Machine-Learning Method for Quality of Transmission Prediction of Unestablished Lightpaths

Cristina Rottondi, Luca Barletta, Alessandro Giusti, and Massimo Tornatore

***Abstract*—Predicting the quality of transmission (QoT) of a lightpath prior to its deployment is a step of capital importance for an optimized design of optical networks. Due to the continuous advances in optical transmission, the number of design parameters available to system engineers (e.g., modulation formats, baud rate, code rate, etc.) is growing dramatically, thus significantly increasing the alternative scenarios for lightpath deployment. As of today, existing (pre-deployment) estimation techniques for lightpath QoT belong to two categories: "exact" analytical models estimating physical-layer impairments, which provide accurate results but incur heavy computational requirements, and margined formulas, which are computationally faster but typically introduce high link margins that lead to underutilization of network resources. In this paper, we explore a third option, i.e., machine learning (ML), as ML techniques have already been successfully applied for optimization and performance prediction of complex systems where analytical models are hard to derive and/or numerical procedures impose high computational burden. We investigate a ML classifier that predicts whether the bit error rate of unestablished lightpaths meets the required system threshold based on traffic volume, desired route, and modulation format. The classifier is trained and tested on synthetic data and its performance is assessed over different network topologies and for various combinations of classification features. Results in terms of classifier accuracy are promising and motivate further investigation over real field data.**

***Index Terms*—Bit error rate prediction; Machine learning; Quality of transmission.**

I. Introduction

Thanks to the widespread adoption of coherent technology, optical communication has significantly progressed in recent years and now offers a plethora of design parameters for lightpath deployment. Several choices such as modulation format, baud rate, forward error correction (FEC) coding, single/multicarrier transmission, adaptive channel spacings, and flex-grid network technologies, among others, offer a variety of "degrees of freedom" to system and network engineers, thus making the number of possible combinations for lightpath deployment grow dramatically.

In this context, predicting the lightpath quality of transmission (QoT) prior to deployment is essential to discern the most effective solution and for an optimized design and planning of the optical network. As of today, existing (pre-deployment) estimation techniques for lightpath QoT can be roughly classified into two categories. On the one hand, sophisticated analytical models (e.g., split-step Fourier method [1]) capturing different physical layer impairments can be used to estimate with great precision the bit error rate and reach of a given lightpath, but they impose high computational requirements that are not compatible with real-time prediction and are not scalable to large network topologies and dynamic network operation. On the other hand, approximated formulas (e.g., simplified power budget with nonlinear-impairment estimations based on a Gaussian model [2]) introduce higher link margins in the calculation of the lightpath budget to compensate for model inaccuracies, thus leading to an underutilization of network resources [3].

An alternative approach to QoT prediction relies on sensing the QoT of already deployed lightpaths by means of optical performance monitors (OPMs) [4] installed at the receiver side and on exploiting the knowledge extracted from field data to predict the QoT of unestablished lightpaths [5]. To this aim, different machine-learning (ML) techniques have been recently investigated, e.g., network kriging [6], case-based reasoning [7], and neural networks [8,9].

In this paper, we investigate and apply a ML-based classifier to predict the probability that the bit error rate (BER) of a candidate lightpath will not exceed the system tolerance threshold, using as features the traffic volume to be served, modulation format, lightpath total length, length of its longest link, and number of lightpath links. To train the classifier, we assume that either BER measurements over already-deployed lightpaths are provided by field OPMs[1] (or directly by optical transceivers) or that, in the absence of real field data, a BER estimation tool (E-Tool) is used to generate synthetic data. In the remainder of this paper, we opt for the latter approach due to the difficulty of

Manuscript received July 11, 2017; revised November 28, 2017; accepted December 5, 2017; published February 1, 2018 (Doc. ID 301828).

C. Rottondi (e-mail: cristina.rottondi@supsi.ch) and A. Giusti are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), University of Lugano (USI)–University of Applied Science and Arts of Southern Switzerland (SUPSI), Lugano, Switzerland.

L. Barletta and M. Tornatore are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy.

https://doi.org/10.1364/JOCN.10.00A286

[1] The proposed classifier is agnostic to the number of input features and could rely on multiple field-measured parameters, if available.

retrieving field data. The classification output is meant to be provided to a routing and spectrum assignment (RSA) algorithm that will make the final deployment decision.

In our performance assessment, we provide a specific focus on how classification performance is influenced by the choice of the training data: relying on historical BER measurements obtained by observing the lightpaths deployed during normal network operations might not suffice to achieve good results. Therefore, it might be necessary to deploy lightpath probes to evaluate in the field the BER of lightpath configurations that would normally not be adopted to serve user traffic.

The remainder of the paper is structured as follows: in Section II we briefly overview the related literature and in Section III we provide some background notions on ML classification. Section IV describes the assumed transmission model and our E-Tool, Section V illustrates the proposed ML binary classifier, and Section VI assesses the classifier performance. Future research directions are discussed in Section VII and conclusions are drawn in the Section VIII.

II. Related Work

The adoption of ML techniques as decisional support tools for the design and planning of optical networks has recently gained considerable attention in the scientific community. A few works have already appeared, which apply ML approaches in both physical and networking layers.

At the physical layer, techniques such as Bayesian filtering have been proposed for parameter estimation in models for laser amplitude and phase noise characterization [10,11], whereas a ML detector based on the distance-weight k-nearest neighbors (kNN) algorithm has been proposed to overcome system impairments (e.g., non-Gaussian symmetric noise, laser phase noise, and nonlinear phase noise) in zero-dispersion and dispersion-managed links [12]. Ridge and kernelized Bayesian regression models have been employed for the characterization and mitigation of power excursions in gain-controlled erbium-doped fiber amplifiers [13].

At the network layer, ML-based frameworks for the control and management of optical networks have been proposed: in Ref. [14], artificial neural networks are used to predict the evolution of the network traffic, whereas, in Ref. [15], reinforcement learning techniques are employed by a resource-allocation agent and incorporated into a cognitive architecture-on-demand control plane.

Coming to the specific target of this study (QoT prediction of a candidate lightpath prior to establishment), some ML-based approaches have been investigated: regression models such as network kriging and least-squares minimization with l_2-norm regularization have been applied in Refs. [6,16] to estimate the QoT of multiple lightpaths in terms of BER, by relying on the measurements obtained from a limited number of "active lightpaths" (i.e., lightpaths that carry dummy traffic and are instead used as measurement probes). In our paper, we assume that the BER of an already established lightpath is measured by means of OPMs installed at the lightpath termination nodes, as described in Ref. [4]. A neural network fed with either synthetic or field data is used in Ref. [9] to evaluate the Q-factor of multicast connections, whereas a case-based reasoning technique (i.e., an artificial intelligence method which makes decisions based on previously observed data stored in a knowledge database) is proposed in Refs. [7,17] to decide whether the BER of an unestablished lightpath will be above or below a given system threshold. In this paper, we address a QoT-prediction problem as in Refs. [6,9,16], but our approach is significantly more complex, as we assume that different combinations of routes and modulation formats can be used for the candidate lightpath, and our proposed ML classifier returns the most suitable combinations (i.e., the ones having the highest probability of ensuring a BER below threshold). Note that a short, summarized version of this study can be found in Ref. [18], but in this extended version, we use a more accurate BER-calculation model for the generation of synthetic data which takes into account nonlinear effects; we consider a more realistic procedure for the generation of training datasets, which emulates the evolution of a dynamic routing and spectrum assignment with first-fit criterion; and we provide a much more extensive performance assessment, considering different network topologies and various sets of classification features.

