Deep Neural Network-Based QoT Estimation for SMF and FMF Links

M. A. Amirabadi, M. H. Kahaei, S. A. Nezamalhosseini, F. Arpanaei, and A. Carena

***Abstract*—Quality of transmission (QoT) estimation tools for fiber links are the enabler for the deployment of reconfigurable optical networks. To dynamically set up lightpaths based on traffic request, a centralized controller must base decisions on reliable performance predictions. QoT estimation methods can be categorised in three classes: exact analytical models which provide accurate results with heavy computations, approximate formulas that require less computations but deliver a reduced accuracy, and machine learning (ML)-based methods which potentially have high accuracy while maintaining at same time very low complexity. To operate an optical network in real-time, beside accurate QoT estimation, the speed in delivering results is a strict requirement. Based on this, only the last two categories are candidates for this application.**

In this paper, we present a deep neural network (DNN) structure for QoT estimation considering both regular single-mode fiber (SMF) and future few-mode fiber (FMF) proposed to increase the overall network capacity. We comprehensively explore ML-based regression methods for estimating generalized signal-to-noise ratio (GSNR) in partial-load SMF and FMF links. Synthetic datasets have been generated using the enhanced Gaussian noise (EGN) model. Results indicate that the proposed DNN-based regressor can provide high accuracy in terms of root mean square error, and requires less computation complexity, compared with other state-of-the-art methods such as extended gradient boosting regressor and closed-form-EGN.

***Index Terms*—Deep neural network, single-mode fiber, few-mode fiber, quality of transmission estimation, regression.**

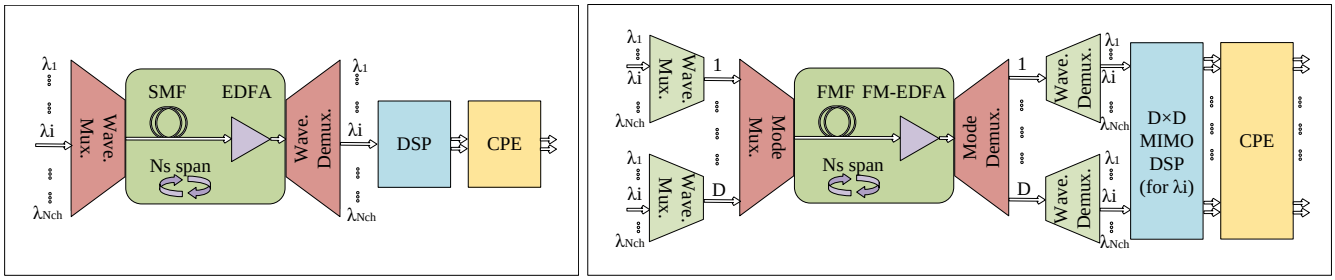
I. Introduction

Single-mode fiber (SMF) communication systems are achieving their theoretical capacity limits due to nonlinear effects [1]. Few-mode fiber (FMF) can significantly increase the capacity of optical networks by combining mode division multiplexing (MDM) with wavelength division multiplexing (WDM) techniques [2], [3]. The quality of transmission (QoT) estimation in SMF and FMF links has crucial importance for optimizing optical network design. Reconfigurable networks need a fast and accurate prediction of lightpaths performance to allow the centralized control and to act in an optimized way. Beside the accumulation of the optical amplified spontaneous emission (ASE) noise introduced by Erbium doped fiber amplifiers, nonlinear effects must be considered as they are dominant in fiber propagation. For this reason, nonlinear interference (NLI) noise estimation in SMF and FMF links is an important aspect in QoT prediction. The NLI noise can be estimated by exact analytical models, e.g. enhanced Gaussian noise (EGN) model [4], [5], or by approximate analytical models, e.g. closed-form (CF)-EGN model [6], [7]. The first option provides accurate results with high computational complexity, and the second choice is faster but less accurate.

Machine learning (ML) has recently been proposed as an alternative approach for QoT estimation in SMF links and can overcome the above-mentioned disadvantages [8]-[20]. In [8] the performance of a data-driven QoT model is investigated in a dynamic metro optical network that supports both unicast and multicast connections. The authors of [9] used ML for the evaluation of optical performance or more generally, to achieve a cognitive network awareness. In [10], ML is used for improving the accuracy of modeling nonlinear impairments on a per-link basis. The authors of [11] deployed different ML methods as regressors to estimate the penalties due to erbium-doped fiber amplifier gain ripple and filter spectral shape uncertainties at the re-configurable add/drop nodes. In [12], two ML-based regression methods are presented for QoT estimation by taking into account fiber attenuation, dispersion and nonlinear coefficients together with amplifier noise figure per span. ML is used in [13]-[15] as classifier to predict if lightpaths satisfy bit error rate requirements. In [16], different ML methods are used as regressors for predicting generalized signal to noise ratio (GSNR) considering full-load links. In [17], authors propose an ML-based QoT estimator which uses precomputed self channel interference values of each WDM channel as feature and total NLI for all channels as labels. An artificial neural network is used as regressor for QoT estimation in [18] in the presence of uncertainty on span lengths, and in [19] considering unestablished lightpaths in a live network with production channels. Authors of [20] report the performance of different ML-based QoT predictors including deep neural network (DNN) for unestablished lightpaths in agnostic optical networks.

In this paper, we propose a DNN-based regressor to estimate the GSNR. The GSNR accounts for the ASE noise introduced by the presence of optical amplifiers and the NLI noise generated by nonlinear effect. In particular, this is the first study on QoT estimation for FMF systems based on joint MDM-WDM. Furthermore, we present a comprehensive investigation and compare the proposed DNN-based regressor with other well-known ML-based regressors as well as the CF-EGN model. The rest of this paper is organized as follows: Section II presents the considered SMF and FMF propagation models for the analyzed links, Section III describes the ML-

M. A. Amirabadi, M. H. Kahaei, and S. A. Nezamalhosseini are with the School of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, 1684613114 Iran, F. Arpanaei was with Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, 10129 Torino, Italy, he is now with Department of Telematic Engineering, Universidad Carlos III de Madrid, 28911, Leganes, Madrid, Spain, and A. Carena is with Politecnico di Torino, Dipartimento di Elettronica e Telecomunicazioni, 10129 Torino, Italy, e-mail: m_amirabadi@elec.iust.ac.ir, kahaei@iust.ac.ir, nezam@iust.ac.ir, farhad.arpanaei@uc3m.es, andrea.carena@polito.it.



