Accurate Quality of Transmission Estimation With Machine Learning

Ippokratis Sartzetakis, Konstantinos (Kostas) Christodoulopoulos, and Emmanouel (Manos) Varvarigos

***Abstract*—In optical transport networks the quality of transmission (QoT) is estimated before provisioning new connections or upgrading existing ones. Traditionally, a physical layer model (PLM) is used for QoT estimation coupled with high margins to account for the model inaccuracy and the uncertainty in the evolving physical layer conditions. Reducing the margins increases network efficiency but requires accurate QoT estimation. We present two machine learning (ML) approaches to formulate such an accurate QoT estimator. We gather physical layer feedback, by monitoring the QoT of existing connections, to understand the actual physical conditions of the network. These data are used to train either the input parameters of a PLM or a machine learning model (ML-M). The proposed ML methods account for variations and uncertainties in equipment parameters, such as fiber attenuation, dispersion, and nonlinear coefficients, or amplifier noise figure per span, which are typical in deployed networks. We evaluated the accuracy of the proposed methods under various uncertainty scenarios and compared them to QoT estimators proposed in the literature. The results indicate that our estimators yield excellent accuracy with a relatively small amount of data, outperforming other prior estimators.**

***Index Terms*—Machine learning; Quality of transmission (QoT) estimation.**

I. Introduction

To accommodate continuous traffic growth and dynamicity, coherent transmission and elastic optical networks (EONs) are the main solutions currently being deployed. EONs promise significant benefits, such as higher spectral efficiency, increased capacity, and reduced network costs [1,2]. EONs provide vast optimization dimensions that enable the network to reach higher efficiency levels.

Traditional network planning relies on abundant margins [3,4] to ensure all lightpaths have acceptable quality of transmission (QoT) until the end of life (EoL). Previous studies worked toward classifying the margins and quantifying their impact on network operation [3,4]. Lowering the margins and increasing the efficiency reduces the network costs, motivating various research studies. The authors of [5,6] studied the planning of an EON over multiple periods to take advantage of the evolution of margins over time. Certain connection parameters (e.g., rate) are adjusted at a given time granularity (e.g., 1–2 years) and new equipment is added, when actually needed, according to traffic and physical layer conditions. Therefore, instead of overprovisioning the lightpaths to ensure acceptable QoT until the EoL, the network is operated with just-in-time provisioning. As a result efficiency is increased, equipment purchase is postponed or avoided, and network costs are reduced.

An EON can be operated even more dynamically, as it is possible to adapt the transmission parameters on a shorter timeframe. More specifically, [7] considers dynamic spectrum allocation in order to adapt the network to periodic (e.g., daily) fluctuations of traffic demands. Also, based on future predicted short-term traffic, it is possible to reconfigure the lightpaths to match the expected traffic [8]. Dynamic network operation can also help in efficiently managing failures or QoT degradations. Reference [9] studies the dynamic adaptation of the connections' parameters according to physical layer conditions (e.g., in the case of QoT degradation). Furthermore, [10] investigates a dynamic restoration scheme to minimize recovery time in the case of a failure.

In light of the above, an accurate QoT estimator is the key component for (i) reducing the margins during planning/upgrades [3] and (ii) realizing a more dynamic operation of the network. In the first case, accurate QoT estimation helps reduce the *design* margin [4] that accounts for inaccuracies of the QoT model and the *system* margins that account for expected degradation of performance (aging, increased interference, failure reparations) until the EoL. Consequently, the network can be operated closer to the actual conditions, significantly reducing network costs [5]. In the second case, the QoT estimation of a candidate lightpath is useful in order to adapt the network to current traffic conditions or failures. For example, we envision a network that automatically adapts to traffic changes through reconfiguration; in order to dynamically adjust the spectrum, forward error correction (FEC), and/or the modulation format used by existing lightpath(s), the QoT of the resulting lightpath(s) has to be estimated before these changes are committed.

Manuscript received November 16, 2018; revised January 10, 2019; accepted January 11, 2019; published February 26, 2019 (Doc. ID 352133).

I. Sartzetakis (e-mail: isartz@mail.ntua.gr) and E. Varvarigos are with the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Zografou, Athens, Greece.

K. Christodoulopoulos was with the National Technical University of Athens (NTUA), Zografou, Athens, Greece. Now, he is with Nokia Bell Labs, Stuttgart, Germany.

https://doi.org/10.1364/JOCN.11.000140

A great deal of recent research effort has been directed toward QoT estimation with machine learning (ML). ML has been applied in various areas of optical networks both at the physical and the network layers [11]. In this paper we focus on ML regression, aiming to estimate a specific value for the QoT metric. In doing so, we investigate two possible solutions based on ML techniques. The first approach is to use a machine learning physical layer model (ML-PLM) with specific input parameters (e.g., fiber attenuation, dispersion, nonlinear coefficients) to estimate the QoT. ML is used to learn the input parameters of the PLM and thus improve its accuracy. The second approach is to use a supervised machine learning model (ML-M) as the QoT estimator. In this case, the choice of the features determines to a great extent the accuracy of the estimator. We select the features of the estimators to be appropriate for a heterogeneous network. We evaluate the accuracy of the estimators under various parameter uncertainty scenarios. We demonstrate excellent estimation accuracy with data monitored from a relatively low number of established lightpaths. In a brownfield scenario (where large amounts of monitoring data are available), the estimators can be used in a straightforward way to provision new lightpaths with reduced margins and decrease network costs. In a greenfield scenario, the operator could initially use some probe lightpaths to train the estimator [12] and provision the lightpaths with reduced margins. Another option could be to initially overprovision the lightpaths and at a later stage (next upgrade) train and use the estimators to adjust the transmission parameters accordingly. Studies such as [6] consider this option by assuming a design margin that is reduced over the network lifetime, as traffic increases and more lightpaths are established in the network.

We note here that this paper is an extension of [13]. We substantially extended our previous work: we generalize and also provide more details about the ML methods and the features used. We also introduce new optimization methods that further improve the estimation accuracy, and we present results for an additional network. Finally, we directly compare our results with other previous work.

The rest of the paper is organized as follows. In Section II we present the previous works on the subject. In Section III we introduce the network scenario and the estimation problem addressed. In Section IV we elaborate on the proposed ML techniques. In Section V we present the simulation results, and in Section VI we conclude the paper.

II. Previous Work

ML techniques have been applied at both the physical and network layers of optical networks. At the physical layer, ML is useful due to the complicated modeling of the transmission. Various physical layer models (PLMs) exist that can trade-off complexity for accuracy. The main problem in using PLMs during the planning and operation of the network (e.g., for estimating the QoT) is that they need as input certain physical layer parameters of the network, which are difficult to measure every time that they are needed and so may not be accurately known. Also, the model itself can be inaccurate. Current approaches for network planning assume an ageing network with evolving physical layer conditions and static operation: lightpaths are provisioned to have acceptable QoT until their EoL without having to be reconfigured. Under these assumptions, high (design and system) margins are used to cover physical layer inaccuracies and also to plan for future deterioration. Thus, ML can be used to improve the accuracy of the PLM, understand the actual conditions, and reduce the margins.