III. Background on Machine Learning Classifiers

This section provides some background on binary classifiers based on ML and describes the performance metrics used to evaluate their effectiveness.

A. Basic Principles

We consider in the following an *instance* (or "sample") as a set of numerical and/or categorical values (or "features") representing an instantiation of our problem. In the context of QoT prediction of unestablished lightpaths, the features characterize the lightpath we want to deploy: an example of a numerical feature is the volume of the traffic request we want to serve, whereas an example of a categorical feature is the modulation format we want to use for transmission. A set of instances, which are considered independent of each other, is named a *dataset*.

We associate with each instance a *class* which is described by a binary value: 1 if it satisfies a given rule (positive instances), 0 otherwise (negative instances). For example, considering the BER as a QoT metric, we associate 1 with an instance if the BER of the lightpath characterized by the features constituting the instance is below a certain threshold T, 0 otherwise.

In this work, we want to build a ML algorithm that, given an instance, predicts its class; this is a *classification* problem and such an algorithm is called a *binary classifier* (when the problem is instead to estimate a real-valued target, the problem is called *regression* and is solved by means of a *regressor*). One can consider a classifier as a function mapping a point of the space of features to a real number. Such a real number is the score of the instance, i.e., its probability of belonging to class True.

Note that features should be chosen in order to contain information that is useful to discriminate the class of an instance. Non-informative features—i.e., features that show no correlation to the class of the instance—are known to reduce the performance of classifiers, even though different classifiers have different sensitivity to this issue. In this regard, we deem as an important contribution of this paper the identification of a methodology for the identification of the most informative features among those initially selected for the QoT classification problem (see Subsection VI.C).

Before it can be used, a classifier needs to be trained by means of a training dataset, i.e., a set of instances whose class is known. In this phase, the classifier learns a mapping between the space of features and the class. Many different classification algorithms have been proposed in the literature [19] and are routinely used by ML practitioners; algorithms differ in terms of achievable accuracy, scalability to datasets with many instances and/or features, computational effort required for training and for testing, sensitivity to outliers, and interpretability of the resulting models. Random forests, logistic classifiers, support vector machines (SVMs), kNN, and neural networks are among the most frequently adopted techniques in recent works.

Once a classifier is trained, it can be used to test an instance that was not part of the training set. Given the numerical features belonging to such a test instance, the output produced by the classifier is the predicted probability $\hat{P}_{\mathrm{pos}}$ that the instance belongs to the positive class. This probability is the output score of the classifier: it will be very close to 1 for instances that are very likely to be positive, and close to 0 for instances that are very likely to be negative; a classifier may also return scores close to 0.5 for instances that are difficult to classify and may belong to either class. Since in practical applications one often needs a single well-defined output, the output score is typically binarized and the test instance is classified as positive if and only if its score is greater than or equal to a threshold $\gamma = 0.5$. Therefore, if for a given testing instance the classifier will produce a score equal to 0.95—which indicates that the instance is very likely positive—one will classify such an instance as positive, and the same would happen if the classifier produces a score equal to 0.51, indicating a large uncertainty in the prediction.

B. Performance Evaluation Metrics

Given a testing dataset, in the following we consider two measures to evaluate the performance quality of a trained classifier: *accuracy* and *area under the receiver operating characteristic curve* (AUC) [19].

The accuracy corresponds to the fraction of the test instances that are correctly classified. This measure is very easy to interpret and understand, but, as a classifier performance metric, suffers from a number of drawbacks:

- Accuracy is affected by the relative frequency of the two classes in the testing set; for example, if 90% of the samples in the testing set belong to one of the two classes, a trivial ("dummy") classifier that always returns the most frequent class (i.e., the class to which the majority of samples belong) will yield a 90% accuracy without actually producing any useful information; one should therefore be careful in interpreting accuracy values as metrics of classifier quality.
- Accuracy depends on the (somewhat arbitrary) choice of the threshold γ used to binarize the classifier outputs. Assume that a classifier consistently assigns a score of 0.6 to negative testing samples and 1 to positive testing samples, in a testing dataset where both classes appear with the same frequency. Such a classifier will classify all instances as positive and will yield a 50% accuracy. But the classifier outputs do contain useful information, and by using a different threshold (such as $\gamma = 0.8$), one may obtain 100% accuracy.
- Accuracy does not capture the ability of a classifier to identify difficult (ambiguous) instances as such. Consider a test dataset whose data is structured into three groups as follows. The first group (25% of the dataset) is composed of negative instances that can be easily identified as such; the second group (25% of the dataset) is composed by positive instances that can be easily identified as such; and the third group covers the remaining 50% of the test instances, which are equally divided between positive and negative instances but are impossible to differentiate from each other. Now consider two classifiers: *i)* Classifier A produces a score close to 0 for instances in the first group, close to 1 for instances in the second group, and randomly scattered close to 0.5 for instances in the third group. *ii)* Classifier B behaves as classifier A for instances in the first and second groups, but, for every instance in the third group, outputs a score which, at random, is either very close to 0 or very close to 1. The accuracy of both classifiers will be approximately 75%, because they will perfectly classify instances in the first two groups and correctly classify half of the instances in the third group. However, we would prefer classifier A, since it clearly differentiates easy (definitely correct) and difficult (probably incorrect) answers; in many application scenarios, knowing how confident a classifier is on an instance allows the user to make more informed decisions.

The AUC is the second metric we report, which obviates these issues and is widely used in the ML literature. Given a trained classifier and a testing set, one may set an arbitrary threshold γ and divide the testing instances into four groups:

- True Positive (TP) samples, i.e., positive samples that were correctly classified;
- True Negative (TN) samples, i.e., negative samples that were correctly classified;
- False Positive (FP) samples, i.e., negative samples that were incorrectly classified as positive;
- False Negative (FN) samples, i.e., positive samples that were incorrectly classified as negative.

Note that TP + FN corresponds to the number of positive samples in the testing dataset, and TN + FP corresponds to the number of negative samples in the testing dataset. We define the True positive rate (TPR) as the fraction of all positive instances that are classified as such, i.e., $\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$.

Conversely, the false positive rate (FPR) is the fraction of all negative instances that are incorrectly classified as positive: $\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$. Note that both the TPR and FPR are in the [0,1] range. An ideal classifier has $\mathrm{TPR} = 1$ and $\mathrm{FPR} = 0$.

By increasing the value of γ, we reduce the number of instances that we classify as positive and increase the number of samples that we classify as negative. This has the effect of decreasing TP while correspondingly increasing FN, and increasing TN while correspondingly decreasing FP. This reduces the TPR and also reduces the FPR.

The receiver operating characteristic (ROC) curve represents the FPR (on the horizontal axis) and the TPR (on the vertical axis) for different values of the threshold γ. For $\gamma = 1$, all instances are classified as negative (except those for which the classifier returned exactly 1.0, which we assume are very few); therefore $\mathrm{TPR} \approx 0$ and $\mathrm{FPR} \approx 0$: this lies on the bottom left of the ROC space. At the opposite end, for $\gamma = 0$, all instances are classified as positive; therefore $\mathrm{TPR} = 1$ and $\mathrm{FPR} = 1$.

