POLITECNICO DI TORINO
Repository ISTITUZIONALE

Reducing probes for quality of transmission estimation in optical networks with active learning

Original
Reducing probes for quality of transmission estimation in optical networks with active learning / Azzimonti, D.; Rottondi, C.; Tornatore, M.. - In: JOURNAL OF OPTICAL COMMUNICATIONS AND NETWORKING. - ISSN 1943-0620. - ELETTRONICO. - 12:1(2020), pp. A38-A48. [10.1364/JOCN.12.000A38]

Availability:
This version is available at: 11583/2768612 since: 2019-11-27T14:41:18Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1364/JOCN.12.000A38

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

12 October 2024

Reducing Probes for QoT Estimation in Optical Networks with Active Learning

Dario Azzimonti, Cristina Rottondi, *Member, IEEE,* and Massimo Tornatore, *Member, IEEE*

***Abstract*—Estimating Quality of Transmission (QoT) of a lightpath before its establishment is a critical procedure for an efficient design and management of optical networks. Recently, supervised Machine Learning (ML) techniques for QoT estimation have been proposed as an effective alternative to well-established, yet approximated, analytic models, that often require to introduce conservative margins to compensate for model inaccuracies and uncertainties. Unfortunately, to ensure high estimation accuracy, the training set (i.e., the set of historical field data, or "samples", required to train these supervised ML algorithms) must be very large, while, in real network deployments, the number of monitored/monitorable ligthpaths is limited by several practical considerations. This is especially true for lightpaths with above-threshold BER (i.e., malfunctioning or wrongly-dimensioned lightpaths), which are infrequently observed during network operation. Samples with above-threshold BERs can be acquired by deploying probe lightpaths, but at the cost of increased operational expenditures and wastage of spectral resources. In this paper, we propose to use *active learning* to reduce the number of probes needed for ML-based QoT estimation. We build an estimation model based on Gaussian processes, which allows to iteratively identify those QoT instances that minimize estimation uncertainty. Numerical results using synthetically generated datasets show that, by using the proposed active learning-approach, we can achieve the same performance of standard offline supervised ML methods, but with a remarkable reduction (at least 21% and up to 95%) in the number of training samples.**

***Index Terms*—QoT estimation; Optical Networks; Lightpath Probing; Active Learning; Gaussian Processes;**

I. Introduction

In modern optical networks, following the adoption of coherent transmission and of a more flexible spectrum grid, network engineers can choose among several different transmission configurations (e.g., among multiple modulation formats and/or different frequency-slot widths). In this context, the choice of the most suitable transmission configuration for a new lightpath becomes a critical decision to ensure efficient resource utilization. Hence, the ability to quickly and precisely estimate the Quality of Transmission (QoT) of a lightpath prior to its deployment has gained even more importance and has attracted considerable research attention [1].

Traditionally, QoT prediction has been performed either using computationally-intensive transmission emulators (as those based on the Split-Step Fourier method [2]) or based on approximated analytic models (such as the Gaussian Noise (GN) model [3] or any of its several more recent extensions [4]). The latter models are widely adopted today due to their simplicity and precision, but they require the introduction of conservative margins to account for uncertainties in the values of some input parameters (e.g. deviation from nominal values in hardware equipment due to aging), incomplete knowledge of network occupation (e.g., the presence of alien wavelengths) or of the network configuration (e.g. the link lengths could be not exactly known in the case of fiber rental).

Machine Learning (ML) has been investigated as a possible new direction to build tools for QoT estimation that promise to avoid scalability or uncertainty limits of the previous approaches. The vast majority of recently-proposed ML-based QoT-estimation tools adopt offline supervised learning, i.e., the ML algorithms are trained using a training set of historical data. Such datasets contain samples of transmission parameters (as, most typically, Bit Error Rate (BER) or Optical Signal to Noise Ratio (OSNR)) collected during network operation by Optical Performance Monitors (OPMs) [5] located at the receivers of already-deployed ligthpaths. Each of these samples is associated to a set of features characterizing the considered lightpath (e.g., length, number of traversed nodes, modulation format used for transmission, etc.). Based on the training data, ML algorithms extract the knowledge necessary to estimate the QoT of future lightpaths (i.e., predicting whether their OSNR/BER will exceed a given system threshold).

However, to obtain high estimation accuracy, the training set needs to be sufficiently large and to contain samples that explore the whole feature space, and such samples might not be always available in a production network. A large number of causes may limit the amount of collectible data (e.g., insufficient telemetry, old legacy equipment, etc.), and, in a more general sense, data might be expensive to acquire (e.g.,at the early operation stage of the system, when historical data are still scarce), and/or to label, and shall be extracted/queried only when necessary. In particular, lightpaths with above-threshold BER (i.e., exhibiting faults or malfunctions) are unlikely to be observed in real deployments due to the conservative system-design strategies (i.e., high margins) typically adopted to guarantee transmission quality. To complement the training set with above-BER-threshold samples, probe lightpaths [6] can be used to acquire data associated with critical transmission configurations that would not be normally adopted for customer traffic. However, collecting these probes incur in additional operational costs and higher occupation of spectral resources.

How to provide accurate ML-based QoT predictions in presence of small/incomplete training sets is an important and scarcely explored research issue. In this paper, we propose an *Active Learning* (AL) method that works on top of a ML

Dario Azzimonti is with the Dalle Molle Institute for Artificial Intelligence, Lugano, Switzerland. Cristina Rottondi is with the Dept. of Electronics and Telecommunications of Politecnico di Torino, Italy. Massimo Tornatore is with the Dept. of Electronics, Information and Bioengineering of Politecnico di Milano, Italy.

predictor based on Gaussian Processes (GP). After an initial training with a limited number of instances, the proposed AL algorithm iteratively asks to collect only a few selected training samples with specific characteristics, with the intent of minimizing the number of required samples. In particular, samples that minimize a specifically tailored acquisition function will be sought for. Such acquisition function is designed to maximize the increase of prediction accuracy at every iteration[1].

The rest of the paper is organized as follows: after a brief overview on related work in Section II, we introduce some background notions on Gaussian Processes and AL in Section III and then describe the proposed AL solution for QoT estimation in Section IV. In Section V we numerically assess its performance, showing that we can obtain higher values of the Area Under the ROC Curve (AUC), where ROC stands for Receiver Operating Characteristic, with much fewer training instances. We draw our conclusions in Section VI.

II. Related Work

Several studies on QoT estimation of unestablished lightpaths have recently appeared (see [8] for a comprehensive survey). Ref. [9], [10] adopted a cognitive *Case Based Reasoning* (CBR) approach, which stores in a database a list of Q-factor measurements, together with a set of characteristics of the associated lightpaths. When a new lightpath has to be deployed, the table entries which exhibit highest similarity to the candidate lightpath are used to make an estimation of the expected Q-factor.

A similar approach is adopted in [11] and [12], respectively to tune design margins in presence of unknown network parameters, or to adjust input parameters for the GN model. In both studies, field data are collected and ingested by a prediction tool that outputs an OSNR estimation based on educated guesses on the unknown network/GN model parameters.

Two alternative methods named *network kriging* and *norm $\mathcal{L}_2$ minimization* are applied in [13]–[16] to perform QoT estimation. These methods require the installation of probe lightpaths carrying dummy traffic to acquire field measurements to compute an estimation of the Q-factor of already established or candidate lightpaths. As probe installation is costly, the proposed methods explore the tradeoff of minimization of the number of deployed probes and maximization of the information gain. Our active learning-based method is conceptually similar: the criterion we adopt to select probe lightpaths to be deployed is the minimization of an acquisition function that quantifies the prediction uncertainty. However, the kriging approach requires that the metrics characterizing a ligthpath can be expressed as a linear combination of the link-level metrics calculated over its links. Such linearity assumption is not necessary in our framework. Moreover, the above mentioned studies do not take into account the coexistence of multiple modulation formats and mainly focus on single-rate WDM networks (flexi-grid networks are addressed only in [16], assuming dual baud rate transmission with single modulation format), whereas we consider traffic requests of different volume that can be served with 6 different modulation formats in a flexi-grid network.