At the networking layer, ML can be used to leverage the vast and evolving amount of information in order to make complicated network optimization decisions [11]. For example, ML can be used to (i) anticipate (and predict) demand and traffic changes and (ii) perform proactive virtual network reconfiguration [8] according to those predictions. ML can also be applied to failure identification, localization, and subsequent reconfiguration for recovery purposes [11].

A major advantage of ML in both the physical and network layers is that it can continuously learn and adapt its parameters over time. This is particularly important in an evolving network where equipment ages, (old) equipment is replaced or repaired, and traffic volume and patterns change. Therefore, ML is key for enabling dynamic network operation.

A great deal of effort has recently been directed toward estimating the QoT of lightpaths with ML. Estimating the QoT is a key functionality in an optical network. Estimation methods include solving Schrödinger equations, simulation software (such as VPItransmissionMaker Optical Systems), analytical models of lower complexity, and feedback-based methods in which measurements and monitoring information are correlated to estimate the QoT of new lightpaths. Methods of the last case can be characterized as ML even though they were not reported as such. Most prior works use end-to-end lightpath parameters, e.g., monitoring information from coherent receivers [14], but there are some cases where link or node monitors are used [15]. Moreover, the focus of most prior works (and of this paper) is on the estimation of an end-to-end QoT metric, i.e., the signal-to-noise ratio (SNR) or bit error ratio (BER), of a new or about-to-be-reconfigured lightpath [16–18]. In [17], a neural network is trained to predict the optical SNR of each wavelength division multiplexing (WDM) channel at every node of the network with significant accuracy. However, the heterogeneity of the network is not addressed.

Considering the method used, some papers use ML classification [18,19] to estimate whether a new lightpath is acceptable or not. However, this poses several disadvantages since in real networks there are no failed lightpaths to train the classifier. Also, the classifier may have to be separately trained for different QoT acceptability thresholds (for different modulation formats, FECs, etc.). These disadvantages do not hold for a regressive QoT estimator. The training is performed using a large range of raw QoT values, and the output of the estimator is the QoT metric.

This metric is then compared to the QoT threshold. Moreover, the classification does not provide information about the margin, i.e., the distance from the acceptable QoT threshold. This is a very useful metric for vendors and operators given an evolving network with depleting margins. The respective information can be used to efficiently provision new lightpaths and plan for the network upgrades.

Another important issue is the features used in the ML model to estimate the QoT. One modeling approach is to use a set of end-to-end features, such as the total length of the lightpath, the number of crossed erbium-doped fiber amplifiers (EDFAs), the baud rate, etc. [18]. However, end-to-end features are prone to misrepresenting reality, mainly due to the heterogeneous nature of deployed networks. More specifically, certain parameters, such as span length, EDFA noise figure, and fiber coefficients, vary from span to span. All networks are to a certain degree heterogeneous, even in a greenfield deployment, as there are parameters and performance variations even for the same equipment type due to the manufacturing process. Moreover, different equipment can be deployed in different areas of the network. As the network ages and evolves, heterogeneity typically increases. The reason could be uneven aging due to, e.g., different environmental conditions, equipment replacement/servicing, and fixing fiber cuts. Thus, certain end-to-end features, such as the total lightpath length, provide a limited amount of information toward achieving accurate QoT estimation. For example, two lightpaths of similar lengths may cross spans with different equipment and exhibit significant variation in their QoT. This variation can be explained only in part by some combination of end-to-end features (such as number of crossed EDFAs). However, since there is not any specific information available for the utilized links of the lightpaths, the estimation accuracy will still suffer and be inferior to that of a link-based estimator. The estimation inaccuracy will typically increase as the network ages and equipment variations accumulate. Proper feature definition can avoid such problems, as we will see in the following. Research in [20] demonstrates that the heterogeneity of a network can have an impact on the accuracy of the estimator. The same paper documents previous work focusing on comparison of different ML methods, such as random forest and neural networks.

Another approach to estimate the QoT is to assume a PLM and use real monitoring data to train its parameters and improve its estimation accuracy. This approach was demonstrated in previous works [21,22] to achieve good accuracy. The advantage of this approach is that even with limited or no amount of training, the model is able to provide estimation. A possible general disadvantage of this approach is that the model itself may not represent precisely the behavior of the physical layer. For example, filtering effects may not be taken into account in the model. Such modeling inaccuracies affect the accuracy of the estimation. In any case, the training of the parameters can to some degree absorb certain inaccuracies. More specifically, the training algorithm can tune the parameter values so that the QoT estimation of the (inaccurate) PLM moves toward the real QoT values reported by the optical performance monitors (OPMs). This approach can yield better accuracy than an (inaccurate) PLM using the real parameter values.

A recent survey [11] includes several recent publications on ML in optical networks and on QoT estimation. In this work we build upon previous research, and we investigate various ML formulations in order to implement two accurate QoT estimators designed to account for network heterogeneity. One of them leverages a novel link-based ML formulation. We also quantify the improved accuracy of the proposed QoT estimators over previous work.

III. QoT Estimation Problem

We assume an EON [1] with elastic transceivers that can adapt a number of parameters (modulation format, and/or baud rate, and/or FEC, and/or transmission power). The nodes consist of reconfigurable optical add/drop multiplexers (ROADMs) with flex-grid capabilities connected through uncompensated fiber links. Each fiber link consists of a number of fiber spans that terminate at an EDFA that compensates for the span loss. We assume that there are no spectrum converters and thus a lightpath is allocated the same spectrum throughout its path. For long connections, regenerators are placed. We represent the network by the graph $G = (N, L)$, where N is the set of nodes and L is the set of links. The set of established lightpaths and their attributes (e.g., modulation format) is denoted by P which will also be referred to as the *state* of the network at a given time. Lightpath $p \in P$ crosses a set of links $l_p \subset L$.

We also assume that we can obtain monitoring information from the coherent receivers deployed in the network. Coherent receivers deployed today are packed with DSP capabilities, so they can be easily extended to function as OPMs [14]. An OPM (receiver) can provide information about the SNR with a certain error. The SNR takes into account all impairments, amplified spontaneous emission (ASE) and nonlinear interference (NLI), and residual dispersion. We use this information to improve the QoT estimation accuracy, which in turn can lead to provisioning new lightpaths with reduced margins or dynamically optimize the network. Moreover, assuming an OPM that can report both the ASE noise and the SNR, we can further improve the estimation accuracy, as we will see in the following.