(a) (b)

Fig. 1. The considered link setup for: a) SMF, and b) FMF.

based QoT estimation models, and Section IV describes the generation of the synthetic dataset. Section V discuss the optimization of the dataset by adjusting the size and selecting only relevant features. Section VI provides the model selection and performance comparison between proposed DNN-based regressor and other ML-based methods. Section VII is the conclusion of this paper.

II. Propagation Models and Link Description

The considered SMF and FMF setups are respectively depicted in Figs. 1(a) and 1(b). The transmitted signal on SMF links is a combination of N_{ch} polarization multiplexed (PM)-WDM channels. In FMF links we have a further level of multiplexing based on D spatial modes. On each mode (a single one for SMF) we consider the propagation of a WDM comb where each channel has a rectangular spectrum (ideal Nyquist shaping with roll-off set to zero). The analyzed link has N_s spans, an amplifier at the end of each span compensates for the fiber attenuation. The signal propagation suffers from both linear and nonlinear effects including chromatic (modal) dispersion, nonlinear Kerr-effect, as well as modal linear and nonlinear coupling. The received signal is ideally demultiplexed and it enters a digital signal processing section for compensating all linear effects. The nonlinear phase rotation is also assumed to be recovered by a carrier phase estimator (CPE) [5].

The received signal of nth channel and pth mode, after the CPE, can be modeled as the sum of the original transmitted signal plus two interfering terms: ASE and NLI noise [4]-[5], both modeled as additive Gaussian noise sources with

$$\sigma^2_{EGN,n,p} = \sum_{q=1}^{D}\Bigg[\sum_{k_2,m_2,n_2} 3\kappa_1^{(k_2,q)}\kappa_1^{(m_2,q)}\kappa_1^{(n_2,p)}P_{k_2,q}P_{m_2,q}P_{n_2,p}X^a_{n,p}(k_2,m_2,n_2,q) + \sum_{k_2,n_2}\kappa_2^{(k_2,q)}\kappa_1^{(n_2,q)}5(P^2_{k_2,q}P_{n_2,p}X^b_{n,p}(k_2,k_2,n_2,q)+$$
$$P_{k_2,p}P_{k_2,q}P_{n_2,q}X^c_{n,p}(k_2,n_2,k_2,q)) + \sum_{n_2}\kappa_3^{(n_2,q)}P^2_{n_2,q}P_{n_2,p}X^d_{n,p}(n_2,n_2,n_2,q)\Bigg],$$
$$where\begin{cases} X^a_{n,p}(k_2,m_2,n_2,q) = \frac{\tilde{\gamma}^2_{pq}}{4}\iiint_{-\infty}^{\infty}|\eta(f,f_1,f_2)|^2 g^{k_2,q}(f+f_1+f_2)g^{m_2,q}(f+f_2)g^{n_2,p}(f+f_1)g^{n,p}(f)df_1df_2df \\ X^b_{n,p}(k_2,k_2,n_2,q) = \frac{\tilde{\gamma}^2_{pq}}{4}\iiint_{-\infty}^{\infty}|\eta(f,f_1,f_2)|^2 g^{k_2,q}(f+f_1+f_2)g^{k_2,q}(f+f_2)g^{n_2,p}(f+f_1)g^{n,p}(f)df_1df_2df \\ X^c_{n,p}(k_2,n_2,k_2,q) = \frac{\tilde{\gamma}^2_{pq}}{4}\iiint_{-\infty}^{\infty}|\eta(f,f_1,f_2)|^2 g^{n_2,q}(f+f_1+f_2)g^{k_2,q}(f+f_2)g^{k_2,p}(f+f_1)g^{n,p}(f)df_1df_2df \\ X^d_{n,p}(n_2,n_2,n_2,q) = \frac{\tilde{\gamma}^2_{pq}}{4}\iiint_{-\infty}^{\infty}|\eta(f,f_1,f_2)|^2 g^{n_2,q}(f+f_1+f_2)g^{n_2,q}(f+f_2)g^{n_2,p}(f+f_1)g^{n,p}(f)df_1df_2df, \end{cases}$$
$$with\begin{cases}\eta(f,f_1,f_2) = \frac{1-e^{-\alpha_p L_s}e^{j|\beta_q(f+f_1-f_2)-\beta_p(f)+\beta_p(f_2)-\beta_q(f_1)|L_s}}{\alpha_p - j|\beta_q(f+f_1-f_2)-\beta_p(f)+\beta_p(f_2)-\beta_q(f_1)|}\times\frac{1-e^{j|\beta_q(f+f_1-f_2)-\beta_p(f)+\beta_p(f_2)-\beta_q(f_1)|N_sL_s}}{1-e^{j|\beta_q(f+f_1-f_2)-\beta_p(f)+\beta_p(f_2)-\beta_q(f_1)|L_s}}.\end{cases} \quad (1)$$

$$\sigma^2_{EGN,n,p} \approx \sum_{q=1}^{D}\Bigg[3\Bigg(\kappa_1^{(n,q)^2}\kappa_1^{(n,p)}P^2_{n,q}P_{n,p}E(n,p,n,q) + 2\sum_{n'\neq n}\kappa_1^{(n',q)^2}\kappa_1^{(n,p)}P^2_{n',q}P_{n,p}E(n,p,n',q)\Bigg) + \Bigg(\kappa_2^{(n,q)}\kappa_1^{(n,p)}5P^2_{n,q}P_{n,p}$$
$$E(n,p,n,q) + 2\sum_{n'\neq n}\kappa_2^{(n',q)}\kappa_1^{(n,p)}5P^2_{n',q}P_{n,p}E(n,p,n',q)\Bigg) + \Bigg(\kappa_3^{(n,q)}P^2_{n,q}P_{n,p}E(n,p,n,q) + 2\sum_{n'\neq n}\kappa_3^{(n',q)}P^2_{n',q}P_{n',p}E(n,p,n',q)\Bigg)\Bigg],$$
$$where\begin{cases} E(n,p,n',q) = N_sB_{ch,n}\frac{\tilde{\gamma}^2_{pq}}{4}\frac{L^2_{eff}}{4\pi\beta_{2_q}L_{eff,a}}\times \\ \Bigg[ln\Bigg(\pi^2\beta_{2_q}L_{eff,a}B_{ch,n}\Bigg(\frac{\beta_{1q}-\beta_{1p}}{2\pi\beta_{2_q}}+f_n-f_{n'}+\frac{B_{ch,n'}}{2}\Bigg)+\sqrt{\Bigg(\pi^2\beta_{2_q}L_{eff,a}B_{ch,n}\Bigg(\frac{\beta_{1q}-\beta_{1p}}{2\pi\beta_{2_q}}+f_n-f_{n'}+\frac{B_{ch,n'}}{2}\Bigg)\Bigg)^2+1}\Bigg)+ \\ ln\Bigg(\pi^2\beta_{2_q}L_{eff,a}B_{ch,n}\Bigg(f_{n'}-\frac{\beta_{1q}-\beta_{1p}}{2\pi\beta_{2_q}}-f_n+\frac{B_{ch,n'}}{2}\Bigg)+\sqrt{\Bigg(\pi^2\beta_{2_q}L_{eff,a}B_{ch,n}\Bigg(f_{n'}-\frac{\beta_{1q}-\beta_{1p}}{2\pi\beta_{2_q}}-f_n+\frac{B_{ch,n'}}{2}\Bigg)\Bigg)^2+1}\Bigg)\Bigg]. \end{cases} \quad (2)$$