Among state-of-the-art learning algorithms, Gaussian Processes, Random Forests and Artificial Neural Networks have been applied to perform the task of QoT estimation. In [17], Gaussian-Processes nonlinear Regression is adopted to predict the BER of an optical communication system using as features the channel input power, lightpath length, symbol rate and inter-channel spacing, whereas in [6] *Random Forests* are used to predict whether the BER of unestablished lightpaths will exceed a given threshold, based on set of features representing the transmission parameters of the lightpath.

Artificial Neural Networks (ANNs) have proved to outperform other learning algorithms in the task of QoT estimation [18]. In [19], a transfer-learning approach is adopted to ensure portability of a Q-factor prediction model over different network topologies, without retraining it from scratch: a few new samples are used to re-train previously learned ANN weights instead of starting from randomly initialized ones. An ANN-based OSNR predictor is proposed in [20] and assessed using field data gathered from an experimental testbed with WDM channels transmitting in the range 60-100km. The authors of [1] design a QoT estimator for intra/inter-domain lightpaths in a multi-domain network scenario with alien wavelengths. In [21], Deep Graph Convolutional Neural Networks are adopted to estimate the QoT of unestablished ligthpaths, also capturing the impact of its deployment on already established ligthpaths in terms of crosstalk, considering a flexi-grid optical network with multicore fibers and four different modulation formats. The accuracy of the proposed framework is assessed in a dynamic traffic scenario. ANNs have been adopted also in [22] in the context of unicast/multicast networks with dynamic traffic and their performance has been compared to that of a traditional Q-factor model, showing under which conditions it can be safely replaced by the data-diven ML approach.

Note that all the above mentioned learning frameworks adopt offline supervised training or, in a few cases, online learning (i.e., when training data becomes available sequentially, for example in mini-batches). Similarly to online learning, in our work the training dataset size increases as the number of iteration grows. However, while in online learning there is usually no control on the training data that becomes available, here we propose a principled way to deploy lightpaths that minimize an appropriate acquisition function in an Active Learning framework. To the best of our knowledge, ours is the first attempt to apply an AL approach for the QoT estimation task.

AL is a growing field of research in machine learning with applications in robotics [23], autoML [24] and uncertainty quantification [25]. AL is deeply rooted in the literature of design of experiments from statistics [26] and in Bayesian optimization [27]. Among AL methods, Gaussian process [28] based algorithms play a prominent role andthey have been long used, e.g., in Bayesian optimization algorithms for global optimization tasks [29] and in the computer experiments

[1]A preliminary version of this study appeared in [7]. Here, we detail the principles of the active learning approach adopted in our framework and provide an entirely novel and thorough numerical assessment of its predictive capabilities

literature [30], [31]. Such Bayesian optimization algorithms were then adapted to the problem of sampling a function around a specified threshold [25], [32], [33]. Here we propose an AL solution for QoT estimation based on the acquisition function presented in [25].

III. Background

A. The framework

An AL algorithm requires two components: a ML model that, given a training set, returns predictions at unobserved inputs, and an *acquisition function* that guides the selection of new instances to be added to the updated training set. These two components form the core of an iterative procedure where: (i) the ML model is fitted to the training set; (ii) the model predictions are used to build an acquisition function; (iii) the minimizer of the acquisition function determines the next data point to be added to the training set; (iv) the procedure is repeated. In the context of QoT estimation, the first component (i.e., the ML model) could either predict whether a certain instance is above or below the critical system threshold with a standard classifier, or it could predict the BER value with a regression model and then threshold the prediction. Here we follow the second approach because the predicted BER value is required to define the acquisition function in the active learning phase. The predictions are obtained with Gaussian Process (GP) regression.

B. Gaussian Processes

GPs can be considered as a Bayesian implementation of kernel methods used in both regression and classification tasks. They are completely characterized by a mean function $m : \mathbb{X} \to \mathbb{R}$ and a covariance kernel $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$, a positive semi-definite symmetric function of two arguments, see [28]. Moreover, GPs are probabilistic models that yield a posterior distribution over the possible values of the model, which means that they are naturally accompanied by an assessment of their uncertainty.

In GP regression we observe a training set of ℓ points in $\mathbb{X}$, $\mathbf{x}_\ell = \{x_1, \dots, x_\ell\}$, coupled with ℓ response values $\mathbf{y} = (y_1, \dots, y_\ell)^T \in \mathbb{R}^\ell$ where

$$y_i = f(x_i) + \varepsilon \tag{1}$$

for $x_i \in \mathbb{X}$, $i = 1, \dots, \ell$ with a measurement error $\varepsilon \sim N(0, \sigma^2_{\text{noise}})$ and we denote by $\mathbf{f} = (f(x_1), \dots, f(x_\ell)) \in \mathbb{R}^\ell$ the function values. The observation model described in (1) can be summarized as $p(\mathbf{y} \mid \mathbf{f}) = N(\mathbf{f}, \sigma^2_{\text{noise}} I_\ell)$, where $I_\ell \in \mathbb{R}^{\ell \times \ell}$ is the identity matrix. Note that for the problem at hand, x_i would be a vector of features describing the ith lightpath in the training set which is coupled with y_i, i.e., the BER value observed for that lightpath.

We assume that the function f is a realization of a GP, thus we are assuming a prior distribution for the vector $\mathbf{f}$ given by $p(\mathbf{f}) = N(\mathbf{m}, K)$, where $\mathbf{m} = [m(x_1), \dots, m(x_\ell)]^T \in \mathbb{R}^\ell$ and $K \in \mathbb{R}^{\ell \times \ell}$ is a positive definite matrix with entries determined by the covariance kernel k, i.e. $K_{i,j} = k(x_i, x_j)$ for $i, j = 1, \dots, \ell$. Given the observation model and a prior distribution we can use the Bayes' theorem to compute the posterior distribution of f given the observations, i.e.

$$p(\mathbf{f} \mid \mathbf{y}) = \frac{p(\mathbf{f}) p(\mathbf{y} \mid \mathbf{f})}{p(\mathbf{y})}. \tag{2}$$

In GP regression, the posterior has the remarkable property of being normally distributed, with analytical expressions for the posterior mean and covariance kernel (see [28], chapter 2).

The GP mean function m and kernel k are prior quantities chosen before observing the data and thus encode our prior knowledge. In particular, the kernel function determines the smoothness of the GP regression fit and can be used to encode prior knowledge on f. For example, if we expect f to be periodic, we can choose a periodic kernel and all prior realizations of the GP will be periodic functions. The BER function we consider in this work, a priori, does not have any specific property we can encode in k. For this reason, here we focus on stationary kernels chosen from a parametric family, such as the squared exponential or the Matérn family. Such choices only encode a prior knowledge about the smoothness of the function, as described in the following section.

The parametric families of kernels mentioned above depend on a few hyper-parameters θ that encode the scale of the output and the characteristic length scale of each input (see [28], chapter 4). The hyper-parameters can be learned from data by maximizing the marginal likelihood of the model. In the GP regression case this function is analytical ([28], chapter 2) and allows for the use of fast gradient-based optimizers.

The posterior distribution of the GP (2) can be used to study when the function f takes values above a certain threshold $T \in \mathbb{R}$. A central tool for this task is the posterior *probability of excursion* $p_\ell(x) = P(f(x) > T \mid \mathbf{y})$, $x \in \mathbb{X}$. This quantity indicates the probability that f exceeds a threshold at an input point x and it can be used as the output probability for a binary classifier. For each testing point x^* we can evaluate $p_\ell(x^*)$ and classify the point as above (resp. below) the threshold T if $p_\ell(x^*) > \gamma$ (resp. $< \gamma$) for a certain discrimination threshold γ. In practice, the value for γ can be chosen with different techniques: we could fix a reference value (e.g. $\gamma = 0.5$) or an adapted value could be chosen by looking at the ROC curve. In this set-up, since the posterior distribution of f is Gaussian, the posterior probability of excursion can be written as

$$p_\ell(x) = \Phi\left(\frac{m_\ell(x) - T}{\sqrt{k_\ell(x,x)}}\right), \tag{3}$$

where m_ℓ and k_ℓ denote respectively the mean and covariance kernel of the posterior distribution of the GP. After training, m_ℓ and k_ℓ have closed-form expressions which are fast to compute for $\ell < 5000$. The function $p_\ell(x)$ will be further used during the AL phase to explore the feature space in order to find instances that lead to near-to-threshold BER values.