In particular, we denote by $Q^*(p|P)$ the actual value of the SNR for path p when the network is in state P and by $Q^*(P)$ the QoT vector that contains the SNR values of all paths $p \in P$. The mapping $Q^*(P)$ is nonlinear and unknown to us. The set of possible states P is huge for any network of a decent size and thus impossible to measure and record (e.g., in a table) for all possible states. For any set of established lightpaths P we denote by $Y(P)$ the vector of their monitored SNR values, which we assume are available. Note that in the absence of monitoring error, $Y(P) = Q^*(P)$. The monitoring error consists of a systematic error and a random error. The systematic error can be significantly reduced through proper calibration of the monitoring equipment. The random error can be reduced by averaging several measurements over a certain time (on a much shorter timescale than a natural change of the monitored value would occur). As monitoring errors are small and can be reduced through a number of traditional techniques, we ignore them from our presentation here for simplicity of exposition. Thus, we will assume we monitor the true QoT for the paths ($Y(P) = Q^*(P)$ for $p \in P$).

We consider the case where a new lightpath $w \notin P$ is about to be established in the network, and we want to estimate (i) what will the QoT of that new lightpath w be and (ii) how are the QoT metrics of the existing lightpaths $p \in P$ going to be affected by the establishment of w. Put more formally, the problem at hand is the following:

New lightpath QoT estimation problem. We are given the measurements $Y(P) = Q^*(P)$ of the vector of QoT metrics at the current network state P. Our objective is to estimate the new QoT vector $Q^*(P \cup \{w\})$ that will be valid after the new lightpath $w \notin P$ is established.

Note that the above problem definition can be extended in a straightforward way to include a candidate reconfiguration of an existing lightpath or the establishment or reconfiguration of several lightpaths.

Clearly, the new vector $Q^*(P \cup \{w\})$ contains both the (new) QoT metrics $Q^*(p|P \cup \{w\})$ of the existing lightpaths $p \in P$, which will generally be affected by the establishment of w and the (future) QoT metric $Q^*(w|P \cup \{w\})$ of the about to be established lightpath w. We want to obtain these estimates before lightpath w is established because the new connection might render infeasible (in terms of QoT) some of the existing lightpaths. So we want to be able to get such estimates without having to go through the hassle of establishing the new lightpath.

IV. Approximating Architectures

An important issue in ML is the selection of the approximation architecture, that is, the choice of a parametric function that suits the problem at hand. In particular, we are interested in choosing a parametric function $\tilde{Q}(r,P)$ that approximates $Q^*(P)$. Here r is a set (or vector) of parameters of the model (physical layer parameters, like span lengths, attenuation coefficients, etc.). The parametric function $\tilde{Q}(r,P)$ does not have to be a closed form expression; it can also be the output of a computation program or a simulation. What is important is that (i) $\tilde{Q}(r,P)$ approximates $Q^*(P)$ relatively well and that (ii) given the vector r, it is relatively easy computationally to obtain $\tilde{Q}(r,P)$ for the given state P. The following two subsections describe two different approximating architectures for $Q^*(P)$, corresponding to different choices for the family of the parametric function used, one based on a PLM and the second based on feature extraction.

A. Machine Learning Physical Layer Model

The first approach considers a PLM (such as the ones investigated in [21] or the GN model [23]) with some default initial physical layer parameter values for each span (fiber attenuation coefficient, EDFA noise, etc.). In this paper we will use the GN model as the PLM, but our analysis is generic and applicable to other PLMs. Note that we do not take into account filtering effects. The initial parameter values are close to the actual ones but not close enough to provide accurate QoT estimation. They could be based on equipment datasheet values or even field measurements that, however, cannot be carried out continuously, so they would be partially outdated/inaccurate. The objective of ML-PLM is to use monitoring information from the established lightpaths to train and learn the physical layer parameters and therefore improve the accuracy of future lightpath QoT estimations.

When we use the PLM model as an approximating architecture, $\tilde{Q}(r,P)$ is a computation program whose set of parameters r consists of the physical layer parameters, denoted by b, which can include the length, fiber attenuation, dispersion, and nonlinear coefficients per span, the noise figure of each EDFA, its gain, etc. We denote by b_j the jth parameter in b. The model also uses the set of lightpath parameters that are included in the network state P, which can be the route, central frequency, baud rate, modulation format, launch power, etc. of each lightpath. It is important to note that the parameters in P are assumed to be known perfectly, while the parameters in b are not accurately known and have to be estimated. Thus, taking the above into account, we will denote the PLM model as $\tilde{Q}(b,P)$.

Using the monitored $Y(P)$ values, and depending on the PLM, we can use appropriate ML techniques to improve the accuracy of the input physical layer parameters b. Thus, ML-PLM refers to a PLM whose parameters are learned through this process. Regarding the learning algorithm, if we have closed forms for the partial derivatives $\partial\tilde{Q}/\partial b_j$ with respect to all the physical layer parameters b_j of b, we could use the gradient descent method to obtain better estimates of b, as is done in [21]. In our case, where we assume a generic case where the function $\tilde{Q}(b,P)$ is unknown, we can use a nonlinear fitting method. Nonlinear regression is appropriate when the observational data are modeled by a nonlinear function of the model parameters. The data are fitted by successive approximations. Note that an advantage of our proposal when compared to [21,22] is that in our approach we assume that the PLM is a black box. Therefore, our proposal is more generic and flexible and can be used with different PLMs. To estimate b, ML-PLM is given some training data as input in the form of a sequence of measurement pairs $(P, Y(P))$, which are considered representative of the true QoT mapping $(P, Q^*(P))$ that is approximated/estimated by the monitors. In the absence of any measuring noise (discussed previously), we should have $(P, Y(P)) = (P, Q^*(P))$. ML-PLM estimates b so that the *distance* between the measurements $Y(P)$ in the training sequence and the approximating architecture/function output of the ML-PLM $\tilde{Y}(b|P) = \tilde{Q}(b,P)$ is minimized. The iterative approximations start with some default initial physical layer parameter vector b^0 for each span (fiber attenuation coefficient, EDFA noise, etc.). In each iteration i we obtain the updated parameters b^{i+1} that minimize the distance, defined as

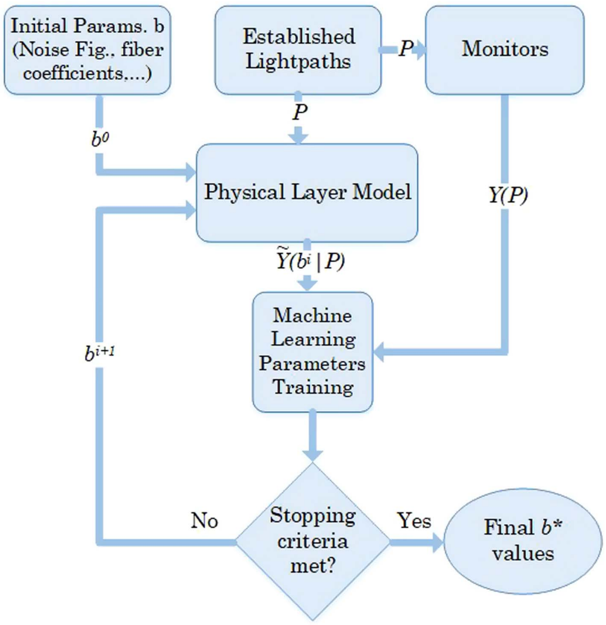

Fig. 1. Block diagram of the ML-PLM.

some function of the error $Y(P) - \tilde{Y}(b^{i+1}|P)$ (e.g., the mean squared error). The iterations stop and we obtain the final b^* values when certain criteria are met; for example, the error function reaches a predefined threshold or the maximum number of iterations are executed. The process is depicted in Fig. 1.