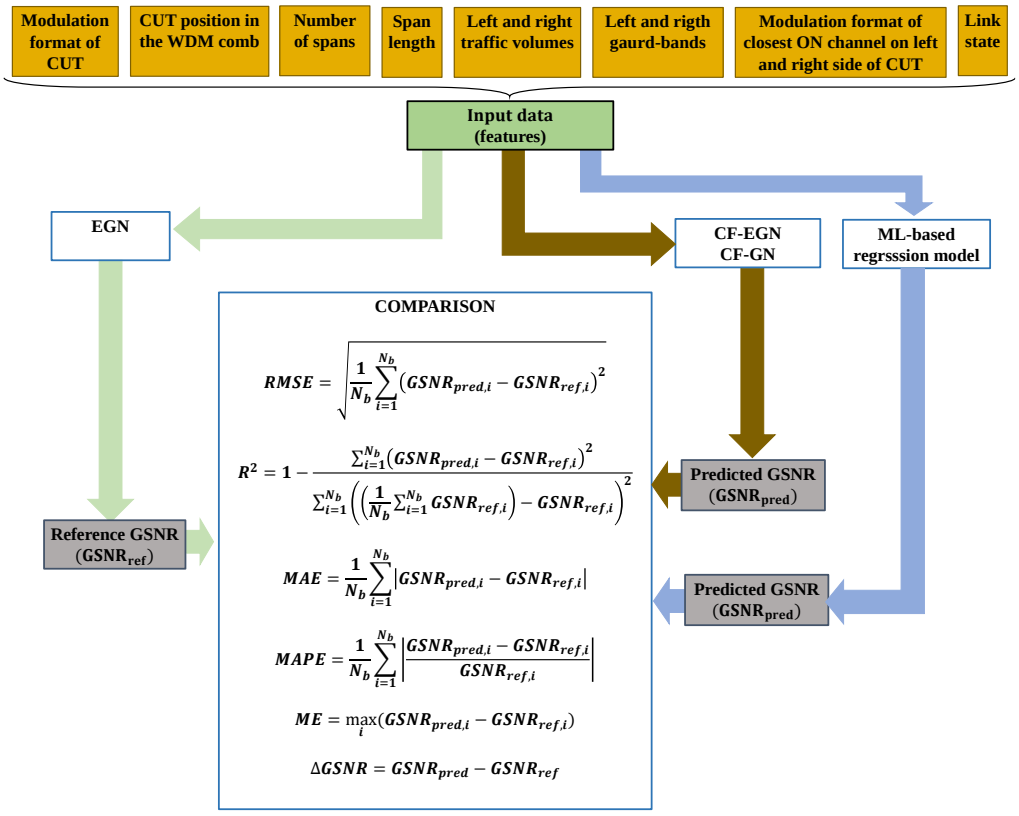


Fig. 2. Diagram describing the process of comparison between approximated (CF-EGN and ML-based) models with reference EGN model.

zero mean and variances $\sigma^2_{ASE,p}$ and $\sigma^2_{EGN,n,p}$, respectively. Here, $\sigma^2_{ASE,p} = N_s F_p (G_p - 1) h\nu \Delta f_n$ where N_s is number of spans, F_p is the amplifier noise figure of pth mode, G_p is amplifier gain of pth mode equal to the span fiber loss of pth mode, h is the Planck's constant, and ν is central frequency [4], [5]. The NLI variance of nth channel and pth mode, $\sigma^2_{EGN,n,p}$, can be evaluated through Eq. (1) [4], [5] where $\kappa_1^{(n,p)} = \mu_2^{(n,p)}$, $\kappa_2^{(n,p)} = \mu_4^{(n,p)} - 2\mu_2^{(n,p)^2}$, and $\kappa_3^{(n,p)} = \mu_6^{(n,p)} - 4\mu_4^{(n,p)}\mu_2^{(n,p)} + 12\mu_2^{(n,p)^3}$, with $\mu_2^{(n,p)}$, $\mu_4^{(n,p)}$, and $\mu_6^{(n,p)}$ denoting the second, fourth, and sixth order moments of the constellation of nth channel and pth mode, respectively. Here, $P_{n,p}$ is the launched power and $g^{(n,p)}(.)$ is the spectral shape of transmitted signal in nth channel and pth mode which is here assumed to be rectangular Nyquist shaping, $\tilde{\gamma}_{pq}$ is nonlinear (coupling) coefficient between pth and qth mode, α_p is attenuation of pth mode, and β_{1p} and β_{2p} are respectively the modal and chromatic dispersion coefficients of pth mode [4], [5]. L_s is the span length. To have a second reference we consider also the CF-EGN model through the formulation for the NLI noise variance of nth channel and pth mode defined by Eq. (2) [6], [7] where $L_{eff} = (1 - e^{-\alpha_p L_s})/\alpha_p$, and $L_{eff,a} = 1/\alpha_p$, and $B_{ch,n}$ and f_n are respectively bandwidth and center frequency of nth channel.

III. ML-BASED QOT ESTIMATION MODELS

Fig. 2 describes the process we followed to statistically compare QoT predictions of approximated (CF-EGN and ML-based) models with the reference EGN model. QoT estimation is performed in terms of GSNR, defined as:

$$GSNR_{n,p} = \frac{P_{n,p}}{\sigma^2_{ASE,p} + \sigma^2_{EGN,n,p}}. \quad (3)$$

To determine the performance of CF-EGN and ML-based regression model we compare predicted GSNR values ($GSNR_{pred}$) with the accurate EGN model, defined as reference ($GSNR_{ref}$). Besides quantifying the level of accuracy of CF-EGN, the main goal of this paper is to propose and analyze ML-based models. We train and optimize ML-based regression models such that $GSNR_{pred}$ becomes close to $GSNR_{ref}$: as performance criteria we consider the root mean square error (RMSE).