C. Active Learning with Gaussian processes

The GP regression model introduced in the previous section is an offline supervised training method: given a training set $(\mathbf{x}_\ell, \mathbf{y}_\ell)$, we build a regression model ($p(\mathbf{f} \mid \mathbf{y})$) and we use it to predict whether the response will be above or below

the threshold T by evaluating $p_\ell(x^*) > \gamma$. We can improve the classification performance of this method by adding new instances to the training set. In AL we achieve this objective by selecting new instances that minimize a specific *acquisition function* based on the current posterior GP distribution.

In this work we are interested in selecting instances that improve the prediction performance of the classification function $\mathbf{1}_{f(x)>T}$ that returns class 1 if the lightpath with features corresponding to the vector x is above threshold T or class 0 otherwise. In the GP regression framework, for each $x \in \mathbb{X}$, $f(x)$ is a random variable, therefore $\mathbf{1}_{f(x)>T}$ is also a random variable with mean $\mathbb{E}_\ell[\mathbf{1}_{f(x)>T}] = P(f(x) > T) = p_\ell(x)$ and variance $p_\ell(x)(1 - p_\ell(x))$. We can then envision a strategy that improves the training set by adding instances in such a way that the future variance of $\mathbf{1}_{f(x)>T}$ is minimized. More precisely, we minimize the acquisition function defined by the integrated variance over $\mathbb{X}$, i.e.

$$J_\ell(x) = \mathbb{E}_\ell \left[\int_{\mathbb{X}} p_{\ell+1}(z)(1 - p_{\ell+1}(z))dz \mid x_{\ell+1} = x \right], \quad (4)$$

where $\mathbb{E}_\ell$ denotes the conditional expectation given the training data $\mathbf{x}_\ell, \mathbf{y}_\ell$ and we are further conditioning the next input point to be $x_{\ell+1} = x$. The acquisition function in (4) was introduced in Ref. [25]. In order to compute the integral in (4), we need to evaluate $p_{\ell+1}$, which is unknown since the $x_{\ell+1}, y_{\ell+1}$ are unknown, however [34] derived the following closed-form formula for Eq. (4)

$$J_\ell(x) = \int_{\mathbb{X}} \Phi_2 \left(\begin{pmatrix} a(z) \\ -a(z) \end{pmatrix}, \begin{pmatrix} c(z) & 1 - c(z) \\ 1 - c(z) & c(z) \end{pmatrix} \right) dz, \quad (5)$$

where $a(z) = (m_\ell(z) - T)/\sqrt{k_{\ell+1}(z,z)}$, $c(z) = k_\ell(z,z)/k_{\ell+1}(z,z)$ and $\Phi_2(\cdot;\Sigma)$ is the c.d.f. of a centered bivariate normal with covariance Σ. The value $k_{\ell+1}(z,z)$ can be computed analytically when evaluating $J_\ell(x)$ without knowing $y_{\ell+1}$ (see [34], [35] for more details).

Other strategies that aim at minimizing other types of uncertainties are also possible: for example, Refs. [25], [34] propose a strategy that minimizes $\mathrm{Var}\left(\int_{\mathbb{X}} p_{\ell+1}(z)dz\right)$ and Refs. [33], [36] develop strategies that minimize the uncertainties on estimates of the whole set $\{x \in \mathbb{X} : f(x) > T\}$.

IV. The Proposed GP-based QoT Estimator

The AL procedure outlined in the previous Section can be adapted to QoT estimation. For this purpose we need: (i) to define a feature space $\mathbb{X}$ where each $x \in \mathbb{X}$ corresponds to a lightpath; (ii) to choose an appropriate GP regression model; (iii) to adapt the acquisition function in (4) to our problem.

A. Feature space

In this work, the GP regression model is trained by using historical BER values associated to 5 lightpath features: number of links traversed by the lightpath, lightpath length, length of the longest traversed link, traffic volume and modulation format. We normalize the 5 input features to the unit hypercube $\mathbb{X} = [0,1]^5$ and we consider the function $f_{BER} : x \in \mathbb{X} \subset \mathbb{R}^5 \to \mathbb{R}$. In order to improve the fit we transform the original values of f_{BER} with the logarithm in base 10. This transformation is applied here purely to increase the fitting power of the regression method, the analysis of the results is computed on the back-transformed values. For this reason, in what follows, we only report back-transformed values for f_{BER} and T.

B. Choice of the GP model

We assume that f_{BER} is a realization of a GP with a prior constant mean function $m(x)$, estimated from the data, and prior covariance kernel $k = k_\theta(x, x')$, where we explicit the dependency of k on some hyper-parameters θ. Moreover, to account for time-varying penalties affecting transmission and for inter-channel crosstalk caused by adjacent lightpaths, we consider the measurement y of f_{BER} as perturbed by a normally-distributed noise, i.e., $y = f_{BER}(x) + \epsilon$, where $\epsilon \sim N(0, \sigma_N^2)$.

The choice of the particular family of kernel k_θ encodes our prior assumptions on f_{BER}. As mentioned in the previous section we do not have strong prior information on the shape of f_{BER} as a function of the 5-dimensional input, therefore we choose standard kernels that only make assumptions regarding the regularity of the function. We consider the kernels Matérn with smoothness parameter $\nu = 3/2$ and $5/2$ and the squared exponential kernel. Note that (see [28], chapter 4) by choosing a Matérn with smoothness parameter $\nu = q + 1/2$, $q \in \mathbb{N}$, we are assuming that our unknown function belongs to C^q, i.e., the space of functions with q continuous derivatives, and with a square exponential that the true function belongs to C^∞, i.e., the space of differentiable functions for all degrees of differentiation.

We fix an initial training set $\mathbf{x}_{\ell_0}$, $\mathbf{y}_{\ell_0}$ of size $\ell_0 > 0$. The initial training set is used to estimate the covariance hyper-parameters θ, by maximizing the likelihood of the GP model (a step automatically done by most GP toolboxes). By plugging in the maximum likelihood estimates for θ we can then compute analytically the posterior mean and covariance. The purpose of this initial step is mainly to provide a good starting point for the active learning phase. In section V we analyze numerically the effect of varying the size ℓ_0.

The trained model provides an estimator for f_{BER} with the posterior GP mean and the estimator for the probability of excursion. Given the ℓ_0 instances of the initial training set, the excursion probability at any $x \in \mathbb{X}$ can be computed with Eq. (3) with $\ell = \ell_0$. Given a discrimination threshold γ, we can classify an instance x as above T if $p_\ell(x) > \gamma$ or below T otherwise. During the AL phase we do not need to choose γ and p_ℓ will only be used within the acquisition function to obtain new instances.