The ROC curve always connects these two extremes. Any classifier that ignores the value of features (e.g., the classifier that always returns the most frequent class, or a classifier that returns answers at random) yields a ROC curve on the diagonal, regardless of the accuracy they can achieve. The ideal classifier (or any classifier that perfectly separates the two classes for at least one value of γ) yields a ROC curve connecting (0,0), (0,1), and (1,1). Classifiers that capture some useful information yield a ROC curve above the diagonal and approach but do not reach the point (0,1) on the ROC space.

The AUC is used as an effective and robust metric for the performance of binary classifiers that does not depend on the specific choice of γ. It ranges from 0.5 (for a useless classifier) to 1 (for an ideal classifier). According to Ref. [20], the value of the AUC is preferable to accuracy when evaluating the quality of classifiers, and it has a very useful intuitive interpretation as follows. Pick a negative and a positive sample at random from the testing dataset, and score both samples with the trained classifier; the AUC of the classifier can be interpreted as the probability that the classifier returns a larger score for the positive sample than for the negative sample. Therefore, for any choice of a negative and positive sample, a classifier with $\mathrm{AUC} = 1$ will score the former lower than the latter, which implies that there exists a threshold γ which perfectly separates negative and positive samples. Conversely, a classifier that returns random scores will have an AUC close to 0.5.

IV. Bit Error Rate Estimation Tool for Synthetic Data Generation

In this section, we discuss the assumptions on the system model and present the E-Tool we developed for the estimation of the BER once a candidate lightpath is deployed.

A. System Model

We assume that optical channels are multiplexed in a flexible grid with standard slice width of 12.5 GHz [21] and elastic transceivers operating at 28 Gbaud with optical bandwidth of 37.5 GHz (i.e., three slices). Superchannels with multiple adjacent transceivers are used to serve traffic demands exceeding the capacity of a single transceiver. We consider transparent links of dispersion uncompensated standard single-mode fibers where the signal power is restored by identical optical amplifiers equally spaced over the links (100 km), with gain $G = 20$ dB and noise figure $F = 5$ dB.

At the receiver, we consider a processing as in Ref. [22] for each transceiver: after coherent detection and analog-to-digital conversion, chromatic dispersion is electronically compensated, and an adaptive equalizer tackles other potential linear channel effects; finally, error counting for determining the pre-FEC BER is performed.

B. BER E-Tool

For the generation of synthetic data, we propose an E-Tool that, on input of a candidate lightpath and modulation format, calculates an estimate of the uncoded BER at the input of the FEC soft decoder (called pre-FEC BER in the following). Modern FEC codes have a threshold behavior: roughly speaking, if the pre-FEC BER is below a value (determined by the FEC code properties), then the BER at the output of the FEC is, with high probability, able to satisfy the BER system requirement. Provided that there exists a FEC code that bridges the gap between the pre-FEC BER and the required system BER, the pre-FEC BER takes the role of the target BER. A typical value for the BER target is $T = 4 \cdot 10^{-3}$ [22], which we adopt in the remainder of the paper.

In linear optical communication systems, i.e., those affected by chromatic dispersion and additive white Gaussian noise (AWGN) only, the pre-FEC BER depends on the pre-FEC signal-to-noise ratio (SNR) through a function determined by the modulation format. Hence, once the target BER is fixed, we can compute the required SNR. For a specific lightpath, the pre-FEC SNR can be estimated by a link budget that takes into account the transmitted power P_{in}, gains, and losses. If the pre-FEC SNR exceeds the required SNR, then the lightpath can be established.

C. Link Budget With Weak Nonlinear Propagation

When weak nonlinear propagation effects start to appear, a typical assumption is that the system behaves as a linear one, where the interference due to nonlinear interactions is treated as an independent contribution of AWGN with power P_{NLI}, as discussed in Ref. [2]. An estimate of the value of P_{NLI} due to inter- and intra-channel interference can be assessed thanks to the analysis in Refs. [23,24]: in particular, it turns out that P_{NLI} depends on the transmitted powers and the modulation formats of all channels. The analysis in Refs. [23,24] enables a fairly accurate estimation of the nonlinear interference power without resorting to computationally expensive simulations based on slip-step Fourier methods.

To simplify the analysis, in order to decouple the value of P_{in} from the modulation formats of all neighboring channels, we assume that all channels transmit at the same power P_{in}: a conservative value for P_{in} is determined using the Gaussian modulation format assumption of [22], where

$$P_{\mathrm{in}} = \frac{G - 22}{3} \text{ [dBmW]} \tag{1}$$

for a standard single-mode fiber.

Once P_{in} is found by Eq. (1), the nonlinear interference power P_{NLI} should be numerically computed, taking into account the actual modulation formats and spectrum occupancy of all channels. However, considering the channel of interest (central channel) and its neighboring channels along the whole lightpath, since the neighbors may change modulation format and bandwidth occupancy due to routing in the network and the flex-grid architecture, for simplicity we compute a conservative value of P_{NLI} by considering the effect along the entire path of the modulation format and the channel bandwidth of the spectrally nearest neighbor that the channel encounters along its path. This is done separately for its left and right neighbor. We claim that, in this way, we compute a conservative value of interference because the closer the neighbor channel, the larger the nonlinear interference. The value of P_{NLI} is numerically assessed using the approach provided in Ref. [24].

The nonlinear interference power has to be converted into a penalty to be accounted for in the link budget. The nonlinear penalty term L_{NLI} is computed as the difference (in dB) between P_{in} and the launch power $P_{\mathrm{in}}^{\mathrm{lin}}$ the transceiver would need to obtain the same reach if P_{NLI} were zero (linear system case). In formulae we have

$$L_{\mathrm{NLI}} = P_{\mathrm{in}} - P_{\mathrm{in}}^{\mathrm{lin}} \text{ [dB]}, \tag{2}$$

where

$$P_{\mathrm{in}}^{\mathrm{lin}} = \mathrm{SNR}_{\mathrm{req}} + L_{\mathrm{other}} + X \text{ [dBm W]}, \tag{3}$$

$$X = 10^{(P_{\mathrm{in}} - L_{\mathrm{other}} - \mathrm{SNR}_{\mathrm{req}})/10} - P_{\mathrm{NLI}}. \tag{4}$$

If X is negative, then L_{NLI} is set to an arbitrary large value, e.g., 50 dB, meaning that the link cannot be established. In Eq. (3), $\mathrm{SNR}_{\mathrm{req}}$ is the required SNR to reach the target BER, and L_{other} is a term that accounts for the following penalties:

- Back-to-back penalties for each modulation format in a 37.5 GHz grid network, whose values are extrapolated from ([22], Table I) and ([25], Fig. 7).
- A system margin which is a random parameter drawn according to an exponential distribution with average 2 dB. The randomization of the latter parameter accounts for the unpredictability of fast time-varying penalties (such as polarization effects [3]). We have chosen the exponential distribution since this is the maximum entropy distribution in the support $[0, \infty)$ with a constraint on the expected value: the maximum entropy principle is used to reflect the lack of knowledge (or information) of the unpredictable penalties.
- Small penalties up to 0.1 dB that account for the routing through reconfigurable optical add/drop multiplexers [26].