The DNN-based regressor model is composed of an input layer with N_f input neurons corresponding to the number of features, N_{hid} hidden layers each with N_{neu} hidden neurons, and a single output neuron. The aim of the training phase is to adjust the weight matrix, $\boldsymbol{W}$, and bias vectors $\boldsymbol{b}$, such that the output converge to the reference GSNR. Therefore, we define the following loss function:

$$L(\boldsymbol{\theta}) = \frac{1}{N_b}\sum_{i=1}^{N_b}(GSNR_{pred,i}(\boldsymbol{\theta}) - GSNR_{ref,i})^2, \quad (4)$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$ is the set of trainable parameters, and N_b is number of batch samples. $\boldsymbol{\theta}$ can be obtained by minimizing the loss function using the stochastic gradient descent method.

TABLE I
GENERATED DATASET DESCRIPTION

Dataset	Description
D1	SMF 3 level sub-band
D2	SMF 7 level sub-band
D3	FMF 3 level sub-band
D4	FMF 7 level sub-band

TABLE II
DEPLOYED TRAIN:TEST DATASET COMBINATIONS

Combination	Train:Test
C1	D1:D2
C2	D2:D2
C3	D3:D4
C4	D4:D4

The well-known ML-based regression models include support vector machine (SVM) [21], K-nearest neighborhood (KNN) [22], decision tree (DT) [23], random forest (RF) [24], extended gradient boosting (XGB) [25], linear regression (LR) [26], ridge regression (RR) [27], and Bayesian ridge regression (BR) [28].

IV. DATASET GENERATION

To obtain a good QoT estimation and avoid biases, the dataset should be large enough and properly generated covering the entire space of the features. We synthetically generate the dataset based on the EGN models [4]-[5]. Common and fixed parameters are the wavelength multiplexing in a fixed-grid allocated in the C-band, ($5\ THz$) centered around $1550\ nm$, with a channel spacing of $75\ GHz$ and transceivers set to work with a symbol-rate of $64\ Gbaud$. Therefore, both SMF and FMF links consist of a maximum of 66 WDM channels multiplexed either over 1 or 3 modes, respectively.

The overall set of possible links analyzed is considered by randomizing:

- the link state, intended as the selection of channels in ON state;
- the modulation format on each channel;
- the number of spans;
- the equal span length.

The modulation format of each channel is randomly chosen with same probability between PM binary phase-shift keying (BPSK), PM quaternary phase-shift keying (QPSK), and PM M-QAM, with M $\in \{8, 16, 32, 64\}$. The number of span composing the link is randomly selected in the range from 1 to 8 spans. All spans in a link have the same length that is randomly selected with a uniform distribution between 80 to $120\ km$.

In all cases, fiber parameters (non-linear coefficients, coupling coefficients in case of FMF, chromatic dispersion, modal dispersion for FMF, and attenuation) are taken from [29]. After each span, an ideal amplifier with $5\ dB$ noise figure compensates for fiber attenuation.

Between randomized link parameters, the most critical is the link state, because it has the larger dimension. We consider the condition where we have an average of 50% *randomly ON* channels. The uniform launch power per channel and mode is optimized case by case depending on the link state. Considering 66 channels with TWO possible states, ON and OFF, we have a total of $2^{66} \cong 7.37 \cdot 10^{19}$ cases for SMF (and $2^{66 \cdot 3} \cong 4.01 \cdot 10^{59}$ cases for FMF). Consider that link state is only one of the randomized input parameters: it must be combined with all other to generate the input space.

To reduce at least the dimension of the link state, we approach the dataset generation on a sub-band basis instead of on a channel basis [30]. We group channels into 11 sub-bands each with 6 channels. Each sub-band has 7 possible levels of filling, depending on the number of channels in the ON state: from 0 to a maximum of all 6. Now the number of total cases is reduced to $7^{11} \cong 1.98 \cdot 10^9$ for SMF (and $7^{33} \cong 7.73 \cdot 10^{27}$ for FMF): a huge reduction but still a very large space to be explored. To further reduce this number we also considered a simplified approach where the whole sub-band can assume only three states: empty, 50% ON and fully ON. Now the total number of case is only $3^{11} \cong 1.77 \cdot 10^5$ for SMF (and $3^{33} \cong 5.55 \cdot 10^{15}$ for FMF). Accordingly, we generate two datasets for each fiber type, SMF and FMF, both based on the sub-band approach: a first one with 7-levels and a second one with 3-levels. Each dataset is composed of 60000 train and 6000 test samples.

Generated datasets are described in Table I, the 3 and 7-level sub-band datasets are respectively named D1 and D2 for SMF link, and D3 and D4 for FMF link. Thereby, we consider the train:test combinations described by Table II.

In order to move to the dataset optimization, we must first define and determine features, intended as a selected list of input parameters or derived quantities describing the link and having an impact on QoT. We selected the following list of features:

- the modulation format of channel and mode under test (CUT) (1st feature)
- CUT position in the WDM comb, i.e. channel and mode under test indices (2nd and 3rd features)
- span length (4th feature)
- number of spans (5th feature)
- the left and right traffic volumes (6th and 7th features)
- the left and right guard bands, i.e. the number of empty channels on left and right side of CUT (8th and 9th features)
- the modulation format of closest ON channel on left and right side of CUT (10th and 11th features for SMF, and 10th to 15th features for FMF)
- the link state, i.e. the number of occupied channels per sub-band as 12th to 22th features for SMF, and 16th to 48th features for FMF. An important aspect of the generated dataset is the consideration of partial load of link state for D modes and N_{ch} channels with N_{sub} sub-bands per mode.

SMF and FMF have 22 and 48 features, respectively. Then the GSNR is calculated as the single label and associated.