C. Active Learning

The aim of the AL phase is to expand the initial training set with additional instances that exhibit near-to-threshold BER values. As explained in the previous section we select the next training instance $x_{\ell+1}$ by minimizing the *acquisition function* in Eq. (4). The integrand in J_ℓ is the variance of the indicator $\mathbf{1}_{f_{BER}(x)>T}$ and it is linked to the probability of misclassification $\min(p_\ell(x), 1 - p_\ell(x))$ [25]. In particular here

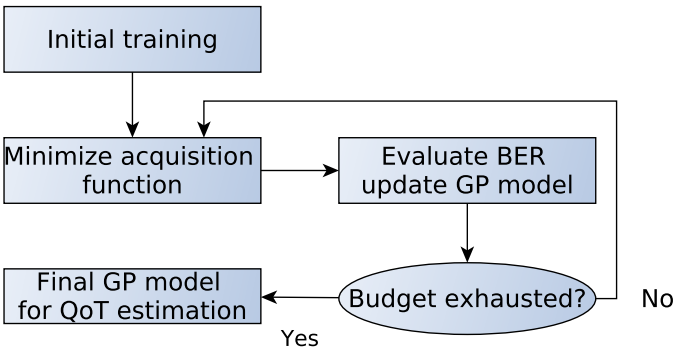

Fig. 1: The active learning framework

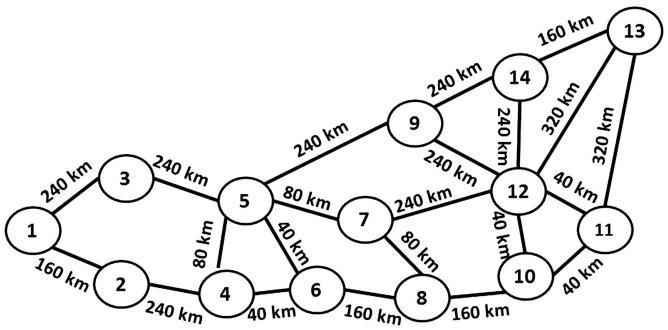

Fig. 2: Japan network topology

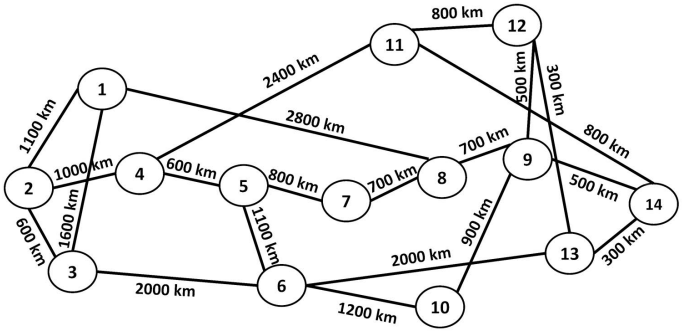

Fig. 3: NSF network topology

we use the semi-analytical form for J_ℓ described in (5). Each evaluation of J_ℓ in this formulation requires the computation of an integral over the input space. We compute the integral with an importance sampling Monte Carlo algorithm [35]. However, note that by using the unconstrained acquisition function as defined in (5), the optimum might be a point in the feature space that does not correspond to any feasible lightpath. For this reason we introduce a constrained version of J_ℓ, here denoted J_ℓ^C, with hard barriers that encode the following constraints:

$$x^{(2)} \geq x^{(3)}$$
$$x^{(2)} \leq x^{(3)} \times x^{(1)}$$
$$x^{(2)} - x^{(3)} \geq (\text{smallest link length in topology}) \times (x^{(1)} - 1)$$

where $x = (x^{(1)}, \ldots, x^{(5)})^T \in \mathbb{R}^5$ with $x^{(2)}$ lightpath length, $x^{(3)}$ length of the longest traversed link and $x^{(1)}$ number of links traversed by the lightpath. By minimizing J_ℓ^C we find the instance $x_{\ell+1}$ coherent with the current network topology that minimizes the integrated probability of misclassification at the next step (ℓ+1). The chosen training instance $x_{\ell+1}$ is first back transformed with a binning procedure that returns only feasible values for the features. Then we can associate a real lightpath on the network topology by taking the path on the graph with the number of links, lightpath length and longest traversed link's length given by the back-transformed values of $x_{\ell+1}^{(1)}$, $x_{\ell+1}^{(2)}$ and $x_{\ell+1}^{(3)}$ respectively. If a ligthpath with such features does not exist on the graph, we select the lightpath with total length and longest link length as close as possible to $x_{\ell+1}^{(2)}$ and $x_{\ell+1}^{(3)}$ among all lightpaths with number of links equal to $x_{\ell+1}^{(1)}$; in case of ties we select randomly among the optimal choices. We assume that this probe lightpath is deployed with traffic volume and modulation format given by the back-transformed $x_{\ell+1}^{(4)}$, $x_{\ell+1}^{(5)}$ respectively and that its BER is measured, so that we can evaluate f_{BER} at point $x_{\ell+1}$ and update the GP model. Note that, since the objective function J_ℓ is constrained to return only lightpath lengths and link lengths within the values allowed by the considered network topology, we can always associate a probe lightpath to a set of features $x_{\ell+1}$. The GP hyper-parameters θ are updated every 10 iterations of the AL procedure in order to reduce the computational cost and to avoid numerical issues associated with frequent hyper-parameter re-estimation. We continue iterating this procedure until either a predefined budget on the number of iterations is reached or the acquisition function value drops below a certain tolerance. The procedure is implemented in the R programming language with the packages DiceKriging [37] and KrigInv [35]. Fig. 1 represents the adopted AL solution in a block diagram.

V. Performance Assessment

We now apply the proposed AL solution for QoT estimation over two realistic network topologies: the Japan and the NSF network, depicted in Fig. 2 and Fig. 3, respectively[2]. We evaluate its performance in terms of AUC. Note that our GP model predicts the BER value, therefore we need to compute p_ℓ and to set a threshold on the probability of excursion to predict whether the BER value is above the threshold T or not. In particular, the AUC is computed by first evaluating the probability of excursion p_ℓ on test data, then, for different threshold levels γ, we can compute the false positive rate (FPR) and the true positive rate (TPR). By plotting FPR versus TPR we obtain the receiver operating characteristics curve (ROC). We can then compute the AUC by evaluating the integral of the ROC numerically. Note that a perfect classification would result in AUC=1, while a completely random classifier achieves AUC=0.5.

A. Dataset generation

To generate synthetic data we use the QTool described in [6]: given as input a candidate lightpath, a traffic amount to be

[2]The code and datasets to reproduce the experiments are available at https://bitbucket.org/darioaz/al_qot_2019/

served, and the modulation format to be adopted for transmission, the QTool calculates the BER as a function of the signal-to-noise ratio (SNR) measured at the input of the channel decoder. The QTool implements the approximated Additive White GN model of dispersion uncompensated transmission over single mode fibers [3], assuming a flexi-grid with 12.5 GHz slice width and elastic transceivers operating at 28 Gbaud with optical bandwidth of 37.5 GHz, using one modulation format among dual polarization (DP)-BPSK, DP-QPSK and DP-n-QAM, with $n = 8, 16, 32, 64$. Traffic demands exceeding the capacity of a single transceiver are accommodated in superchannels containing multiple adjacent transceivers. The Qtool considers transparent links of dispersion uncompensated standard single-mode fibers with 0.2 dB loss per km, where the signal power is restored by identical optical amplifiers equally spaced over the links (100 km), with 20 dB gain and 5 dB noise figure. The QTool also adds randomly-distributed penalties to account for the uncertainty of the model, according to an exponential distribution with average of 2 dB.

For both topologies we fix a threshold $T = 4 \cdot 10^{-3}$ and build the initial training sets with a fixed proportion $\tau \in [0, 1]$ of instances with BER values above T. We consider three settings:

- $\tau = 0$, i.e. all instances have BER values below T. This models a situation where the network has been deployed according to conservative policies (i.e., applying consistent design margins).
- $\tau = 0.05$, in this case the number of probe lightpaths exhibiting above threshold BER already deployed in the network is assumed to be 5% of the total number of monitored ligthpaths (the remaining 95% consists of established lightpaths carrying user traffic).
- $\tau = 0.1$, in this case the number of already deployed probe ligthpaths is assumed to be 10% of the total amount of monitored lightpaths.

Below we provide the specific details of the dataset for the two topologies.

1) Japan topology: We generated an instance by randomly choosing a source-destination node pair and a path connecting them among the 3 shortest paths, a modulation format uniformly sampled among BPSK, QPSK and DP-n-QAM with $n = 8, 16, 32, 64$, a traffic demand uniformly selected in the range $[50-500]$ Gbps with 50 Gbps granularity and evaluating the BER with the QTool [3].

2) NSF topology: The NSF topology has links which are significantly longer than those of the Japan topology, therefore high BER values are more frequently obtained by uniform random sampling of routes, modulation formats, and traffic volumes, especially for configurations with high traffic and highly efficient modulation formats. To limit the number of instances with above threshold BER, the traffic demand was uniformly selected in the range $[50-300]$ Gbps, and 64-QAM was excluded from the set of modulation formats.