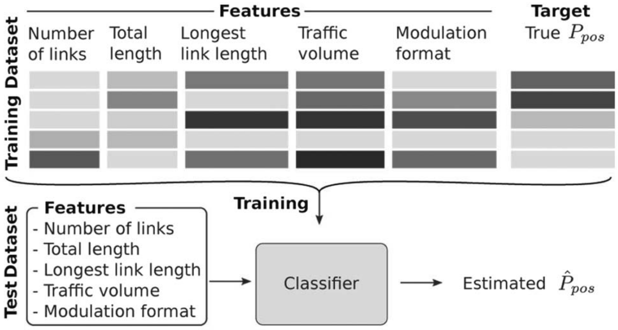


Fig. 1. Classifier structure.

Finally, the SNR at the input of the FEC decoder is estimated as

$$\mathrm{SNR}_{\mathrm{FEC}} = P_{\mathrm{in}} + 58 - F - G - L_{\mathrm{other}} - L_{\mathrm{NLI}}\text{[dB]}, \tag{5}$$

and converted to the corresponding pre-FEC BER value that depends on the modulation format of the central channel.

V. Proposed ML Classifier for QoT Classification

A. Classifier Description

As depicted in Fig. 1, our proposed classifier considers the following five features:

- number of links of the lightpath,
- lightpath total length (in km),
- length of its longest link (in km),
- traffic volume it serves (in Gb/s),
- modulation format used for transmission.

Note that none of these features accounts for cross-channel nonlinear effects. We can also consider six additional features in case complete knowledge of the lightpaths already deployed in the networks is available[2]:

- the smallest left/right guardband sizes separating the considered (super)channel from the nearest left/right neighboring (super)channels (i.e., we account for the worst case over all links traversed by the considered lightpath);

[2] This assumption is realistic especially in the case of incumbent network operators with proprietary infrastructures, but less likely to apply in the case of a network infrastructure shared among multiple operators, where alien lightpaths might be present.

- the traffic volume and modulation format of the left/right nearest neighboring (super)channels (i.e., the neighboring (super)channels separated by the smallest guardband, among all the left/right neighbors over every link traversed by the considered lightpath).

These six additional features capture information on cross-channel nonlinear effects.

The target variable that the classifier tries to predict is a binary variable, which is true if and only if the lightpath BER is lower than the system threshold $T = 4 \cdot 10^{-3}$. Note that the BER value is affected by other factors than those captured by the classification features (e.g., time-varying penalties). Therefore, it may occur that two dataset entries whose set of features are exactly the same exhibit different BER values and in turn different values of the target variable, i.e., the association between feature values and BER value is not deterministic. The classifier is trained on a *training dataset* and quantitatively evaluated on a separate *testing dataset* (see Fig. 1).

Since the classifier requires feature values that are numeric and that have comparable ranges to avoid numerical instability, we pre-process features as follows: *i)* the modulation format feature, which can take one of six possible categorical values, is replaced by six distinct binary features (one for each possible format); for each instance, the feature corresponding to the modulation format will take value 1, whereas the other five will take value 0; *ii)* the values of each feature are offset and rescaled to ensure that their distribution in the whole training set has mean 0 and standard deviation 1. At training time, we estimate the offset and scaling parameters for each feature. When the classifier is applied to test samples, the feature values are rescaled using such parameters.

On input of a test instance, the output produced by our classifier is the predicted probability $\hat{P}_{\mathrm{pos}}$ that the instance belongs to the positive class. The instance is then classified as positive if and only if such probability is greater than or equal to a threshold $\gamma = 0.5$.

B. Dataset Generation

For the dataset generation, we rely on synthetic data simulating measurements obtained from the field. We assume that the network is operated by a first-performing lightpath RSA according to margined calculations; then, when the lightpath is deployed, its actual (not margined) BER is used for training purposes. In the following, we first describe how RSA is performed, and then the procedures used to generate the training and testing datasets.

1) Routing and Spectrum Allocation: We consider a dynamic scenario in which traffic requests are generated by a Poisson process and cease after a negative exponential service time. Whenever a new request arrives, we precalculate three shortest paths and select among them the shortest one with enough available spectrum resources. Spectrum is assigned according to the first-fit algorithm. The modulation format chosen for transmission is the highest format compatible with lightpath length [note that, to emulate margined calculations, reaches are computed as in Section IV, with the worst-case assumption that neighboring channels are separated by the smallest possible guardband (12.5 GHz) and transmit using the highest modulation format (64-QAM), also considering a fixed system margin of 2 dB].

2) Generation of Training Datasets: Generally, ML assumes that a large enough amount of labeled data (i.e., instances for which the class is known *a priori*) are available for training purposes. Such data should be representative of the whole feature space (i.e., we should have historical data of deployed lightpaths satisfying the BER threshold as well as violating it) to construct a good prediction model, but, on a real network, collecting data over the whole feature space is extremely difficult. If we assume that historical data is derived from the operation of a real network, it is extremely unlikely that the actual BER of a deployed lightpath exceeds T (in fact, due to the link margins introduced in the reach computations of state-of-the-art RSA algorithms, all the lightpaths that are actually deployed satisfy the BER threshold), with the consequence that *we will obtain a dataset almost entirely constituted by positive instances*. Similarly, it is also extremely unlikely that the actual BER of a deployed lightpath is much lower than T, as margined formulas will always try to return BERs lower than but close to the threshold.

Therefore, in this paper, to model the creation of a balanced training dataset representative of the whole feature space, we assume that three possible approaches can be adopted during the training phase:

a) *Historical Data:* Training data are simply derived from actual deployed lightpaths. This approach is subject to all the shortcomings just described.

b) *Random Probes:* This approach consists of provisioning additional probe traffic requests over unoccupied spectrum portions, choosing their route and modulation format with the aim of artificially covering the whole feature space.

c) *Selective Probes:* This approach assumes that, each time we deploy a new lightpath according to margined formulas, probe traffic is momentarily transmitted over the same route using the least spectrally efficient modulation format with reach below the lightpath length (e.g., if the modulation format chosen for the lightpath deployment is 8-QAM, the probe traffic is transmitted using 16-QAM). Note that selective probes are easier to implement than random probes, but they do not allow us to cover the whole feature space.[3]

[3] Note also that, after evaluating the BER of the probe lightpath, we remove it, deploy the incoming traffic request, and compute the new BER. Moreover, we recompute the BER of every neighbor lightpath, since the installation of the new lightpath may change some of the features (i.e., the ones characterizing their nearest neighbors). The re-evaluated BER values will be included as new instances in the training dataset.

We now provide some more specific insights on how random probes are chosen. We randomly select a source–destination node pair and a traffic request in the range [50,500] Gbps with 50 Gbps granularity. We name each triplet of source node, destination node, and traffic volume a "scenario." For each scenario, we randomly select a route within the k shortest paths (in our simulations, $k = 3$), and one out of six possible modulation formats [i.e., dual polarization (DP)-BPSK, DP-QPSK, and DP-n-QAM, with $n = 8, 16, 32, 64$]. We also randomly select the left/right guardbands separating the lightpath from its neighbor channels over each link (with uniform distribution in the range [12.5,112.5] GHz) and their modulation format and traffic volume (with the same procedure just described), and we individuate the nearest left/right neighbors. At this point, the E-Tool can be used to evaluate the BER, and its output is considered the *ground truth*.