V. DATASET OPTIMIZATION

Dataset optimization is the procedure of adjusting the size of the dataset, preprocessing the features and the labels,

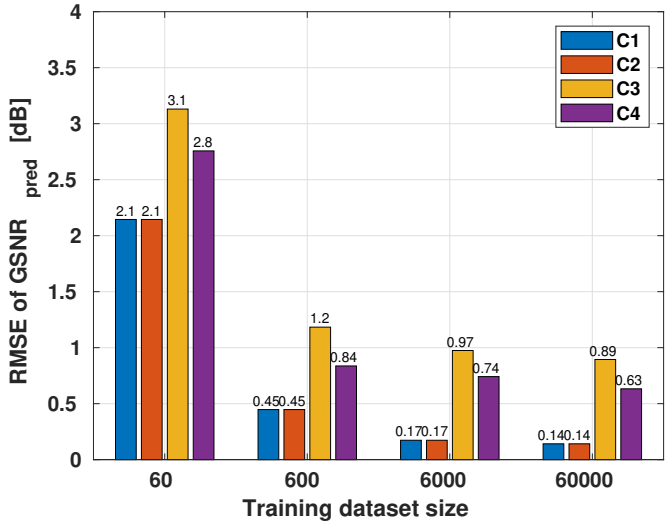


Fig. 3. RMSE values for training dataset sizes $60, 600, 6000$, and 60000, leveraging combinations C1, C2, C3, and C4.

selecting the relevant feature set, and reducing the feature dimension. Therefore, for dataset optimization we need to compare the result of applying different dataset adjustments on an employed ML model. Here, due to space limitation, we only report the results for the DNN-based regressor structure provided by "MLPRegression" package from Python/Scikit-learn library [31].

A. Dataset size adjustment

To evaluate the impact of the dataset size on the performance, in Fig. 3 we show the RMSE values for training dataset sizes $60, 600, 6000$, and 60000, leveraging combinations C1, C2, C3, and C4. The test dataset size is set to 6000 over all the paper. As seen, the DNN-based regressor provides the same performance in C1 and C2 while the performance in C4 is better than C3. Actually, in SMF, the regressor only learns the information about inter/intra channel nonlinear interactions while in FMF case, it should also learns the inter/intra modal nonlinear interactions as well as coupling. Therefore, the FMF case is a more complex scenario and the DNN requires more dataset point to properly train. The obtained results show that increasing the number of data points reduces the RMSE in all combinations. With a dataset of 60000 samples, the RMSE for C1 and C2 is 0.14 dB while the RMSE for C3 and C4 is 0.89 and 0.63, respectively. The main issue towards training a DNN-based regressor for QoT estimation is dataset generation. We rely on synthetic dataset generation by an accurate model such as EGN. We generate the 60000 point D1, D2, D3, and D4 datasets in 3 months by utilizing 200 parallel CPUs. For SMF, steep descent shown in Fig. 3 is such that further dataset size increase does not improve RMSE so much. However, in FMF, the obtained values for RMSE per dataset size indicate the RMSE can be improved more by further increasing the dataset points. We use this dataset size in the remainder of this paper.

B. Feature preprocessing

Regressors tend to weight larger the bigger features, in fact, features with variances larger orders of magnitude than the others prohibit regressor to learn properly from other features. Feature preprocessing, a common requirement for regression, is the method of changing the raw dataset into a more proper representation through scaling, transformation, normalization, and discretization. Scaling is the method of individual standardization of the features which presents them in a fixed range in order to handle highly varying features. The common scalers are Standard, Min-Max, and Max-Abs scalers. Standard scaler is a quick and easy way for scaling the features into a zero mean and unit variance version.

The Min-Max scaler scales the features between a given minimum-maximum value, often between zero and one is preferred. The Max-Abs scaler scales the maximum absolute value of each feature to the unit value. In scaling, we change the range of features while in transformation, we change the shape of features distribution. The general transformation methods include quantile and power transformers. These non-linear transformers are based on monotonic transformations of the features. Quantile transformation is a non-parametric transformer and maps the feature distribution to uniform between $[0, 1]$. This method deploys a rank transformation and smooths out unusual distributions. Quantile transformation is less affected by outliers compared with scaling. Power transformation is a parametric transformer and maps the feature distribution close to Gaussian to stabilize the variance and minimize the skewness. Normalization scales each feature to have a unit norm. Discretization separates continuous features into discrete values and transforms the features with continuous attributes into features only with nominal attributes. It is similar to constructing discrete histograms for the continuous features. Histograms counts features that fall into bins, however, discretization assigns feature values to bins.

Fig. 4 shows the RMSE values for different preprocessing methods including scaling, transformation, normalization, and discretization, considering combinations C1, C2, C3, and C4. Min-Max scaler has better performance than the others, and normalization obtains higher RMSE values than the others. In SMF, Standard, Min-Max, Max-Abs scaling, and quantile transformation provide the same RMSE values. However, in FMF, the choice of feature preprocessing affects the performance. In conclusion, we choose the Min-Max scalar for feature preprocessing.

C. Feature selection

Feature selection is the procedure of isolating the most relevant, non-redundant, and consistent features to be use in regression. We have so far considered 22 and 48 features in the regressor models. An important question is which features are more important to achieve good regression performance, as removing worthless features leads to a regressor with less cost and complexity while removing the worthwhile features degrades the performance. Hence, we now evaluate the usefulness of each feature by comparing the regression performance after training the regressor, considering seven

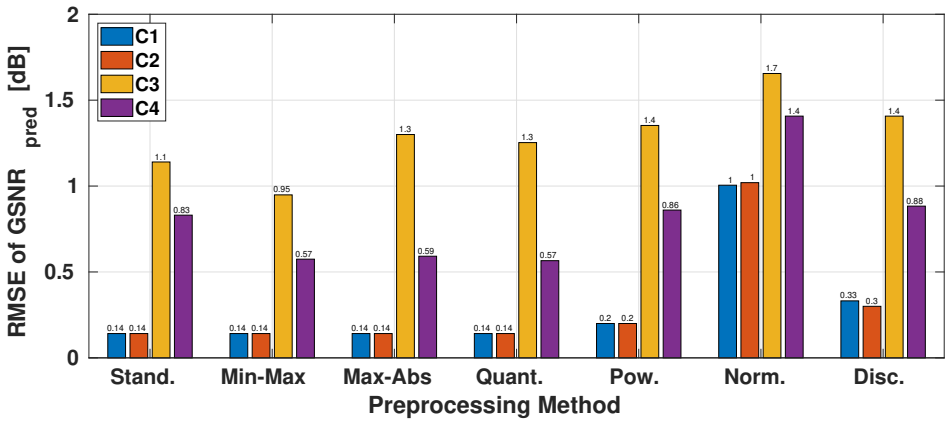


Fig. 4. RMSE values for different preprocessing methods including scaling, transformation, normalization, and discretization, considering combination C1, C2, C3, and C4.