For both topologies, the test set was constructed by generating a separate set of $E = 2000$ instances, by randomly selecting lightpaths and evaluating the BER function following the same mechanism adopted for the generation of the training set.

[3]Note that a random selection of the modulation format with uniform distribution captures the following three categories of lightpaths: i) lightpaths adopting the most spectrally efficient modulation format exhibiting a transmission reach that exceeds the lightpath length; ii) lightpaths adopting less aggressive modulation formats, due e.g. to more conservative design approaches; iii) lightpaths adopting modulation formats that are not feasible (i.e., exhibiting transmission reaches shorter than the lightpath length), such as probe lightpaths or mistakenly deployed lightpaths.

B. AUC Evaluation on Japan topology

We consider, for all choices of τ, an initial training set of size $\ell_0 = 100$ and we repeat up to 400 AL iterations, adding one instance at a time. We compare the AL approach to a standard non-active (non-AL) supervised ML approach where training is performed over a dataset of $\ell' = 500$ samples generated by taking the same first 100 samples as in the initial training set of the AL and then by uniformly sampling the remaining 400 instances in such a way that the final proportion τ_{end} of instances above T is equal to the final proportion obtained with the AL procedure. This way, the main difference between the final AL dataset and the randomly sampled dataset resides exclusively in the type of points selected and not in the number of points above the threshold that compose the training sets. We consider three kernel functions k: a Matérn covariance with smoothness parameter $\nu = 3/2$ (Mat32), Matérn with $\nu = 5/2$ (Mat52) and a squared exponential kernel (SE) (see [28], chapter 4). All models use Automatic Relevance Determination (ARD) kernels, which give an indication of the features relevance over the output.

In Figure 4, the AUC values obtained with AL and with random samples are represented by the dotted curves for the three kernel functions described above. Colors and line types denote different kernels. AL is compared to a standard offline ML, whose performance is averaged over 20 runs and shown with an horizontal line (note that the shaded area represents the 90% confidence interval). In all cases, we notice how the AL approach achieves higher values than the 90% upper confidence interval (indicated by the crossing of the curves with the top of the gray region) after a few iterations (less than a hundred). This indicates that: i) an accurate choice of the training samples is key for the classifier predictive capabilities and ii) a limited amount of selected probes can lead to satisfactory results, while saving a large amount of training data.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	total	above T	total	above T	total	above T
Mat32	6.0%	5.8%	12.8%	15.4%	23.0%	28.9%
Mat52	9.8%	11.7%	23.8%	26.5%	24.8%	27.9%
SE	9.8%	11.2%	23.2%	29.5%	27.0%	32.3%

TABLE I: Proportion of instances (total and with BER value above T) required by AL procedure over random sampling to achieve the same AUC value in the Japan topology (absolute values reported in Appendix A).

To highlight the practical value of reducing the number of probes needed to reach a certain targeted AUC, in Table I, column "total", we report the proportion of instances required

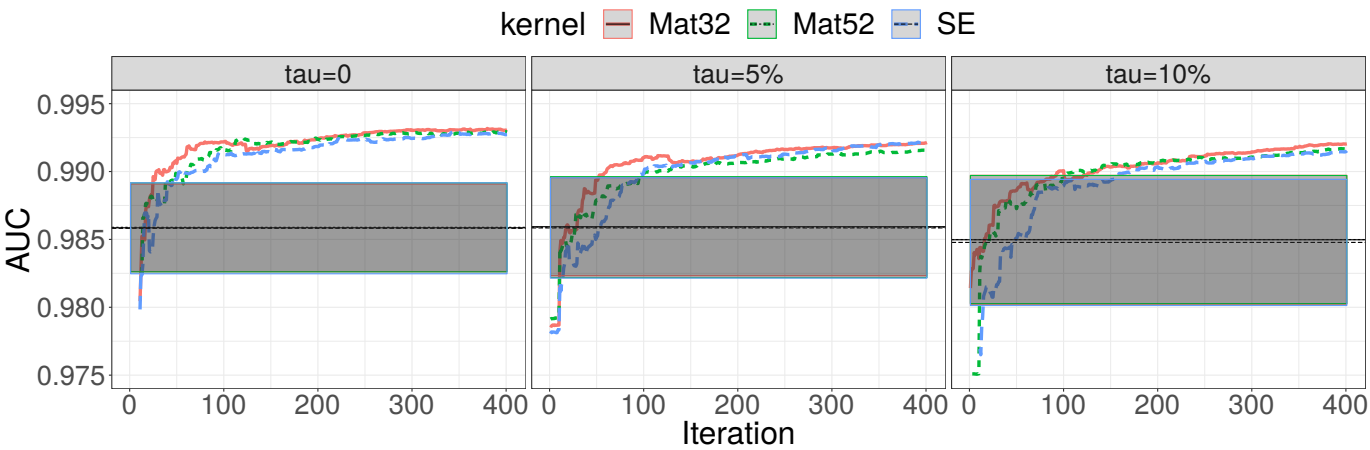

Fig. 4: AUC values versus iteration number for the Japan topology. Performance of offline supervised learning reported as horizontal line.

by the AL approach over the instances required by random sampling to achieve the same AUC values. We say that AL and random sampling achieved the same AUC when the value for AL is equal to the 90% confidence upper bound of the random sampling AUC. When the initial proportion of points above T is smaller, AL requires much fewer instances than random sampling to achieve similar AUC. For example, in the case $\tau = 0$ we notice how AL requires only between 6% and 10% of the instances needed for random sampling. This percentage tends to increase when τ gets larger, while always remaining below 27%. Table I also shows, in column "above T", the proportion of the number of instances with BER above T required by AL over the number of instances with BER above T required by random sampling. This metric is important in practice because BER values above T imply that transmission along those lightpaths violates service level agreements, thus such lightpaths must be used exclusively for probing and cannot carry user generated traffic. While this quantity shows the same behavior as the total number of instances, it is interesting to note that as τ increases the competitive advantage of AL decreases and, even accounting for the initial number of instances above T, AL requires a larger proportion of instances. This seems to indicate that the choice of the instances above T is a key factor to achieve good performance, in fact when many instances above T are chosen randomly ($\tau = 0.1$) we observe that AL requires more instances outdo random sampling. This further reinforces the idea that AL is especially useful at the early stages of the deployment of a new network, when only a limited amount of ligthpaths are installed and monitored.

Both Table I and Figure 4 do not show large differences between the three kernels tested for the GP method. The Matérn kernel with $\nu = 3/2$ shows a faster increase and achieves larger AUC values than the other kernels; Matérn $5/2$ is also slightly better than the squared exponential kernel. This indicates that the function f_{BER} might not be very smooth, with discontinuities in the first or in the second derivatives.

We evaluated the active learning procedure with the AUC metric which is independent of the discrimination threshold chosen. In practice, however, for a given discrimination threshold a high rate for false positives has an important impact on network operations. The acquisition function chosen in the AL procedure forces exploration of the feature space thus also reducing false positives [25]. In Appendix C we show on one representative example that AL reduces on average false positives faster than random sampling as the training set size increase.

C. Impact of initial training set size in the Japan topology

In the previous section the size of the initial training set was chosen large enough to avoid any numerical issue in the optimization of the hyper-parameters for any kernel. We now study the impact of the choice of ℓ_0 (i.e., the size of the initial training set) on the results discussed above, for the Matérn kernel with $\nu = 3/2$.

We consider the dataset introduced in the previous section and we fix a prior constant mean estimated from the data. We conduct the same experiment with $\ell_0 = 25, 50, 75, 100$ and we run the active learning procedure for 200 iterations. We compare the results for $\tau = 0, 0.05, 0.1$. Figure 5 shows the AUC values as a function of the total number of evaluations, i.e. the initial training set plus the AL iterations. In the case $\ell_0 = 100$ we observe that, for all τ, the method achieves high AUC values after a few iterations. On the other hand, in case of a small ℓ_0, such as $\ell_0 = 25$, we see that the starting AUC value is lower and it takes a higher number of iterations to achieve good AUC values. Nonetheless, $\ell_0 = 25$ achieves higher or comparable AUC values to $\ell_0 = 100$ after 225 (25 initial plus 200 AL iterations or 100 initial plus 125 AL iterations) function evaluations.