3) Generation of Testing Datasets: For the generation of the *testing dataset*, we randomly select M scenarios ($M = 50$ in our simulations), but now, for each scenario, we consider *all* 18 possible combinations of three routes and six modulation formats to be able to test the feasibility of all deployment options, and for each combination we provide a prediction. We name the combination of scenario, route, and modulation format a "setting" (for a total of $50 \cdot 3 \cdot 6 = 900$ settings). The features of their neighbor channels are generated with the same procedure used for generating random probes. Note that, for each setting, we repeat the BER calculation 100 times to obtain 100 instances of the exponentially distributed random variable that emulates fast time-varying impairments (see Subsection IV.C) and derive a statistical estimation of the probability, P_{pos}, that BER $<T$. Such probability will be compared to the predicted probability $\hat{P}_{\mathrm{pos}}$ produced as output by the classifier for each setting: the more closely $\hat{P}_{\mathrm{pos}}$ approaches P_{pos}, the better the performance of the classifier.

VI. Numerical Assessment

Numerical assessment has been performed using the Japan and National Science Foundation (NSF) topologies, depicted respectively in Figs. 2 and 3.

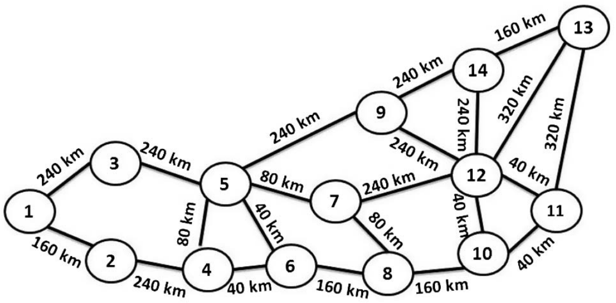


Fig. 2. Japan network topology.

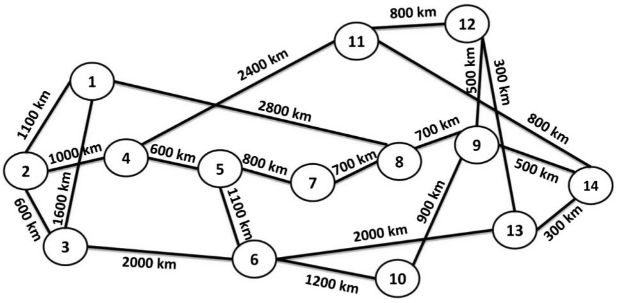


Fig. 3. NSF network topology.

TABLE I
Dataset Description

Dataset	A	B	C
Topology	Japan	NSF	Japan
Probing approach	Random	Random	Selective
Training dataset size	90,000	90,000	10,639
Percentage of probe inst. (in training dataset)	94.5%	94.5%	20.3%
Testing dataset size	90,000	90,000	–
Positive inst. (train)	53,121	9,081	9,845
Positive inst. (test)	52,120	8,515	–

For our experiments we generate three different (training and testing) datasets (A, B, and C, as described in Table I) and assess the classification performance by evaluating the *accuracy* and the *AUC*.

A. Comparison of Learning Methods for Classification

Using dataset A, we have compared three kNN classifiers [19] with $k = 1, 5, 25$ and five Random Forest (RF) classifiers [27] with 1, 5, 25, 100, 500 estimators. As a benchmark, we have additionally trained a dummy classifier which learns the most frequent class in the training set and always returns such a class for any testing instance, disregarding the feature values. The classifiers are trained with instances including all 11 features listed in Subsection V.A. In Table II, for each classifier, we report the training time and time required to evaluate one testing instance using a standard i5 processor: with the exception of the 1-nearest-neighbor classifier and the RF with one estimator, other options all perform comparably. Therefore, in the remainder of the paper, we adopt the RF classifier with 25 estimators for the next experiments since it provides a good tradeoff between performance and computational time.[4]

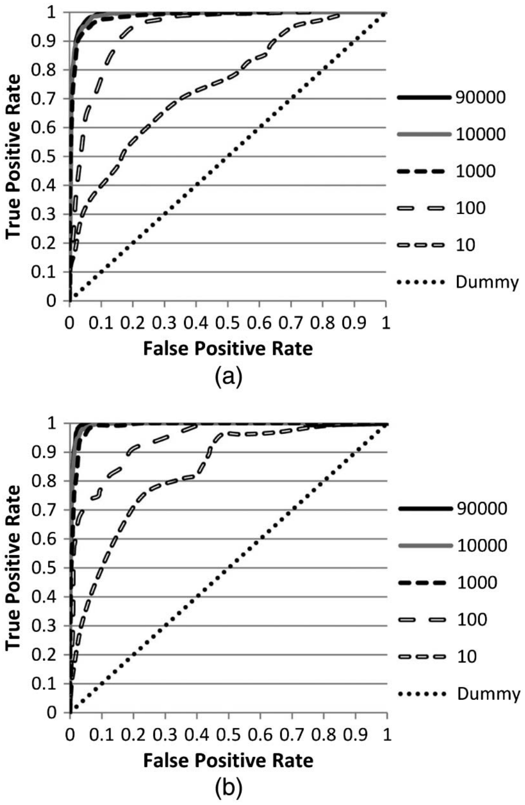
[4] For the sake of comparison, note that the running time of the method proposed in [24] and used to generate our synthetic datasets is on the order of a few seconds per instance on a standard i5 processor.

TABLE II
PERFORMANCE COMPARISON OF THE NINE CONSIDERED CLASSIFIERS

Classifier	Train. Time (s)	Classif. Time (μs)	AUC	Accuracy
Dummy Cl.	0.04	0.34	0.50	0.51
kNN $k = 1$	0.59	144.78	0.86	0.86
kNN $k = 5$	0.57	307.87	0.95	0.89
kNN $k = 25$	0.55	641.72	0.97	0.91
RF 1 est.	0.10	0.31	0.92	0.92
RF 5 est.	0.35	0.57	0.98	0.96
RF 25 est.	1.47	1.85	0.99	0.96
RF 100 est.	5.67	6.72	0.99	0.96
RF 500 est.	28.69	31.55	0.99	0.96

B. Impact of the Training Set Size

We now evaluate the impact of the number of training instances on the classification performance. Figure 4 shows the ROC curves obtained for datasets A [Fig. 4(a)] and B [Fig. 4(b)] when the classifier is trained on subsets of the training set composed of 10, 100, 1000, and 10,000 randomly sampled instances (we consider values up to 1000 to be realistic for field datasets), as well as the ROC curve of the classifier trained on the whole training dataset. Results obtained for both network topologies with 1000 training instances closely approach those obtained using the whole training dataset. Therefore, in the following, results will be obtained by training the classifier with only 1000 samples.



(a)

(b)

Fig. 4. ROC curve for different training set sizes, evaluated over datasets (a) A and (b) B. (a) Japan topology. (b) NSF topology.

