38] CHRON Project Deliverable D2.1, Specification of the Network Scenario and Service Requirements Nov. 2011 [Online]. Available: <http://www.ict-chron.eu>.

# A. Resumen en Español

El presente TFM realiza una revisión de la aplicación de técnicas de aprendizaje automático en los sistemas y redes de comunicaciones ópticas. Además, estudia y compara las características de diversos métodos de aprendizaje automático, tales como: máquinas de vectores soporte, regresión logística, árboles de clasificación y regresión (CART) y bosques aleatorios, para predecir la calidad de la transmisión al emplear circuitos ópticos en redes de comunicaciones ópticas con encaminamiento por longitud de onda. Los modelos desarrollados en el TFM obtienen mejores prestaciones que propuestas anteriores, fundamentalmente en términos de tiempo de cálculo, posibilitando su utilización en modo on-line incluso en redes altamente dinámicas, amén de ser más sencillos.

El punto de partida es el artículo [35] en el que se plantea un estimador cognitivo para estimar la calidad de transmisión en redes ópticas de enrutamiento por longitud de onda. Este estimador consta de dos módulos operando conjuntamente: un decisor que clasifica los lightpaths en función de su longitud total y un módulo basado en la técnica denominada Case-Based Reasoning que los clasifica atendiendo al caso más parecido almacenado en su base de datos. El decisor estima si la calidad de un lightpath es baja o alta en función de si la longitud de éste es mayor al límite superior o menor al límite inferior de un área de incertidumbre en el que es difícil predecir en principio la calidad de transmisión. Los umbrales de esta área de incertidumbre se calculan de manera artificial con la mera observación de los datos (longitudes pequeñas son más probables de ofrecer una mayor calidad de transmisión, mientras que longitudes muy altas afectan negativamente a dicha calidad). El módulo CBR se encarga de clasificar precisamente los lightpaths dentro del área de incertidumbre haciendo uso de una base de conocimiento que puede optimizarse mediante técnicas de aprendizaje y olvido, es decir, nuevos casos que aporten verdadero significado pueden almacenarse en dicha base de datos y casos redundantes pueden eliminarse.

Con estos antecedentes, lo que se plantea en este TFM es explorar las posibles ventajas derivadas del uso de métodos de aprendizaje automático en la resolución de este problema de clasificación binaria. Para ello se establece el siguiente enfoque:

1. Se replica el artículo en las condiciones exactas en las que éste se desarrolló (incluyendo mismas suposiciones, datos y entornos de programación y simulación), pero sustituyendo el módulo CBR por un algoritmo conocido como máquinas de vectores soporte (SVM). El estudio, por lo tanto, se realiza también en MATLAB.
2. Demostrar que en esas condiciones, SVM ofrece iguales o mejores prestaciones en términos de acierto en clasificación y en un tiempo de



clasificación por lightpath menor (aspecto crítico de diseño en redes dinámicas).

3. Una vez demostrado que SVM supera las prestaciones del sistema propuesto en el artículo, se traslada el estudio a un entorno de programación que facilite la comparación entre distintos algoritmos de aprendizaje automático (R, con ayuda del paquete caret).
4. En ese nuevo entorno se replica de nuevo el artículo con SVM y se proponen nuevas alternativas: regresión logística, árboles de decisión y bosques aleatorios. Si alguna de ellas supera las prestaciones de SVM en este nuevo entorno, quedará demostrado que también superarían las del sistema propuesto en el artículo.
5. Finalmente, puesto que el decisor propuesto en el artículo es una herramienta artificial fundamentada en unos umbrales de decisión que definen el área de incertidumbre de manera poco sólida, se diseña un nuevo escenario en el que los algoritmos de aprendizaje automático son los únicos responsables de clasificar los lightpaths, es decir, los clasifican sin importar la longitud total de los mismos (el propio algoritmo aprenderá la importancia de la longitud del lightpath en el problema de clasificación propuesto).

Los algoritmos de aprendizaje automático seleccionados son entrenados mediante validación cruzada (concretamente con 10-fold cross-validation), proceso en el que además se optimiza, dependiendo del escenario, uno de dos posibles parámetros de rendimiento, acierto o área bajo la curva ROC, mediante la búsqueda de la mejor sintonización de los parámetros propios de cada método. Por ejemplo, SVM posee como parámetros sintonizables  $C$  y  $\gamma$ , regresión logística posee  $\alpha$  y  $\lambda$ , CART posee el parámetro de complejidad y bosques aleatorios el  $mtry$  (número de predictores elegidos aleatoriamente).

Antes de cada entrenamiento se separan unos datos de test con el fin de que éstos no sean utilizados en ningún paso de la fase de entrenamiento. Tanto los conjuntos de datos para entrenamiento como para test son controlados con una semilla para que las comparaciones entre modelos y escenarios sea lo más justa posible.