The results of this experimental study suggest that the size ℓ_0 of the initial training set might not be a key parameter for the procedure as we observe a relatively fast convergence to high AUC values for all choices of ℓ_0. Here, however, we consider synthetic data and an application of the same procedure on real-world measurements might be subject to much higher noise and might require larger ℓ_0 in order to achieve stable results.

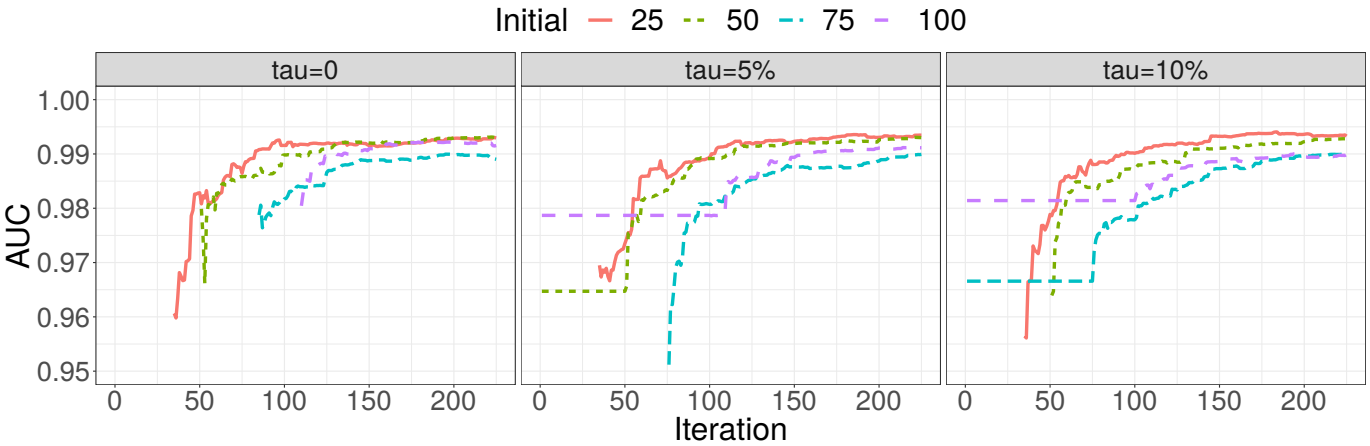

Fig. 5: AUC values versus iteration number, using the Mat32 kernel. Comparison of different initial designs for the Japan topology.

D. AUC evaluation on NSF topology

For the NSF topology, we consider an initial training set of size $\ell_0 = 100$ for each choice of τ and we repeat 500 AL iterations adding one instance at the time to the training set. In this case, we consider a larger number of iterations because the method results in more unstable AUC values. Again, we compare AL with a standard non-AL approach trained on a dataset composed by the same initial training set of the AL method augmented with 500 instances sampled uniformly among all lightpaths compatible with the NSF topology. The final proportion τ_{end} of instances with BER values above T in the random sampling case is equal to the proportion obtained with the AL approach. We consider the same three kernel functions as for the Japan topology: Mat32, Mat52 and SE. Figure 6 compares the AUC values obtained with AL and non-AL. The AUC performance of AL is plotted as a function of the iteration number, the average AUCs obtained from 20 runs of the non-AL approach are reported as horizontal lines and the 90% confidence bound is identified by the shaded area.

In the case $\tau = 0$, we observe that AL crosses the upper 90% confidence bound after as few as 222 iterations. In this case, more iterations than in the Japan topology are needed to achieve similar performance to the non-AL approach. This can be explained as, based on the output of the QTool, we observed that BER values above T in this topology are often much higher than T, thus making the exploration of the feature space for values around the threshold harder. Both the AL and the non-AL approaches then choose rather similar instances, resulting in smaller performance gains of the AL approach compared to the Japan topology. A reduced exploration of the feature space around T (see also results reported in Appendix B for test datasets restricted to instances exhibiting $T > 10^{-5}$) also makes hyper-parameter estimation harder, resulting in AUC performances which are more unstable, as shown by the jumps in AUC values in Figure 6. Table II shows the proportion of instances required by AL to achieve better performance than the non-AL approach. Note that here we considered AL better than non-AL only if the AUC value obtained by AL remained higher than non-AL for all the successive iterations. For example, in the case $\tau = 10\%$ with Mat32 kernel (rightmost plot in Fig. 6, red solid line), AL was considered better than non-AL from iteration 470 and not from the first crossing at iteration 111, because of the oscillations of the AL AUC value around the 90% confidence bound.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	total	above T	total	above T	total	above T
Mat32	44.0%	46.8%	72.8%	75.6%	93.8%	94.2%
Mat52	41.4%	43.2%	84.8%	86.8%	89.8%	91.0%
SE	59.8%	62.4%	93.8%	94.0%	83.8%	86.9%

TABLE II: Proportion of instances (total and with BER value above T) required by AL procedure over random sampling to achieve the same AUC value, when applied to the NSF topology (absolute values reported in Appendix A).

Similarly to the results obtained for the Japan topology, also here we notice that the performance does not vary greatly between different kernels. The Matérn kernel with $\nu = 3/2$ obtains slightly better performance, possibly indicating that the function f_{BER} is not very smooth. More importantly, it is also more stable in the hyper-parameter estimation phase, leading to more stable AUC performances.

As for the Japan topology, AL achieves the best performance gains in the case $\tau = 0$ for all kernels, thus making a strong argument for using AL directly in the early deployment phase of an optical network.

Finally, Table III shows the average time in seconds required for the different operations in the AL procedure for the NSF topology. We report two GP training times: (1) includes the hyper-parameter optimization, which is computed only every 10 iterations, (2) is the GP training time at fixed hyper-parameters. The overall time required by the procedure increases from 40 seconds in iteration number 100 to 203 seconds in iteration number 500.

According to the results reported in this Section, we obtained reductions of at least 73% in the training set size needed to achieve satisfactory classification performance, w.r.t. a standard offline supervised learning approach on the Japan

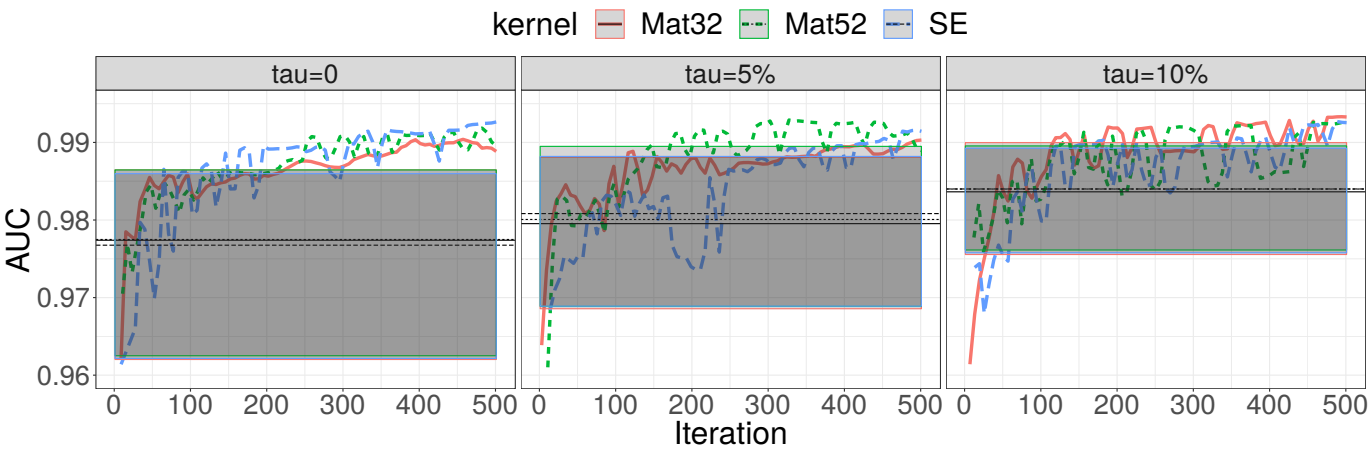

Fig. 6: AUC values versus iteration number achieved by the AL approach applied to the NSF topology. Performance of offline supervised learning as horizontal line.