TABLE III
EXAMPLE OF CLASSIFICATION OUTPUT FOR A TRAFFIC REQUEST OF 500 GBPS FROM NODE 8 TO 7 IN THE JAPAN NETWORK

		Lightpath		
	Modulation Format	8, 10, 12, 7	8, 6, 5, 7	8, 7
True P_{pos}	DP-BPSK	0.88	1.00	1.00
	DP-QPSK	0.86	1.00	1.00
	DP-8-QAM	0	0.31	1.00
	DP-16-QAM	0	0	1.00
	DP-32-QAM	0	0	1.00
	DP-64-QAM	0	0	0.36
Predicted $\hat{P}_{\mathrm{pos}}$	DP-BPSK	0.88	1.00	1.00
	DP-QPSK	0.78	1.00	1.00
	DP-8-QAM	0	0.56	1.00
	DP-16-QAM	0	0	1.00
	DP-32-QAM	0	0	1.00
	DP-64-QAM	0	0	0.32

We now focus on analyzing the classification output: it is worth noting that results for each test scenario (i.e., a triplet composed of source, destination, and traffic amount) can be arranged in a table, where each cell corresponds to one setting (i.e., a possible choice of modulation format and route for that triplet). Table III exemplifies the output provided by the classifier and compares the true P_{pos} of each setting (top) to the corresponding predicted $\hat{P}_{\mathrm{pos}}$ (bottom) for a traffic request of 500 Gbps from node 8 to node 7. Since the $\hat{P}_{\mathrm{pos}}$ of a given setting indicates the predicted probability that, when transmitting over the lightpath with the modulation format indicated by the considered setting, the BER will not exceed the threshold T, the closer $\hat{P}_{\mathrm{pos}}$ approaches 1, the safer the choice of that setting would be from a network design perspective. This output can be exploited to make the final decision about the deployment of the new lightpath by any RSA method: if the predicted probability is close to 0, the setting should not be adopted. Conversely, if $\hat{P}_{\mathrm{pos}}$ approaches 1, the network engineer can decide whether to adopt such a setting for the lightpath deployment based on the risk he/she accepts from a design perspective.

C. Analysis of Feature Relevance

We have so far considered 11 features in the proposed ML classifier. An important question at this point is which features are more important to achieve good accuracy and AUC. This question is of high practical value, as collecting more or less features poses higher or lower burdens in terms of monitor deployment and control complexity. In principle, removing irrelevant features would make the system less costly and complex to manage. Hence, we now evaluate the usefulness of each feature by comparing the classification performance after training the classifier over training datasets A and B, considering seven different subsets (S1 to S7) of the 11 features listed in Subsection V.A. The considered subsets

TABLE IV
Considered Feature Subsets

	S1	S2	S3	S4	S5	S6	S7
Number of links	✓	✓	✓	✓			
Lightpath length	✓	✓	✓	✓	✓	✓	
Length of longest link	✓	✓	✓	✓			
Traffic volume	✓	✓	✓		✓		✓
Modulation format	✓	✓		✓	✓	✓	✓
Guardband, modulation format, and traffic volume of nearest left and right neighbor	✓						

are listed in Table IV. Moreover, for each subset of features, we test the classifier over the full testing dataset and over a subset of its instances with BER in the range $[4 \cdot 10^{-4}, 4 \cdot 10^{-2}]$, i.e., focusing on the test samples which are near to the threshold T and, thus, more "difficult" to classify. The obtained results are reported in Fig. 5. In the case of the NSF topology we first notice that, when focusing on test instances "near to threshold," both metrics decrease w.r.t. the values obtained over the full testing dataset, whereas, for the Japan topology, such decrease is much less pronounced. This shows that the classification performance is still acceptable even for "difficult" test instances (i.e., those with BER values closely approaching T).

Results also show that, in both topologies, training the classifier with the feature sets S1, S2, and S5 leads to the highest and comparable AUC values. Note that S1 includes all 11 features, whereas S2 excludes the features characterizing the nearest-neighbor lightpaths. However, if we focus on the AUC of "near to threshold" instances, the results obtained in scenario S1 are slightly higher, which leads us to conclude that information on the closest neighbors does provide some insight into classification of the instances with BER close to T, as intuition would suggest.[5]

In particular, S5 includes only three attributes (total lightpath length, traffic volume, and modulation format) which suggest that information on the number of links and length of the longest link are not very useful if the previous three features are used. Indeed, in the transmission model implemented in our E-Tool, transmission impairments due to the traversal of intermediate nodes are on the order of 0.1 dB per node and thus have negligible impact on the BER computation. Similarly, knowing the length of the longest link of the lightpath does not bring much additional information on system impairments once the number of links and the lightpath length are known: this is due to the fact that both linear and nonlinear penalties are mainly determined by the two latter attributes.

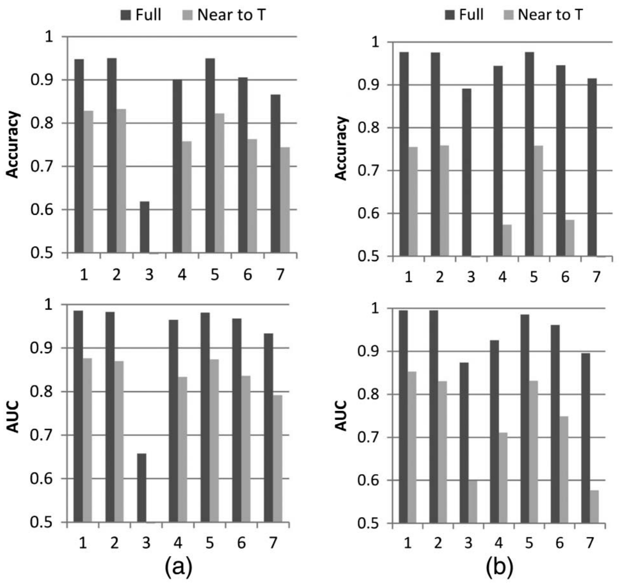


Fig. 5. Accuracy and AUC, depending on the feature set selection, for datasets (a) A and (b) B. (a) Japan topology. (b) NSF topology.

However, if we further remove either the traffic volume or the lightpath length from set S5 (as in subsets S6 and S7, respectively), classification performance degrades both in terms of AUC and accuracy, especially for "near to threshold" instances. Results similar to those achieved with feature set S6 are also obtained for the feature set S4, which also includes the number of links and the length of the longest link to the features already included in S6. Performance degradation becomes extremely severe when eliminating the modulation format from the feature set, as done in S3 (which contains only the traffic volume and the lightpath characteristics). With such training features, the AUC is slightly higher than 0.6 for the Japan topology, meaning that the improvement w.r.t. a random classification (which would return 0.5) is scarce.

Note, however, that our choice of modeling time-varying penalties with a random variable with exponential distribution (due to the lack of information about the true statistical distribution of such penalties, as discussed in Subsection IV.C) is the most conservative one and that different distributions (e.g., the uniform distribution with bounded support, which was adopted in Ref. [18]) would lead to better performance.

D. Impact of the Approach Adopted During the Training Phase to Collect Training Data (Historical, Random Probes, Selective Probes)

Finally, we focus on dataset C, and we first evaluate the classification performance after removing from the training set all the instances obtained by probing lightpaths carrying dummy traffic (i.e., we include only historical data as

[5] Note that, in our settings, nonlinear effects start to produce noticeable penalties only with large modulation formats, i.e., 32- and 64-QAM. In such scenarios, they cause a BER reduction of up to one order of magnitude in the worst case (i.e., in the case of two large neighbor channels using 64-QAM and separated from the considered lightpath by a 12.5 GHz guardband). However, these modulation formats are unlikely to be used in links with medium–long distances (above 300–400 km), as those in the NSF network. This is the reason why nonlinear effects due to neighboring channels are not significant in our case studies. Moreover, since we are not setting the launch power P_{in} to its optimal value (we are using the conservative value obtained with Gaussian formats, inferred from [23]), the impact of nonlinear effects is limited.