TABLE III
CONSIDERED FEATURE SETS

	F1	F2	F3	F4	F5	F6	F7
Modulation format of CUT	*	*	*	*	-	*	*
CUT position in the WDM comb	*	*	*	*	*	*	*
Span length	*	*	*	*	*	-	*
Number of spans	*	*	*	*	*	-	*
Left and right traffic volumes	*	*	-	*	*	*	-
Left and right guard bands	*	*	*	-	*	*	-
Modulation format of closest ON channel on left and right side of CUT	*	*	*	*	-	*	-
Link state	*	-	*	*	*	*	-

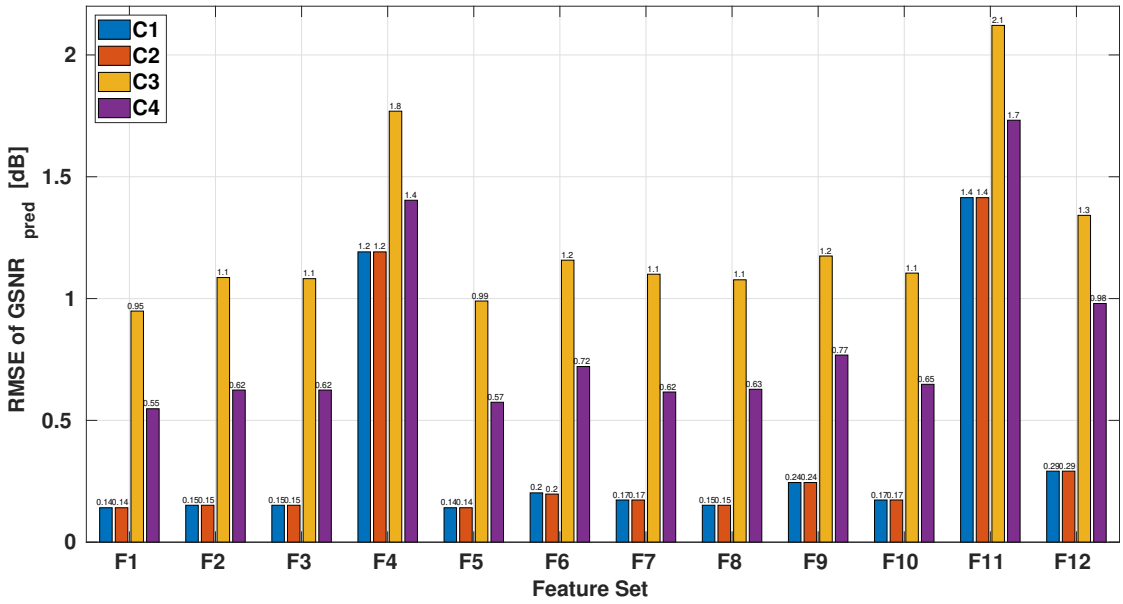


Fig. 5. RMSE values for feature sets $F1$, $F2$, $F3$, $F4$, $F5$, $F6$, and $F7$, considering combination C1, C2, C3, and C4.

different feature sets (F1 to F7) of features listed in Section IV. The considered subsets are listed in Table III. The obtained RMSE values for these feature sets are reported in Fig. 5, considering combinations C1, C2, C3, and C4. Results also show that, in both topologies, training the regressor with the feature sets F1, F4, and F5 leads to the highest and comparable RMSE values. Note that F1 includes all features, whereas F4 and F5 exclude the features characterizing left and right guard bands and modulation format, respectively. However, results obtained in scenario F1 are slightly higher, which leads us to conclude that information on these does provide some insight into regression, and we properly designed features. Nothing can be removed because it is already a sort of best feature set including all important parameters affecting optical signal propagation. However, if we further remove either the Link state or the left and right traffic volumes from set F1 (as in subsets F2 and F3, respectively), regression performance degrades. In particular, results related to F7 show that attributes characterizing the neighbor ON channels (left and right traffic volumes, left and right guard bands, modulation format, and Link state) are very useful. Performance degradation becomes extremely severe when eliminating the span length and number of spans from the feature set, as done in F6.

D. Feature reduction

Feature dimension reduction decreases the processing time and required memory. It removes multi-collinearity and improves the interpretation of the parameters of the regression model. One of the most popular techniques for dimension reduction is principal component analysis (PCA) which identifies the patterns in features based on their correlation. In a nutshell, PCA finds the directions of maximum variance in high-dimensional features and depending on the PCA components projects them into a new subspace with equal or lower dimensions than the original one. PCA assumes some dimensions of the dataset are affected by sources of noise, artifact, or interference and removes them while we generated the dataset synthetically based on EGN model without considering any of these sources, therefore, the generated dataset has proper dimensions and no reduction is required.

E. Label preprocessing

Besides feature preprocessing, label preprocessing is also crucial in regression. The common label preprocessing approaches include scaling by $log(.)$ and $log_2(.)$. Note that the calculated label (the GSNR) is in dB which means that the scaling is already done while dataset generation and further scaling would not change anything. Therefore, we do not need to deploy any more preprocessing on the Labels.

VI. Model selection

Model selection is the task of tuning hyperparameters for each of available ML models and selecting one of them. Regarding this, we first tune hyperparameters of a selected set of regressors described by section III, and compare the complexity and performance of different ML-based regressors.

A. Hyperparameter tuning

Reporting the hyperparameter tuning results for selected set of regressors requires a large space. Thereby, here we demonstrate hyperparameter tuning for DNN-based regressor, and just report the tuned hyperparameter values for other regressors. Hyperparameter tuning for DNN-based regressor studies its convergence through adjusting iteration number, learning rate, number of hidden layers, number of hidden neurons, and optimizer. Figs. 6(a), 6(b), 6(c), 6(d), and 6(e), present RMSE values respectively for different iteration numbers, learning rates, number of hidden layers, number of hidden neurons, and optimizers, for combinations C1, C2, C3, and C4. The DNN-based regressor is iteratively trained considering the stochastic gradient descent (SGD) optimizer with learning rate 0.001, batch size of 100, and the ReLU and linear activation functions in the hidden and last layers as baselines. We early terminate the training if the RMSE value does not improve for 30 iterations to avoid overfitting. It should be noted that the DNN weight and bias parameters are randomly initialized before training, therefore, even with the same dataset and DNN, the optimization trajectories might be different. However, the train and test RMSE values are consistent and similar. Fig. 6(a) shows convergence for C1 and C2 happens after 100 iterations while in C3 and C4 it occurs after 50 iterations. Fig. 6(b) shows the learning rate 0.001 is a proper choice for C1, C2, C3, and C4. Figs. 6(c) and 6(d) depict that 2 hidden layers with 1000 hidden neurons provide a convergence for C1, C2, C3, and C4. Fig. 6(e) plots RMSE values for Adam, SGD, RMSprop, Adadelta, Adagrad, Adamax, Nadam, and Ftrl optimizers. As seen, in SMF, Adam, SGD, RMSprop, Adamard, and Nadam, and in FMF, Adam, SGD, Adamard perform close. However, Adam achieves lower RMSE values than the others and is a good choice for both SMF and FMF.