	$\tau = 0$	$\tau = 0.05$	$\tau = 0.10$
train (1)	30.0 s (7.4 s)	32.0 s (10.9 s)	32.0 s (10.5 s)
train (2)	34.8 ms (14.7 ms)	39.2 ms (22.3 ms)	33.5 ms (14.5 ms)
AL	59.8 s (38.6 s)	56.97 s (41.9 s)	57.42 s (35.8 s)
QTool	12.4 s (4.7 s)	12.9 s (5.2 s)	11.9 s (4.1 s)
test time	0.218 ms ($2.86 \cdot 10^{-2}$ ms)		

TABLE III: Average times in seconds (standard deviation in parenthesis) for the operations required by the AL procedure in the NSF topology.

topology. For the NSF topology, the reduction in training set size is more limited. However, for an initial training set created on a network where consistent design margins were applied (case $\tau = 0$), we still achieve a reduction of around 65%. The different performance in the two network topologies is linked to the behavior of f_{BER} around T. In the Japan topology, probably due to the reduced length of the network links, it is possible to explore the feature space at instances with f_{BER} values close to $T = 4 \cdot 10^{-3}$. In the NSF topology instead, we observe that most of the instances with BER values above T achieve much higher values, for example, a random sampling of 500 instances with 84% of the values above T has a mean BER value equal to 0.10, much higher than T. AL also chooses most instances with BER values much higher than T, thus the performance gains are reduced in this case.

VI. Conclusion

This paper explores how active learning can be used for QoT estimation when the number of lightpaths available to collect BER training instances is limited. Using an active learning approach based on Gaussian Processes, we obtained reductions up to 75% in the training set size needed to achieve satisfactory classification performance, w.r.t. a standard offline supervised learning approach. These results are extremely promising for an effective application of active learning for QoT estimation in network deployments where availability of above threshold probes is scarce, but the extent of the reduction on the amount of required data shall be validated in presence of real field data.

As future work, the use of kernels especially developed for BER function regression could bring substantial improvements, whereas more recent, safe learning objective functions [33] could improve the stability of the results. Moreover, additional features to capture wavelength-dependent effects such as cross-phase modulation and fluctuations in amplifiers gain profiles could be integrated in the learning model.

Acknowledgments

Massimo Tornatore acknowledges funding from the European Community under Grant 761727 Metro-Haul project and from the NSF under Grant 1716945. Dario Azzimonti acknowledges funding from SNSF grant number 167199.

References

[1] R. Proietti, X. Chen, A. Castro, G. Liu, H. Lu, K. Zhang, J. Guo, Z. Zhu, L. Velasco, and S. J. B. Yoo, "Experimental demonstration of cognitive provisioning and alien wavelength monitoring in multi-domain eon," in *Optical Fiber Communication Conference*. Optical Society of America, 2018, p. W4F.7. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=OFC-2018-W4F.7

[2] E. Ip and J. M. Kahn, "Compensation of dispersion and nonlinear impairments using digital backpropagation," *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 20, pp. 3416–3425, Oct 2008.

[3] 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," *IEEE/OSA Journal of Lightwave Technology*, vol. 29, no. 1, pp. 53–61, 2011.

[4] A. Carena, G. Bosco, V. Curri, Y. Jiang, P. Poggiolini, and F. Forghieri, "EGN model of non-linear fiber propagation," *Optics Express*, vol. 22, no. 13, pp. 16 335–62, 2014.

[5] K. Christodoulopoulos, P. Kokkinos, A. Di Giglio, A. Pagano, N. Argyris, C. Spatharakis, S. Dris, H. Avramopoulos, J. Antona, C. Delezoide *et al.*, "Orchestra-optical performance monitoring enabling flexible networking," in *Transparent Optical Networks (ICTON), 2015 17th International Conference on*. IEEE, 2015, pp. 1–4.

[6] C. Rottondi, L. Barletta, A. Giusti, and M. Tornatore, "Machine-learning method for quality of transmission prediction of unestablished lightpaths," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. A286–A297, 2018.

[7] D. Azzimonti, C. Rottondi, and M. Tornatore, "Using Active Learning to Decrease Probes for QoT Estimation in Optical Networks," in *Optical Fiber Communications Conference (OFC) 2019*. IEEE, Mar. 2019, pp. 1–3.

[8] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An overview on application of machine learning techniques in optical networks," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1383–1408, Secondquarter 2019.

[9] 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," *IEEE/OSA Journal of Lightwave Technology*, vol. 31, no. 6, pp. 942–951, Jan. 2013.

[10] I. de Miguel, R. J. Durán, T. Jiménez, N. Fernández, J. C. Aguado, R. M. Lorenzo, A. Caballero, I. T. Monroy, Y. Ye, A. Tymecki *et al.*, "Cognitive dynamic optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A107–A118, Oct. 2013.

[11] E. Seve, J. Pesic, C. Delezoide, and Y. Pointurier, "Learning process for reducing uncertainties on network parameters and design margins," in *Optical Fiber Communications Conference (OFC) 2017*. IEEE, Mar. 2017, pp. 1–3.

[12] S. Oda, M. Bouda, O. Vassilieva, Y. Hirose, T. Hoshida, and T. Ikeuchi, "Network capacity improvement by quality of transmission estimator with learning process," in *2017 European Conference on Optical Communication (ECOC)*. IEEE, 2017, pp. 1–3.

[13] Y. Pointurier, M. Coates, and M. Rabbat, "Cross-layer monitoring in transparent optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 3, no. 3, pp. 189–198, Feb. 2011.

[14] N. Sambo, Y. Pointurier, F. Cugini, L. Valcarenghi, P. Castoldi, and I. Tomkos, "Lightpath establishment assisted by offline QoT estimation in transparent optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 11, pp. 928–937, Mar. 2010.

[15] M. Angelou, Y. Pointurier, D. Careglio, S. Spadaro, and I. Tomkos, "Optimized monitor placement for accurate QoT assessment in core optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 4, no. 1, pp. 15–24, Dec. 2012.

[16] I. Sartzetakis, K. Christodoulopoulos, C. Tsekrekos, D. Syvridis, and E. Varvarigos, "Quality of transmission estimation in WDM and elastic optical networks accounting for space–spectrum dependencies," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 9, pp. 676–688, Sep. 2016.

[17] J. Thrane, J. Wass, M. Piels, J. C. Diniz, R. Jones, and D. Zibar, "Machine learning techniques for optical performance monitoring from directly detected PDM-QAM signals," *IEEE/OSA Journal of Lightwave Technology*, vol. 35, no. 4, pp. 868–875, 2017.

[18] R. M. Morais and J. Pedro, "Machine learning models for estimating quality of transmission in DWDM networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D84–D99, 2018.

[19] W. Mo, Y.-K. Huang, S. Zhang, E. Ip, D. C. Kilper, Y. Aono, and T. Tajima, "Ann-based transfer learning for QoT prediction in real-time mixed line-rate systems," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*. IEEE, 2018, pp. 1–3.

[20] P. Samadi, D. Amar, C. Lepers, M. Lourdiane, and K. Bergman, "Quality of transmission prediction with machine learning for dynamic operation of optical WDM networks," in *2017 European Conference on Optical Communication (ECOC)*. IEEE, 2017, pp. 1–3.

[21] T. Panayiotou, G. Savva, B. Shariati, I. Tomkos, and G. Ellinas, "Machine learning for QoT estimation of unseen optical network states," *arXiv preprint arXiv:1812.07254*, 2018.

[22] T. Panayiotou, S. P. Chatzis, and G. Ellinas, "Performance analysis of a data-driven quality-of-transmission decision approach on a dynamic multicast-capable metro optical network," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 98–108, 2017.

[23] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, Feb 2015.