When we want to establish new lightpaths (and/or reconfigure existing ones), we want to estimate their QoT. Thus, we have a new set P', and we estimate $\tilde{Y}(b^*|P') = \tilde{Q}(b^*, P')$ using the obtained physical layer parameters b^*. (In the usual case where only one new lightpath w has to be established, $P' = P \cup \{w\}$). We establish the new lightpath(s), monitor them to get the actual SNR $Y(P') = Q^*(P')$, and obtain the (test) error $Y(P') - \tilde{Y}(b^*|P')$. We then repeat the training with the new network state to further improve the accuracy of b and of the ML-PLM.

B. Machine Learning Model

The second investigated approach tries to obtain an approximation of the actual QoT vector $Q^*(r, P)$ by using feature extraction. This process maps the state P into a matrix $X = f(P)$, called the *features matrix* of P, that is the collection of the feature vectors of all the lightpaths. In this case, and with respect to the generic definition of the estimation function $\tilde{Q}(r, P)$, r is a set of the ML-M parameters (coefficients) denoted by Θ, whose size depends on the particular model [linear regression, neural network, support vector machine (SVM), etc.].

The features matrix X is computed for the established lightpaths in P and is chosen to summarize in a heuristic way the important characteristics of P with respect to the QoT estimation problem. A sample set of features could be end-to-end features, such as the lightpath length, the link load, and the number of crossed EDFAs, which indirectly or directly capture the physical layer parameters. The main idea is that through the change of variables from P to $X = f(P)$, the unknown mapping $Q^*(P)$ of the true QoT metrics can be approximated well by a "relatively smooth" function $\tilde{Q}(\Theta, X) = \tilde{Q}(\Theta, f(P))$. Then a ML-M $\tilde{Q}(\Theta, X)$ is trained with an appropriate regression technique, such as linear regression, neural networks, or SVM. To perform training, we must be given as in the previous section some training data, in the form of a set of pairs $(P, Q^*(P))$.

After implementing and testing several of such methods and several feature extraction functions $f(P)$, we observed that the choice of features is of utmost importance for the estimation accuracy, at least considering medium size optical networks (10–20 nodes) and various traffic loads. Thus, in the following we present a formulation that achieved the best performance.

We organize the feature matrix X so that each row corresponds to one lightpath $p \in P$, while the columns represent (link, feature) pairs, for the links on the lightpath and the values of the chosen features. In particular, we choose a set of three sets of link-level features. We also consider an additional feature to represent the bias term (denoted by BT) [24]. The bias term can account for the remaining monitoring errors that cannot be reduced by any other means. We follow a link-level feature formulation to take into account the heterogeneity of the network. More specifically, the ML estimation algorithm considers each lightpath's link attributes separately during the training process. This results in the algorithm being able to distinguish between two lightpaths that have, for example, the same or similar length but contain different links and therefore possibly exhibit different QoT due to span/link parameter variations. We do not consider span level features, since a lightpath will always cross all the spans of a link.

To be more specific, we defined three sets of link-based features that correspond to the major classes of impairments affecting the QoT. More specifically, we define A as a $|P| \times |L|$ link-level feature matrix designed to account for the ASE noise. Element A_{pl}, corresponding to lightpath p and link l, is set equal to 1 if it contains link l, and is set to zero otherwise. We also define the link-level feature matrix S to account for the self-channel interference (SCI) noise. Element S_{pl} is set equal to the lightpath's baud rate raised to the power of −2 if lightpath p contains link l, and is set to zero otherwise. Finally, we define a link-level feature matrix W to account for the interference of neighboring lightpaths [cross channel interference (XCI)]. The elements of this matrix are derived from an equation that involves the baud rate of the respective lightpath, and for each of its neighboring lightpaths, the distance of the neighboring lightpath and its baud rate, following Eq. (40) of [23].

We concatenate all the link-level feature matrices into one feature matrix $X = f(P)$, defined as

$$X = \begin{bmatrix} BT & A & S & W \end{bmatrix}_{|P| x (3|L|+1)}.$$

Focusing on a lightpath p, the above described features are designed to represent the lightpath's parameters that affect the noise contributors (considering the major types of impairments, both linear and nonlinear) per link. Assume $x_{p,j}$ is the jth feature of lightpath p and that the related noise contribution is denoted by $n_j(x_{p,j}, \Theta)$. Each feature $x_{p,j}$, was designed so that the noise contribution of the related impairment/link depends (close to) linearly on it, i.e., $n_j(x_{i,j}, \Theta) = x_{i,j}\theta_j$, where θ_j is the related impairment/link coefficient.

We assume that the noise of the different types of impairments is additive on a link level and is also additive per impairment (as is the inverse of the SNR). This is an assumption that is made by the GN model, which was demonstrated to achieve good accuracy [23]. This assumption makes possible the correlation of the noise contributions at impairment and link levels. In other words, the total noise accumulated over lightpath p is given by $\sum_j n(x_{p,j}, \Theta)$. Given linear features–noise functions n_j, we can estimate the total noise of all the lightpaths by

$$\tilde{Y}(\Theta, \mathrm{X}) = X \cdot \Theta.$$

Considering that we monitor the SNR for the lightpaths and there is no monitoring error, we set, for lightpath p,

$$Z(p) = \frac{B_N(p) \cdot l_p(p)}{B_r(p) \cdot Q^*(p)},$$

where $B_N(p)$ is the noise bandwidth, $l_p(p)$ is the launch power, $B_r(p)$ is the baud rate, and $Q^*(p)$ is the SNR of lightpath p. Z is the sum of all the noise of all impairments of all the links that the lightpath p crosses, following the assumption of additive noise of the GN model. Given $Y(P)$ (monitoring information) and setting $\tilde{Y}(P) = Z(P)$, we train our model $\tilde{Y}(\Theta, X)$ and obtain the coefficients (thetas) Θ using an appropriate ML training algorithm. Under the assumption of linear features–noise functions $n_j(\tilde{Y}(\Theta, X) = X \cdot \Theta)$, a linear regression/gradient descent algorithm is suitable for learning Θ. So we can iteratively obtain Θ^i until the training error becomes low where we obtain the final Θ^*. The block diagram of ML-M is depicted in Fig. 2.