TABLE V
AUC Comparison of Probing Approaches

Training Set	AUC (Full Testing Dataset)	AUC (Testing Dataset near to T)
C (historical)	0.77	0.74
C (selective, 5% probes)	0.85	0.76
C (selective, 10% probes)	0.87	0.77
C (selective, 25% probes)	0.89	0.78
C (selective, 50% probes)	0.89	0.77
A (random)	0.98	0.87

described in Subsection V.B) and randomly sampling a set of 1000 instances, then repeat the experiment by replacing either 50, 100, or 500 instances with randomly chosen instances among the ones obtained via selective probing. AUC results averaged over 50 trials are reported in Table V and compared to results obtained by training the classifier over 1000 randomly chosen instances from dataset A. Results show that, as expected, relying exclusively on historical data leads to low AUC values, as the vast majority of them belong to the class of positive instances. Including instances obtained from selective probes in the training dataset (which mostly belong to the class of negative instances) improves the AUC: the highest improvement is obtained with 250 probes (increase by more than 0.12 when evaluating the classifier over all the instances of the testing set, and of almost 0.05 when focusing on test instances with BER in the range $[4 \cdot 10^{-4}, 4 \cdot 10^{-2}]$). Additional increase of the probing instances up to 500 did not lead to further performance improvements, and even caused performance degradation above 500. However, the overall performance is still lower than the one obtained when using training instances drawn from dataset A, which is constituted of almost 95% of instances obtained by random probes, thus ensuring a more exhaustive coverage of the whole feature space. It follows that a good classification performance can be achieved only at the price of an extensive deployment of random probe lightpaths.

E. Quantifying Potential Resource Savings

From the point of view of network design, quantifying the benefits of ML-aided QoT prediction in terms of resource savings (e.g., spectrum occupation and number of installed transceivers) is the main—yet unanswered—question. Resource savings depend on how the probabilistic output of the QoT classifier is integrated into the RSA: such a research direction is still largely unaddressed in the literature, and we briefly discuss the topic in the following.

A potential role of the classifier is to identify the lightpaths for which margined formulas were too conservative, i.e., where the next modulation format (the one with twice the number of constellation points) would still have led to a below-threshold BER. Therefore, for each given lightpath and corresponding modulation format obtained with margined formulas (conservative option), we construct a classification instance considering the next modulation format (aggressive option); such an instance may yield BER $< T$ (and therefore the aggressive option would allow saving resources over the conservative option) or not (which implies that a costly reconfiguration is necessary if the aggressive option is chosen). An RSA decisional algorithm may adopt the aggressive option if $\hat{P}_{\text{pos}} \geq \gamma$, and the conservative option otherwise. In the case of a false positive, we would wrongly choose the aggressive option and incur reconfiguration costs; in the case of a false negative, we would waste resources using the conservative option when the aggressive option would have been more efficient. By using large values of γ, we minimize the risk of false positives but have to accept more false negatives.

TABLE VI
Upper Bounds on Resource Savings Enabled by the Proposed QoT Classifier

γ	Average Reduction in Number of Installed Transceivers [%]	Average Reduction in Overall Spectrum Occupation [%]	Classification Accuracy	FPR
0.5	17.14	16.70	0.84	0.13
0.6	15.84	15.44	0.82	0.11
0.7	14.51	13.88	0.77	0.08
0.8	11.41	10.73	0.70	0.05
0.9	6.90	6.31	0.48	0.01
1	1.17	0.96	0.27	0.00

We compute an upper bound on the potential savings by considering the 2161 selective probe instances included in dataset C, and computing the resource savings *ignoring reconfiguration costs*, i.e., ignoring the impact of false positives. For values of γ ranging from 0.5 to 1, Table VI reports the percent savings in the number of installed transceivers and overall spectrum occupation; the classification accuracy; and the false positive rate, i.e., the fraction of below-threshold BER instances which were incorrectly classified as positive. We observe that savings are limited to 1% when $\gamma = 1$, i.e., the case with no false positives. By lowering γ, we are more prone to choosing the aggressive option: savings can reach 17% when γ is set to 0.5, at the expense of a larger false positive rate (FPR $= 0.13$). Future work must be devoted to identifying the best tradeoff between classification uncertainty and reduction of resource utilization, as well as identifying the technologies (e.g., bandwidth variable flexible transceivers, probabilistic constellation shaping) that could take most advantage of the adoption of ML-based QoT prediction techniques. More importantly, note that the assumptions used to generate our dataset are conservative in both the calculation of nonlinear impairments and fast time-varying penalties, and hence we expect resource savings to be more significant with realistic measurement datasets.

VII. Future Research Directions

Future work should address the application of *online* ML mechanisms, which are specifically designed for scenarios where data becomes available in a sequential order: whenever a new datum arrives, it is used to improve the current prediction model implemented by the ML algorithm, and the newly acquired knowledge will be adopted to make decisions at the next step. This online mechanism naturally fits dynamic RSA approaches, where traffic requests are generated at different moments in time and must be routed and allocated in a commensurate spectrum portion upon arrival. Moreover, *active* ML techniques [28] could be adopted to mitigate the issue of installing probe lightpaths: active ML algorithms are capable of interactively querying the user, asking to observe data with specific characteristics. This way, the number of samples to build an accurate predictor may be reduced. Therefore, if the process of generating data is costly (as in the case of probe lightpath deployment) active learning is a candidate approach to reduce the cost of dataset generation. However, when considering a real optical network scenario, it may be impossible to satisfy some of the queries of a ML active algorithm. For example, the algorithm may ask to observe the measurements obtained over a 1300 km long lightpath, but the deployment of such a lightpath may be impossible due to the structure of the network topology (i.e., a succession of consecutive links with total length of 1300 km may not exist). Therefore, a thorough investigation of the effectiveness of active ML techniques in reducing the training dataset size while taking into account the constraints imposed by the network structure and topology is necessary. Furthermore, considering that the consequences of classification errors may be catastrophic in terms of violation of Service Level Agreements stipulated with customers and content providers, and could lead to unacceptable QoS degradation, *cost-sensitive* ML learning approaches [29] should be investigated; such approaches allow for the definition of misclassification costs to penalize specific types of prediction errors. Costs caused by different kinds of errors can be arbitrarily defined, and the learning objective is to minimize the expected costs.

VIII. Conclusion

This paper proposes a machine-learning method to predict the quality of transmission of optical lightpaths prior to deployment: based on the lightpath characteristics (total length, length of the longest link, and number of lightpath links), modulation format used for transmission, and traffic volume to be served, the proposed algorithm predicts whether the bit error rate of the candidate lightpath will exceed a given system threshold. The performance of the classification algorithm is evaluated over a wide set of simulation scenarios. Results show that high values of accuracy and AUC can be achieved, though at the price of extensive deployment of probing lightpaths necessary to evaluate in the field the BER of lightpath configurations that would normally not be adopted to serve user traffic. Based on the reported results, our proposed classifier can be considered a useful component for integration into RSA decision tools.

Acknowledgment

Part of the work leading to these results has been supported by the European Community Metro-Haul project under grant agreement no. 761727.

References

[1] J. Shao, X. Liang, and S. Kumar, "Comparison of split-step Fourier schemes for simulating fiber optic communication systems," *IEEE Photon. J.*, vol. 6, no. 4, pp. 1–15, Aug. 2014.

[2] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "The GN-model of fiber non-linear propagation and its applications," *J. Lightwave Technol.*, vol. 32, no. 4, pp. 694–721, 2014.