[24] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization*, C. A. C. Coello, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 507–523.

[25] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez, "Sequential design of computer experiments for the estimation of a probability of failure," *Statistics and Computing*, vol. 22, no. 3, pp. 773–793, 2012.

[26] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statist. Sci.*, vol. 10, no. 3, pp. 273–304, 08 1995.

[27] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, Jan 2016.

[28] C. K. Williams and C. E. Rasmussen, "Gaussian processes for machine learning," *the MIT Press*, vol. 2, no. 3, p. 4, 2006.

[29] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum." in *Towards Global Optimization*, L. Dixon and E. G. Szego, Eds. Elsevier, 1978, vol. 2, pp. 117–129.

[30] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statististical Science*, vol. 4, no. 4, pp. 409–435, 1989.

[31] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[32] V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N.-H. Kim, "Adaptive designs of experiments for accurate approximation of a target region," *Journal of Mechanical Design*, vol. 132, no. 7, p. 071008, 2010.

[33] D. Azzimonti, D. Ginsbourger, C. Chevalier, J. Bect, and Y. Richet, "Adaptive design of experiments for conservative estimation of excursion sets," *arXiv preprint arXiv:1611.07256*, 2018.

[34] C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet, "Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set," *Technometrics*, vol. 56, no. 4, pp. 455–465, 2014.

[35] C. Chevalier, V. Picheny, and D. Ginsbourger, "The KrigInv package: An efficient and user-friendly R implementation of kriging-based inversion algorithms," *Computational Statistics & Data Analysis*, vol. 71, pp. 1021–1034, 2014.

[36] C. Chevalier, D. Ginsbourger, J. Bect, and I. Molchanov, "Estimating and quantifying uncertainties on level sets using the Vorob'ev expectation and deviation with Gaussian process models," in *mODa 10 Advances in Model-Oriented Design and Analysis*, D. Uciński, A. Atkinson, and C. Patan, Eds. Physica-Verlag HD, 2013.

[37] O. Roustant, D. Ginsbourger, and Y. Deville, "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization," *Journal of Statistical Software*, vol. 51, no. 1, pp. 1–55, 2012.

APPENDIX

A. Additional results

In this appendix we report some additional results to complement Section V.

Tables IV and V report the number of instances saved by the AL procedure with respect to a random sampling. The AUC compared are the AL procedure and the 90% upper bound value obtained from 20 random sampling experiments.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	total	above T	total	above T	total	above T
Mat32	376	259	349	236	308	197
Mat52	361	250	305	208	301	204
SE	361	253	307	189	292	193

TABLE IV: Number of instances (total and with BER value above T) saved by AL procedure compared to 90% upper bound for random sampling in the Japan topology.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	total	above T	total	above T	total	above T
Mat32	280	225	136	103	31	25
Mat52	293	239	76	56	51	39
SE	201	161	31	25	81	55

TABLE V: Number of instances (total and with BER value above T) saved by AL procedure compared to 90% upper bound for random sampling in the NSF topology.

B. Near-to-treshold test dataset

Here we also consider a restricted version of the test dataset used in Section V. In order to evaluate the performance of the classifier for instances around the threshold value we consider a dataset where all BER values are greater or equal to $1 \cdot 10^{-5}$. In this case, the AUC values are always smaller than for the full test dataset considered in Section V, see Table VI and table VII.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2
Mat32	0.9932	0.9688	0.9921	0.9692	0.9920	0.9676
Mat52	0.9929	0.9694	0.9916	0.9678	0.9917	0.9671
SE	0.9929	0.9686	0.9922	0.9695	0.9915	0.9665

TABLE VI: Japan topology, best AUC values obtained with AL in full (Test 1) and reduced (Test 2) test dataset.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2
Mat32	0.9923	0.9840	0.9928	0.9821	0.9933	0.9832
Mat52	0.9919	0.9837	0.9928	0.9866	0.9927	0.9844
SE	0.9926	0.9853	0.9918	0.9847	0.9926	0.9844

TABLE VII: NSF topology, best AUC values obtained with AL in full (Test 1) and reduced (Test 2) test dataset.

Tables VIII and IX show the proportion of instances required by AL over a random sampling scheme. The results can be interpreted as explained in Section V for the full test dataset. Also in this case we see the same trends already shown in the main text with a proportion of instances increasing as τ increases. Note that for the restricted test data, the proportions of instances needed by the AL approach to outdo the random sampling approach are not very different than the proportions obtained on the full test dataset. This indicates that the method should perform well also in the extreme case of testing only instances near to the threshold T.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	total	above T	total	above T	total	above T
Mat32	11.2%	12.0%	12.8%	15.4%	23.5%	29.2%
Mat52	15.0%	17.3%	23.5%	26.5%	24.8%	27.9%
SE	12.8%	14.3%	24.8%	31.7%	71.0%	74.3%

TABLE VIII: Proportion of instances (total and with BER value above T) required by AL over random sampling to achieve the same AUC. Japan topology, reduced test dataset.

	$\tau = 0.00$		$\tau = 0.05$		$\tau = 0.10$	
	total	above T	total	above T	total	above T
Mat32	43.6%	46.3%	49.2%	53.1%	50.4%	53.9%
Mat52	41.2%	43.2%	31.0%	35.7%	89.8%	91.0%
SE	59.8%	62.3%	81.8%	83.1%	83.8%	86.9%

TABLE IX: Proportion of instances (total and with BER value above T) required by AL over random sampling to achieve the same AUC value. NSF topology, reduced test dataset.

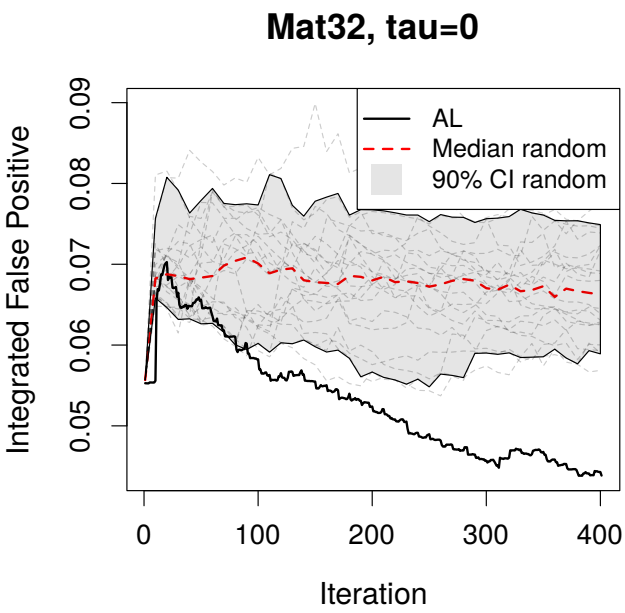

Fig. 7: IFP values versus iteration number, using the Mat32 kernel. Random sampling versus AL for the Japan topology.

C. Study on false positives

Since the occurence of false positive from ML QoT estimator results in the deployment of an unfeasible path (a highly undesirable situation for an operator), in this subsection we empirically show on one example that AL tends to reduce false positives faster than random sampling. First of all note that our classification algorithm is probabilistic, therefore comparing false positive would require choosing a discrimination threshold for each iteration. Since we are comparing models trained on different datasets, there is no unique choice for a discrimination threshold which makes the comparison fair. Therefore we consider the following quantity

$$IFP := \frac{1}{\text{Vol}(\{\text{true false}\})} \int_{\{\text{true false}\}} p_\ell(x) dx$$

which takes values in $[0, 1]$, called here integrated false positive, IFP. In the ideal case of a perfect classification, IFP would be zero as the classifier would predict a positive class with probability 0 for all negative cases, i.e., there are no false positive for any discrimination threshold. On the other hand, in case of a classifier that assigns equal probabilities to both classes, IFP would be equal to 0.5.

Figure 7 shows the integrated false positive value for each iteration of the AL and random procedures. We consider the Japan test case with $\tau = 0$ and Matern kernel with $\nu = 3/2$. The new instances selected by the AL procedure lead to a smaller IFP, while increasing the size of the training set with randomly selected instances does not necessarily decrease it.