Note that the ML-M concept is quite generic. We can assume more complicated features–noise functions n_j and more complex and non-additive noise attributes. This would require other ML models, such as neural networks or SVM, and appropriate learning techniques to train our estimator $\tilde{Y}(\Theta, X)$.

Another possibility, under the linear features–noise functions assumption, is to use a constrained least squares solver. The advantage of this method is that we can define additional constraints that the solution must satisfy. For example, we can define constraints that exploit the expected QoT relationship of certain links. For example, a link that is more than 200 km longer than another is highly unlikely to contribute lower noise. The additional constraints help the algorithm to provide better estimations for the thetas, particularly in cases with a small amount of information (lightpaths).

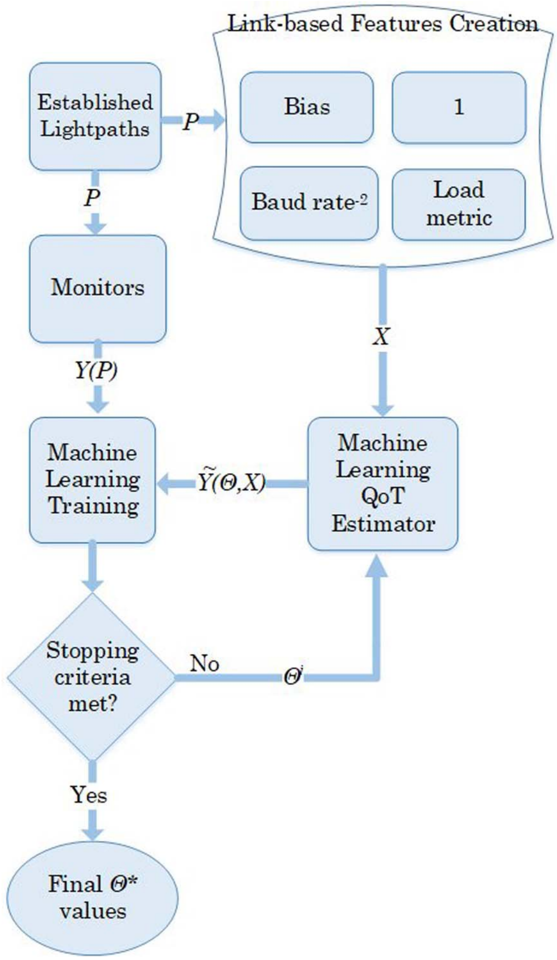

Fig. 2. Block diagram of the ML-M.

The least square solver solves the fitting problem in the form of

$$\min_{\Theta} \|X \cdot \Theta - Y(P)\|_2^2$$
$$\text{such that } E \cdot \Theta \leq 0, \qquad 0 \leq \Theta,$$

where E is the (linear) inequality vector defined as follows:

For all links l_1 in L, for all links l_2 in L, such that $\text{length}(l_1) \leq \text{length}(l_2) + h$, for all sets of features i in $\{A, S, W\}$,

$$\Theta_{i,l_1} - \Theta_{i,l_2} \leq 0,$$

where h is the reference distance (e.g., 200 km), and $\Theta_{i,l}$ corresponds to the parameter of link l in the set of features i. Note also that we constrain Θ to be positive, since these parameters correspond to positive noise values.

If more detailed monitored parameters are available, we can appropriately extend our model to include those. For example, assuming OPMs that can report the ASE in addition to the SNR, we can individually map to the ASE metrics the feature matrix that corresponds to the ASE noise and also *jointly* exploit the SNR measurements, yielding

$$\tilde{Y}(X|P) = \begin{bmatrix} Z \\ ASE \end{bmatrix}_{2|P|x1},$$
$$X = \begin{bmatrix} BT_1 & A & S & W \\ BT_2 & A & 0 & 0 \end{bmatrix}_{2|P|x(3|L|+1)}.$$

Thus, in that case, the algorithm will have more information to calculate the respective thetas. The above formulations rely on monitoring and ML to approximate the behavior of the physical layer without complicated analytical functions.

V. Performance Results

We evaluated the proposed QoT estimation architectures through simulation experiments. We assumed the DT topology with 12 nodes and 40 bidirectional links (Fig. 3) and the NSFNET topology with 14 nodes and 22 bidirectional links (Fig. 4). We assumed four traffic loads of 100, 200, 300, and 400 total connections with uniformly chosen sources–destinations and random baud rates from the set {32, 43, 56}.

Regarding the physical layer, we assumed SSMF fiber with mean attenuation coefficient 0.22 dB/km, mean dispersion coefficient 16.7 ps/nm/km, and mean nonlinear coefficient 1.3 1/W/km. We also assumed a span length of 80 km and an EDFA noise figure of 5 dB. We set the launch power at 0 dBm, and we operate in the linear regime. For the ML-PLM case, the b vector consisted of the attenuation coefficient, the dispersion coefficient, and the nonlinear coefficient. For the ML-M case, we used the features described in Section IV.B. The aforementioned *mean* values also serve as the initial conditions b^0 in the ML-PLM estimation algorithm. The actual (unknown) values of these

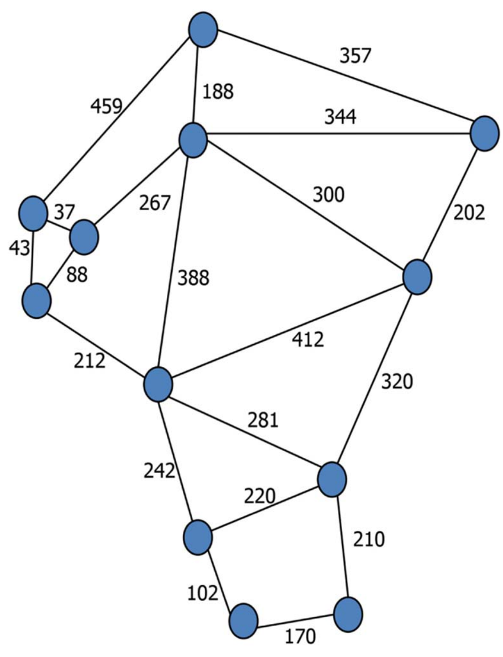

Fig. 3. DT topology with the link lengths in kilometers.