Para el escenario en el que se replica exactamente el artículo en MATLAB, se simula el sistema propuesto tanto para la red alemana, Deutsche Telekom (DT) como para la europea (GÉANT2). Grupos de 6000 y 36000 datos de test se generan aleatoriamente para cada una de estas redes, respectivamente y grupos de entre 500 y 5000 datos de entrenamiento pertenecientes únicamente al área de incertidumbre se generan también aleatoriamente para entrenar SVM. Como variables de salida se obtienen el acierto en clasificación (que comprende la acción conjunta del decisor más

SVM) y el tiempo de computación por lightpath. Como novedad frente al artículo de referencia también se obtiene el tiempo de entrenamiento del SVM y su acierto interno (testando los propios datos de entrenamiento). Los experimentos se repiten 100 veces para obtener valores con significado estadístico. Este estudio también se realiza particularizado únicamente para el área de incertidumbre. De hecho, el estudio para GÉANT2 sólo se realiza dentro de esta particularización, puesto que los tiempos de entrenamiento para SVM son demasiado elevados. En todos los casos SVM mejora ligeramente el acierto en clasificación con respecto a CBR y mejora sustancialmente el tiempo de computación por lightpath (hasta 40 veces más rápido que CBR normal y hasta 5 veces más rápido que el CBR optimizado).

Una vez demostrado el hecho de que SVM mejora las prestaciones del sistema propuesto en el artículo en sus mismas condiciones, el estudio se replica de nuevo en R con los nuevos algoritmos alternativos. En este escenario y los siguientes, el estudio se particulariza para DT y los algoritmos son entrenados una única vez (para distintos tamaños del set de datos de entrenamiento) y se testean con 10 grupos de test para obtener resultados con significado estadístico. Bosques aleatorios y regresión logística alcanzan niveles de acierto en clasificación perfectamente comparables a los de SVM en este nuevo escenario y con tiempos de computación por lightpath significativamente menores. Además sus tiempos de entrenamiento son varios órdenes de magnitud más pequeños. CART, por el contrario, ofrece peores prestaciones en acierto, aunque es con diferencia el más rápido entrenando y clasificando.

Una vez demostrado que hay alternativas mejores a SVM en estos últimos escenarios, se puede suponer que son a su vez mejores que el sistema basado en CBR propuesto en el artículo.

El último paso consiste en trasladar dichas alternativas a un nuevo escenario en el que el decisor es eliminado. En este escenario SVM ya ha sido descartado como candidato ideal y el estudio se particulariza para CART, bosques aleatorios y regresión logística. Los datos de entrenamiento se extienden para poder afrontar este cambio. De esta manera, los sets de datos de entrenamiento contienen los mismos casos usados anteriormente (que se encontraban dentro de la zona de incertidumbre) y a éstos se les añaden datos fuera de la zona de incertidumbre en tal número que la proporción de datos dentro y fuera del área de incertidumbre se corresponde con la original de la base de datos completa. La fase de test se realiza exactamente con los mismos datos del escenario anterior, para seguir asegurando comparaciones justas entre escenarios y modelos.

Bosques aleatorios y regresión logística alcanzan de nuevo valores de acierto en clasificación comparables a SVM en el escenario con decisor, con valores de computación por lighthouse ligeramente mayores (el decisor tenía una influencia dramática sobre el tiempo de computación) pero todavía significativamente por debajo de los tiempos obtenidos por CBR. CART, de nuevo obtiene peores prestaciones en acierto y significativamente mejores prestaciones en tiempo. Queda demostrado entonces que bosques aleatorios y regresión logística son candidatos perfectamente válidos para sustituir por sí solos el sistema completo decisor más CBR descrito en el artículo de referencia.

Falta decidir el mejor candidato entre ambos, puesto que, aunque bosques aleatorios ofrece mejores prestaciones en acierto, es más lento a la hora de predecir y, sobre todo, de entrenar, que regresión logística. Para poder hacer la elección, se hace uso de un último grado de libertad en el diseño: el umbral de decisión. Se razona que un operador de red podría estar más interesado en sistemas que estimaran la calidad de transmisión cometiendo muy pocos errores al identificar casos positivos (categoría de calidad alta), es decir, que conllevaran muy pocos o ningún falso positivo (que fueran muy específicos), puesto que éstos inciden directamente en la calidad del servicio. Por ello se estudia incrementar el umbral de decisión (la clasificación se efectúa con un umbral de 0.5 por defecto, tras haber asignado un valor de probabilidad de pertenencia a una cierta clase para cada caso) para favorecer la disminución de falsos positivos (subiendo la métrica de especificidad), aunque ello conlleve una bajada de la tasa de acierto al provocar una bajada de la sensibilidad del sistema (se obtienen más falsos negativos). En este contexto, bosques aleatorios ofrece un mejor balance entre especificidad y sensibilidad, pudiendo aumentar el primero sin sancionar en exceso el segundo con umbrales de decisión mucho más razonables que regresión logística. Por todo ello se selecciona bosques aleatorios como el mejor candidato.