[3] Y. Pointurier, "Design of low-margin optical networks," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. A9–A17, 2017.

[4] K. Christodoulopoulos, P. Kokkinos, A. Di Giglio, A. Pagano, N. Argyris, C. Spatharakis, S. Dris, H. Avramopoulos, J. C. Antona, C. Delezoide, P. Jennevé, J. Pesic, Y. Pointurier, N. Sambo, F. Cugini, P. Castoldi, G. Bernini, G. Carrozzo, and E. Varvarigos, "Orchestra-optical performance monitoring enabling flexible networking," in *17th Int. Conf. on Transparent Optical Networks (ICTON)*, Budapest, Hungary, 2015, pp. 1–4.

[5] E. Seve, J. Pesic, C. Delezoide, and Y. Pointurier, "Learning process for reducing uncertainties on network parameters and design margins," in *Optical Fiber Communication Conf.*, Los Angeles, California, 2017, paper W4F.6.

[6] N. Sambo, Y. Pointurier, F. Cugini, L. Valcarenghi, P. Castoldi, and I. Tomkos, "Lightpath establishment assisted by offline QoT estimation in transparent optical networks," *J. Opt. Commun. Netw.*, vol. 2, no. 11, pp. 928–937, 2010.

[7] T. Jiménez, J. C. Aguado, I. de Miguel, R. J. Durán, M. Angelou, N. Merayo, P. Fernández, R. M. Lorenzo, I. Tomkos, and E. J. Abril, "A cognitive quality of transmission estimator for core optical networks," *J. Lightwave Technol.*, vol. 31, no. 6, pp. 942–951, 2013.

[8] F. N. Khan, T. S. R. Shen, Y. Zhou, A. P. T. Lau, and C. Lu, "Optical performance monitoring using artificial neural networks trained with empirical moments of asynchronously sampled signal amplitudes," *IEEE Photon. Technol. Lett.*, vol. 24, no. 12, pp. 982–984, 2012.

[9] T. Panayiotou, S. Chatzis, and G. Ellinas, "Performance analysis of a data-driven quality-of-transmission decision approach on a dynamic multicast-capable metro optical network," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. 98–108, 2017.

[10] D. Zibar, L. H. H. de Carvalho, M. Piels, A. Doberstein, J. Diniz, B. Nebendahl, C. Franciscangelis, J. Estaran, H. Haisch, N. G. Gonzalez, J. C. R. F. de Oliveira, and I. T. Monroy, "Application of machine learning techniques for amplitude and phase noise characterization," *J. Lightwave Technol.*, vol. 33, no. 7, pp. 1333–1343, 2015.

[11] D. Zibar, M. Piels, R. Jones, and C. G. Schäeffer, "Machine learning techniques in optical communication," *J. Lightwave Technol.*, vol. 34, no. 6, pp. 1442–1452, 2016.

[12] D. Wang, M. Zhang, M. Fu, Z. Cai, Z. Li, H. Han, Y. Cui, and B. Luo, "Nonlinearity mitigation using a machine learning detector based on k-nearest neighbors," *IEEE Photon. Technol. Lett.*, vol. 28, no. 19, pp. 2102–2105, 2016.

[13] Y. Huang, C. L. Gutterman, P. Samadi, P. B. Cho, W. Samoud, C. Ware, M. Lourdiane, G. Zussman, and K. Bergman, "Dynamic mitigation of EDFA power excursions with machine learning," *Opt. Express*, vol. 25, no. 3, pp. 2245–2258, 2017.

[14] F. Morales, M. Ruiz, and L. Velasco, "Virtual network topology reconfiguration based on big data analytics for traffic prediction," in *Optical Fiber Communication Conf.*, Anaheim, California, 2016, paper Th3I-5.

[15] G. Zervas, K. Banias, B. R. Rofoee, N. Amaya, and D. Simeonidou, “Multi-core, multi-band and multi-dimensional cognitive optical networks: An architecture on demand approach,” in *14th Int. Conf. on Transparent Optical Networks (ICTON)*, Coventry, England, 2012, pp. 1–4.

[16] Y. Pointurier, M. Coates, and M. Rabbat, “Cross-layer monitoring in transparent optical networks,” *J. Opt. Commun. Netw.*, vol. 3, no. 3, pp. 189–198, 2011.

[17] I. deMiguel, R. J. Durán, T. Jiménez, N. Fernández, J. C. Aguado, R. M. Lorenzo, A. Caballero, I. T. Monroy, Y. Ye, A. Tymecki, I. Tomkos, M. Angelou, D. Klonidis, A. Francescon, D. Siracusa, and E. Salvadori, “Cognitive dynamic optical networks [Invited],” *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A107–A118, 2013.

[18] L. Barletta, A. Giusti, C. Rottondi, and M. Tornatore, “QoT estimation for unestablished lightpaths using machine learning,” in *Optical Fiber Communications Conf. and Exhibition (OFC)*, Los Angeles, California, 2017, pp. 1–3.

[19] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 128. Springer-Verlag, 2006, pp. 1–58.

[20] C. Ferri, J. Hernández-Orallo, and P. A. Flach, “A coherent interpretation of AUC as a measure of aggregated classification performance,” in *28th Int. Conf. on Machine Learning (ICML)*, 2011, pp. 657–664.

[21] “Spectral grids for WDM applications: DWDM frequency grid,” ITU-T Recommendation G.694.1, Feb. 2012 [Online]. Available: http://www.itu.int.

[22] G. Bosco, V. Curri, A. Carena, P. Poggiolini, and F. Forghieri, “On the performance of Nyquist-WDM terabit superchannels based on PM-BPSK, PM-QPSK, PM-8QAM or PM-16QAM subcarriers,” *J. Lightwave Technol.*, vol. 29, no. 1, pp. 53–61, 2011.

[23] A. Carena, G. Bosco, V. Curri, Y. Jiang, P. Poggiolini, and F. Forghieri, “EGN model of non-linear fiber propagation,” *Opt. Express*, vol. 22, no. 13, pp. 16335–16362, June 2014.

[24] R. Dar, M. Feder, A. Mecozzi, and M. Shtaif, “Accumulation of nonlinear interference noise in fiber-optic systems,” *Opt. Express*, vol. 22, no. 12, pp. 14199–14211, June 2014.

[25] X. Zhou, L. E. Nelson, P. Magill, R. Isaac, B. Zhu, D. W. Peckham, P. I. Borel, and K. Carlson, “High spectral efficiency 400 Gb/s transmission using PDM time-domain hybrid 32–64 QAM and training-assisted carrier recovery,” *J. Lightwave Technol.*, vol. 31, no. 7, pp. 999–1005, 2013.

[26] T. Rahman, A. Napoli, D. Rafique, B. Spinnler, M. Kuschnerov, I. Lobato, B. Clouet, M. Bohn, C. Okonkwo, and H. de Waardt, “On the mitigation of optical filtering penalties originating from ROADM cascade,” *IEEE Photon. Technol. Lett.*, vol. 26, no. 2, pp. 154–157, Jan. 2014.

[27] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[28] B. Settles, “Active learning literature survey,” Technical Report 1648, University of Wisconsin, Madison, Wisconsin, 2010, pp. 55–66.

[29] C. Elkan, “The foundations of cost-sensitive learning,” in *17th Int. Joint Conf. on Artificial Intelligence*, 2001, pp. 973–978.