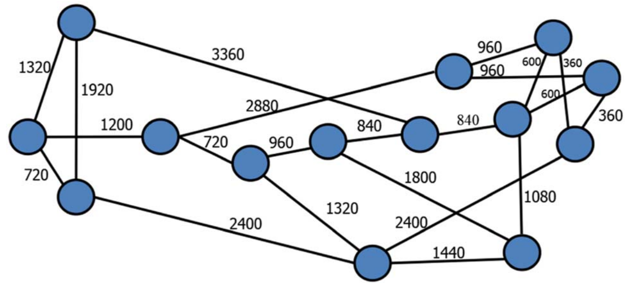

Fig. 4. NSFNET topology with the link lengths in kilometers.

coefficients are in fact drawn in the simulations according to a uniform random uniform distribution with these means. In particular, we examined the following uncertainty scenarios: (a) the attenuation coefficient of all spans is constant ($u_a = 0$) and so are the nonlinear and dispersion coefficients ($u_{n\&d} = 0$); (b) the attenuation coefficient is constant ($u_a = 0$), while the dispersion and nonlinear coefficient vary uniformly by 20% $u_{n\&d} = 0.2$; and (c) all three fiber parameters vary ($u_a = u_{n\&d} = 0.2$). In the case of the ML-PLM, we initialized the parameters in b to the mean coefficient values, and we used the GN model with the varied (random and thus unknown) values as the ground truth producing the training sequence. The estimation objective we considered was the SNR.

For each traffic load and physical layer parameter uncertainty setting, we executed 300 iterations. We used the following three metrics to evaluate estimation accuracy: the mean squared error (MSE) and the maximum overestimation and minimum underestimation over the performed iterations. The maximum overestimation is a useful metric because it gives us the QoT margin that has to be used to always be on the safe side in provisioning the network (never overestimate the QoT so that we will not establish an inappropriate lightpath).

Training of ML-PLM was done with nonlinear regression and, more specifically, we used the function lsqcurvefit of MATLAB. For the ML-M we evaluated neural networks, linear regression, and a constrained least square solver. We used 80% of the lightpaths for training, 10% for validation, and 10% for testing. In all the traffic loads, we exclude from the testing set any lightpaths that include links for which we have no QoT information at all. Regarding the different algorithms that we evaluated for the ML-M, we noticed that all of them provided similar accuracy. The constrained least squares solver provided slightly better results since it had the advantage of the additional constraints. The specific constraints that we used were length similarity constraints for all the links (set at 200 km). The disadvantage of the neural network algorithm is that it is prone to overfitting, so its parameters (number of neurons, dropout rate, etc.) should be carefully chosen, and their values may differ depending on the specific traffic and network. Also, the neural network algorithm typically requires more time for the training phase than the other algorithms. In all the results described in the following, we use a constrained least squares solver for the ML-M.

We also obtained results for other types of features that use only end-to-end parameters of the lightpaths as opposed to the link-level that our model uses. Our objective was to develop the best possible end-to-end estimator and compare it with the link-level estimators. The set of features we considered is similar to the one used in [18]. More specifically, we assumed an end-to-end machine learning model (E-ML-M) with the following features X for each lightpath: number of EDFAs, number of hops, total path length, baud rate used, and a load metric. The load metric is similar to that of ML-M but aggregates the link metrics into one metric since E-ML-M considers only end-to-end features.

In Figs. 5 and 6 we present the results for the DT network. In Fig. 5 we can see that both ML-PLM and ML-M achieve very good mean squared error. The ML-M has consistent performance for all the aforementioned coefficient uncertainty scenarios (a), (b), and (c) simulated, since it does not assume any previous knowledge of the physical layer parameters. Also, the selected features X ensure that the performance will remain consistent regardless of the variations of the parameters in each span. We note that as the sources of uncertainty increase, the ML-PLM needs more lightpaths (which serve as training data) to achieve the same accuracy. This is expected since, when there are no variations, the ML-PLM has previous knowledge of the input/physical layer parameters that is very close to the real values and the fitting algorithm can more easily find the suitable solution. Note that the reported accuracy of ML-PLM in [13] is worse than the one reported here, because in [13] we did not consider independent fitting of all the coefficients and also we used different stopping criteria for the algorithm (e.g., optimality tolerance). ML-PLM is more accurate than ML-M. One reason for this is that ML-PLM has perfect knowledge of the PLM, in the sense that the ground truth function $Q^*(P)$ has been taken to be similar to the GN model function $\tilde{Q}(b,P)$ and needs to fit only the parameters b that are unknown. ML-M, on the other hand, has no such *a priori* advantage and relies only on the monitoring metrics and the features in order to learn to estimate the QoT.

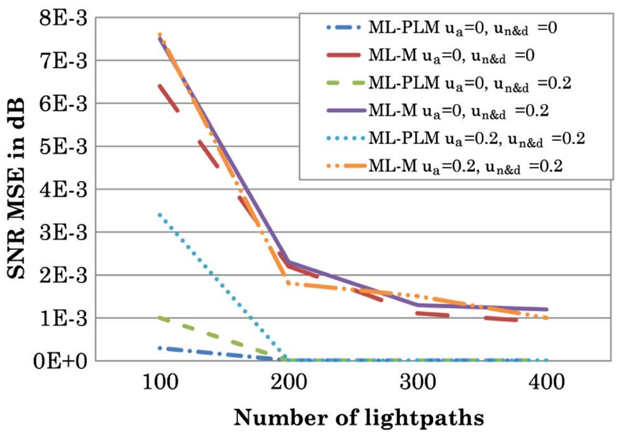

Fig. 5. MSE in dB for the ML-M and ML-PLM (DT network).

In Fig. 6 we can see the maximum deviations of the two algorithms. We report two metrics: maximum overestimation and minimum underestimation of the SNR. ML-PLM's deviations are of the order of 10^{-2} dB. As in the case of the MSE, the ML-PLM needs more lightpaths to achieve the same accuracy when the uncertainties of the parameters increase. The ML-M performs relatively well with a small amount of training lightpaths, and it is able to significantly reduce the error for a high number of lightpaths (≥300). The ML-M's maximum overestimation is of the order of 0.2 dB, while the minimum underestimation is 0.7 dB. If we consider lightpaths that cross more than one link, then the maximum deviation is 0.3 dB when 400 lightpaths are used for training. Note that the maximum overestimation (0.2 dB) corresponds to the design margin that an operator should employ to never provision a lightpath with unacceptable QoT. It is worth noting that there is an asymmetry with respect to the underestimation and overestimation values of ML-M. This could be explained by the constraints $(0 \leq \Theta)$ that tend to estimate lower SNR values, leading to relatively lower maximum overestimation than underestimation. Note that this asymmetry is to our favor, since we need to avoid the overestimation error.