For SVM, we use radial basis function (RBF) kernel with the γ equal to the inverse of multiplication of number of features and variance of features without limitation on maximum iteration number. The KNN is employed by five neighbors and all points in each neighborhood are weighted equally. Ball tree algorithm with 30 leaves was used to compute the nearest neighbors based on euclidean distance metric. In DT (and also in DTs inside RF and XGB), we deploy the mean squared error as the function to measure the quality of a split, and the variance reduction as feature selection criterion which minimizes the L_2 loss based on the mean of each terminal node. At each node, we choose the best split by considering all features, and expand the trees until all leaves contain less than 2 samples. The minimum number of samples at each leaf node is 1. In RF and XGB use squared error loss with 0.1 learning rate and 100 DTs. In RR, we consider $\alpha = 1$ as the constant multiplied by L_2 term, controlling regularization strength. For optimizing the weights, we use stochastic average gradient descent with 1000 iterations. In BR, we use $\alpha_1 = 10^{-6}$ as shape parameter of Gamma distribution prior over α, $\alpha_2 = 10^{-6}$ as rate parameter of Gamma distribution prior over α, $\lambda_1 = 10^{-6}$ as shape parameter of Gamma distribution prior over the λ, and $\lambda_2 = 10^{-6}$ as rate parameter of Gamma

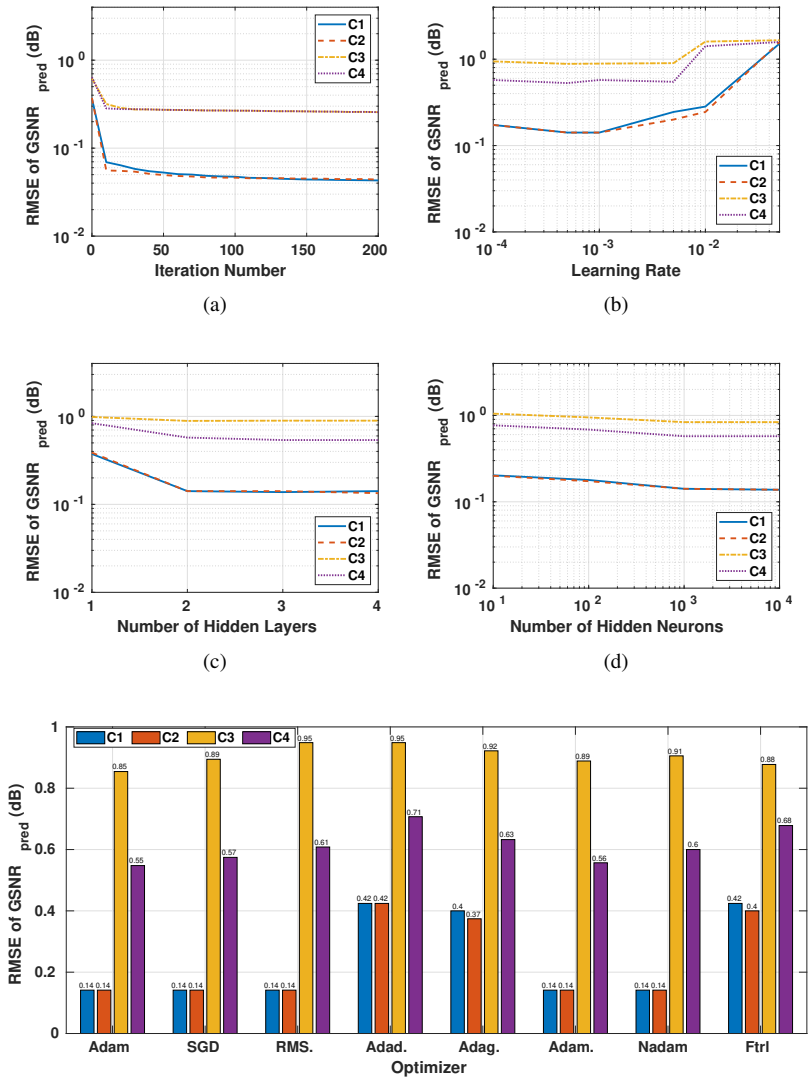


Fig. 6. RMSE values for different a) iteration numbers, b) learning rates, c) number of hidden layers, d) number of hidden neurons, and e) optimizers, for combinations C1, C2, C3, and C4.

distribution prior over λ. The maximum number of iterations is 300.

B. Comparison between different ML-based regressors

Fig. 7 shows the RMSE versus normalized runtime for regression methods DNN, SVM, KNN, DT, RF, XGB, LR, RR, BR, and CF-EGN for combinations a) C1, b) C2, c) C3, and d) C4. DNN-based regressor always performs better than other ML-based regressors and CF-EGN in terms of RMSE. DNN-based regressor is 100 times faster than CF-EGN. DNN-based regressor has a more complex structure compared with other ML regressors and that is why it is slower than the others (except SVM and KNN). However, this difference is not so much and DNN-based regressor is fast enough for real time QoT estimation applications. As seen, each regressor perform the same in C1 and C2 while performance in C4 is better than C3. Since the dataset D3 does not contain all required information about feature space of D4 and needs to be increased.