If we take into account that a reference value for the design margin is approximately 2 dB [3], then the benefit in margin reduction that ML-M can provide is 1.8 dB, while for ML-PLM it is approximately 1.95 dB. The improved

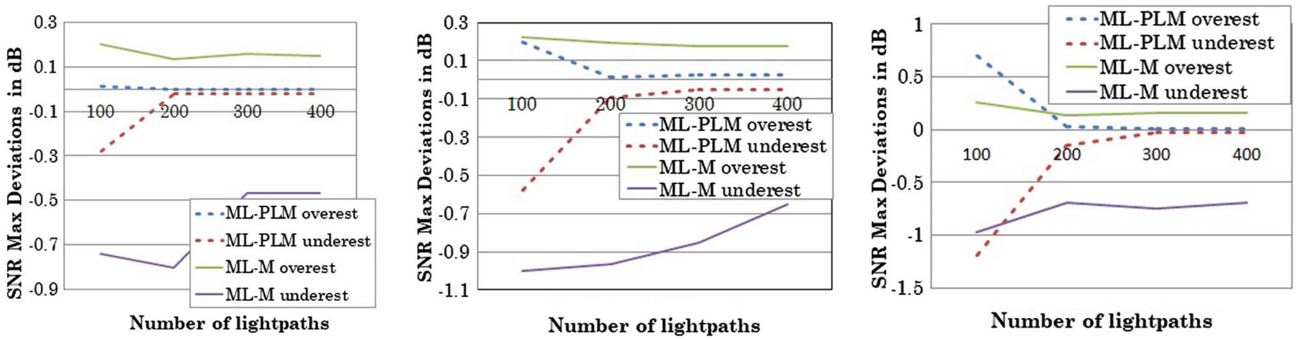

Fig. 6. Maximum deviations for ML-M and ML-PLM for (a) $u_a = 0$, $u_{n\&d} = 0$; (b) $u_a = 0$, $u_{n\&d} = 0.2$; and (c) $u_a = 0.2$, $u_{n\&d} = 0.2$ (DT network).

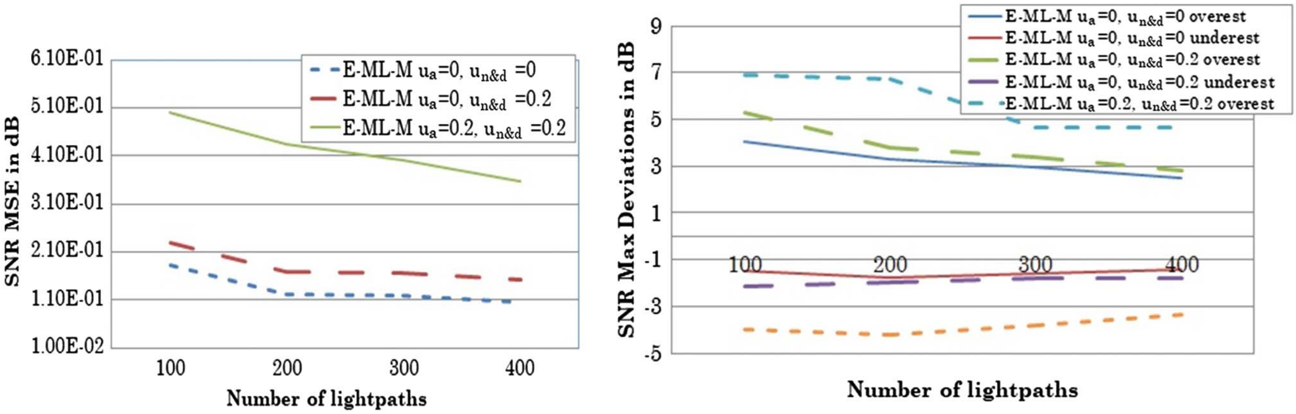

Fig. 7. (a) MSE and (b) maximum deviations of the SNR for E-ML-M with end-to-end features (DT network).

estimation accuracy has been shown to translate to significant equipment savings [6,16].

In Fig. 7 we present the accuracy of the E-ML-M estimator that used end-to-end features. The results are similar to those reported in [20]. We notice that even in the case where there are no parameter uncertainties, the accuracy is worse than that of our proposal. One reason is that in E-ML-M the SNR is estimated indirectly using relatively high level end-to-end networking features (e.g., based on number of hops, total length). Our formulation breaks down the SNR to its three components, ASE noise, SCI, and XCI, and provides relevant features for each one of them. Another reason is that the end-to-end features cannot capture subtle differences between the links of a lightpath. For example, E-ML-M cannot capture the exact load of each link of a lightpath (which seems from the results to play an important role), but calculates the mean load of all the links the lightpath contains. As the parameter variations increase, we notice considerably worse accuracy. As we have mentioned previously, this is due to the use of end-to-end features that provide a limited amount of information in a heterogeneous network. This validates the importance of using features that capture the main dependencies and nonlinearities, by incorporating our prior knowledge or intuition on the QoT estimation problem.

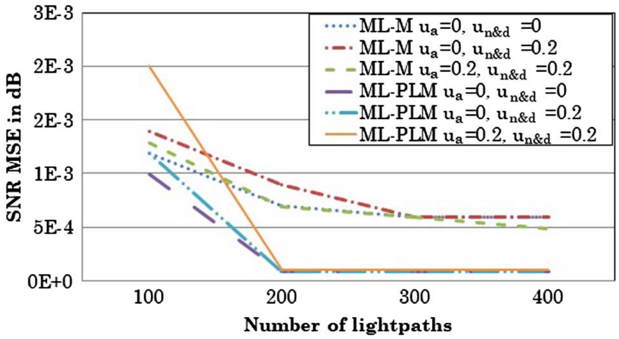

Fig. 8. MSE in dB for the ML-M and ML-PLM (NSFNET network).

In Figs. 8 and 9 we present the ML-PLM and ML-M performance for the NSFNET network. The results and the conclusions that can be derived are similar to those obtained for the DT network. In all cases the accuracy of the estimators is a bit better than in DT. One reason is that the NSFNET has longer links. Therefore, the SNR of the lightpaths is typically lower and less prone to estimation errors. Also, NSFNET has almost half the number of links of the DT network, implying that the estimation algorithm requires a smaller number of lightpaths to obtain good estimation accuracy. It is also important to note that the ML-PLM's running time can be significantly higher for the NSFNET than for the DT topology (tens of minutes for NSFNET as opposed to a couple of minutes for DT using

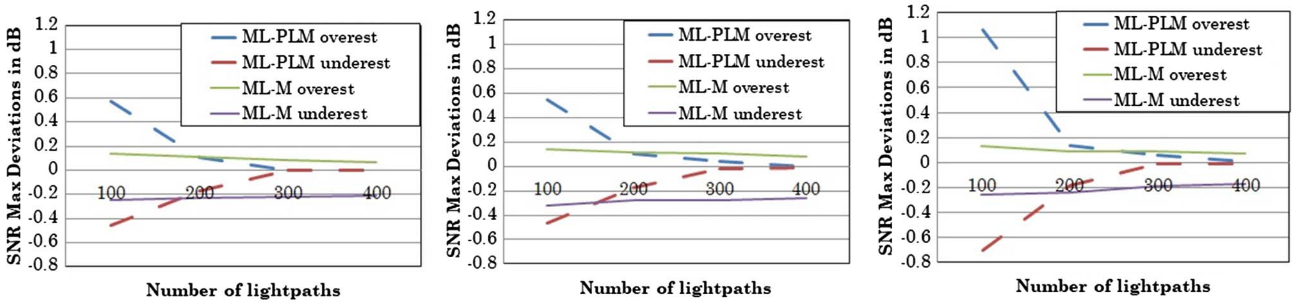

Fig. 9. SNR maximum deviations for ML-M and ML-PLM for (a) $u_a = 0$, $u_{n\&d} = 0$; (b) $u_a = 0$, $u_{n\&d} = 0.2$; and (c) $u_a = 0.2$, $u_{n\&d} = 0.2$ (NSFNET network).

MATLAB and a quad core CPU at 4 GHz). The reason is that the longer links of the NSFNET (which correspond to more spans) are more difficult to learn through training. To reduce the complexity, we assumed a simplified topology of the NSFNET, where each link consisted of spans of identical parameters. In that case, the number of parameters requiring estimation was significantly reduced, and the running time was thereby reduced to a couple of minutes for the NSNET (and about 30 s for the DT). The running time of the ML-M is not affected by the length of the links since it considers only the end-to-end metrics and the per-link features. It requires approximately 0.2 s up to a couple of seconds. Note that these running times refer to the training phase only. The subsequent estimations require much less time to obtain (<0.1 s) and are suitable for dynamic (re)configuration scenarios.

VI. Conclusion

We studied two machine learning (ML) approaches for QoT estimation: the analytical physical layer model (ML-PLM) and the machine learning model (ML-M). We presented a novel ML-M formulation that accounts for the heterogeneous nature of a network. Both models achieve high accuracy using monitoring information from few established lightpaths. ML-PLM is more accurate than ML-M but requires a larger running time for training purposes. ML-M is more flexible in that it does not require any previous physical layer knowledge and is applicable to a large variety of networks (core, metro, dispersion compensated, etc.) with small or no modifications. Both approaches performed much better than ML algorithms that use end-to-end features. The high accuracy of the proposed models makes them suitable for a dynamic network that continuously evolves. Also, network costs can be significantly reduced, since the operating margins can be reduced through more accurate QoT estimation. Future work includes the application of similar ML techniques in other use cases and the extension of the algorithms to take into account filtering effects.

Acknowledgment

The research of I. Sartzetakis was co-financed by Greece and the European Union (European Social Fund—ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY).

References

[1] V. Lopez and L. Velasco, Eds., *Elastic Optical Networks: Architectures, Technologies, and Control*, Springer, 2016.

[2] O. Rival and A. Morea, "Cost-efficiency of mixed 10-40-100 Gb/s networks and elastic optical networks," in *Optical Fiber Communication Conf.*, Los Angeles, California, Mar. 2011.

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

[4] J. L. Auge, "Can we use flexible transponders to reduce margins?" in *Optical Fiber Communication Conf.*, Anaheim, California, Mar. 2013.

[5] J. Pesic, T. Zami, P. Ramantanis, and S. Bigo, "Faster return of investment in WDM networks when elastic transponders dynamically fit ageing of link margins," in *Optical Fiber Communication Conf.*, Anaheim, California, Mar. 2016.

[6] P. Soumplis, K. Christodoulopoulos, M. Quagliotti, A. Pagano, and E. Varvarigos, "Multi-period planning with actual physical and traffic conditions," *J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A144–A153, Jan. 2018.

[7] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Time-varying spectrum allocation policies and blocking analysis in flexible optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 1, pp. 13–25, Jan. 2013.

[8] D. Rafique and L. Velasco, "Machine learning for network automation: overview, architecture, and applications [Invited Tutorial]," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D126–D143, Oct. 2018.

[9] I. Sartzetakis, K. Christodoulopoulos, and E. Varvarigos, "Cross-layer adaptive elastic optical networks," *J. Opt. Commun. Netw.*, vol. 10, no. 2, pp. A154–A164, Feb. 2018.

[10] A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Dynamic restoration with GMPLS and SDN control plane in elastic optical networks [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 2, pp. A174–A182, Feb. 2015.

[11] 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," arXiv:1803.07976, Dec. 2018.

[12] I. Sartzetakis, K. Christodoulopoulos, and E. Varvarigos, "Improving QoT estimation accuracy through active monitoring," in *19th Int. Conf. on Transparent Optical Networks*, Girona, Spain, July 2017.

[13] I. Sartzetakis, K. Christodoulopoulos, and E. Varvarigos, "Formulating QoT estimation with machine learning," in *European Conf. On Optical Communication*, Rome, Italy, Sept. 2018.

[14] 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*, Budapest, Hungary, July 2015.

[15] A. P. Vela, B. Shariati, M. Ruiz, F. Cugini, A. Castro, H. Lu, R. Proietti, J. Comellas, P. Castoldi, S. J. B. Yoo, and L. Velasco, "Soft failure localization during commissioning testing and lightpath operation," *J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A27–A36, Jan. 2018.

[16] I. Sartzetakis, K. Christodoulopoulos, C. P. Tsekrekos, D. Syvridis, and E. Varvarigos, "Quality of transmission estimation in WDM and elastic optical networks accounting for space–spectrum dependencies," *J. Opt. Commun. Netw.*, vol. 8, no. 9, pp. A676–A688, Sept. 2016.

[17] 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 *European Conf. on Optical Communication*, Gothenburg, Sweden, Sept. 2017.

[18] L. Barletta, A. Giusti, C. Rottondi, and M. Tornatore, "QoT estimation for unestablished lightpaths using machine learning," in *Optical Fiber Communication Conf.*, Los Angeles, California, Mar. 2017.

[19] S. Aladin and C. Tremblay, "Cognitive tool for estimating the QoT of new lightpaths," in *Optical Fiber Communication Conf.*, San Diego, California, Mar. 2018.

[20] R. M. Morais and J. Pedro, "Machine learning models for estimating quality of transmission in DWDM networks," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D84–D99, Oct. 2018.

[21] E. Seve, J. Pesic, C. Delezoide, S. Bigo, and Y. Pointurier, "Learning process for reducing uncertainties on network parameters and design margins," *J. Opt. Commun. Netw.*, vol. 10, no. 2, pp. A298–A306, Feb. 2018.

[22] M. Bouda, S. Oda, O. Vassilievam, M. Miyabe, S. Yoshida, T. Katagiri, Y. Aoki, T. Hoshida, and T. Ikeuchi, "Accurate prediction of quality of transmission based on a dynamically configurable optical impairment model," *J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A102–A109, Jan. 2018.

[23] P. Poggiolini, "The GN model of non-linear propagation in uncompensated coherent optical systems," *J. Lightwave Technol.*, vol. 30, no. 24, pp. 3857–3879, Dec. 2012.

[24] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006, Chap. 3.