Fig. 8 introduces the CDF of $\Delta GSNR = |GSNR_{pred} - GSNR_{ref}|$, for DNN, XGB, and CF-EGN, considering combinations a) C1, b) C2, c) C3, and d) C4. Here, to avoid congestion, we only compare our proposed DNN-based regressor with XGB which has the best performance among ML-based regressors, and CF-EGN which is a well-known conventional method and could be an alternative approach. At 99% of cases, the GSNR estimation error by DNN-based regressor is lower than 0.3 dB, 0.3 dB, 1.2 dB, and 1 dB for C1, C2, C3, and C4, respectively. In C1 and C2 (the simpler scenarios) DNN takes the advantages of its complex structure and provides slightly better results at 99% of cases than XGB while in C3 and C4 (the complex cases) DNN and XGB perform the same. Although XGB has less normalized runtime than DNN, DNN is fast enough for real time applications and we could trade a small loss of speed for higher accuracy in C1 and

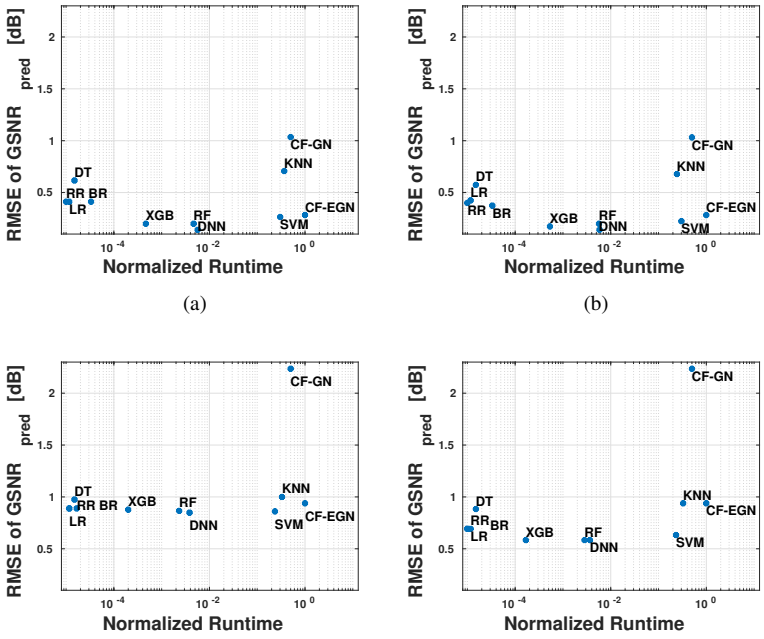


Fig. 7. RMSE versus normalized runtime for regression methods DNN, SVM, KNN, DT, RF, XGB, LR, RR, BR, and CF-EGN for combinations a) C1, b) C2, c) C3, and d) C4.

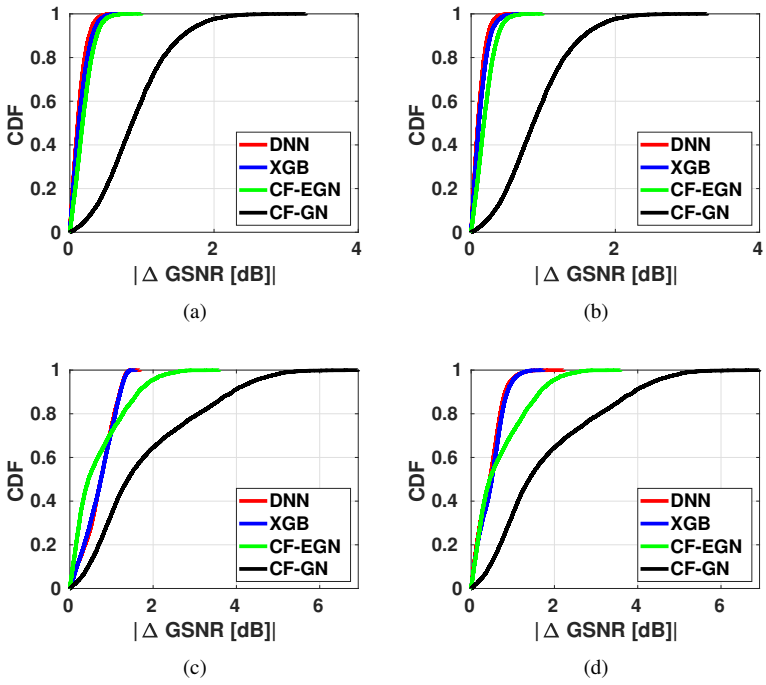


Fig. 8. CDF of $\Delta GSNR$, for proposed DNN, XGB, and CF-EGN, considering combinations a) C1, b) C2, c) C3, and d) C4.

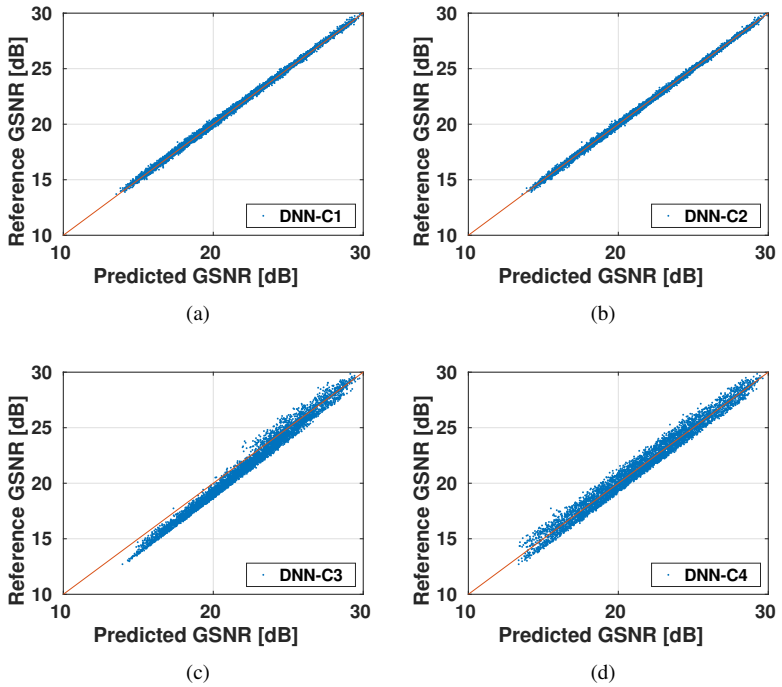


Fig. 9. Scatterplots of reference GSNR and predicted GSNR by DNN-based regressor, for combinations a) C1, b) C2, c) C3, and d) C4.

C2. At 99% of cases, the GSNR estimation error by CF-EGN is lower than 0.8 dB, 0.8 dB, 2.5 dB, and 2.5 dB for C1, C2, C3, and C4, respectively. In C1 and C2, the CF-EGN provides closer GSNR estimations to DNN at 99% of cases than C3 and C4. The CF-EGN is accurate for cases with channel bandwidth close to the symbol rate [7] which is hard to achieve while dealing with a fully randomized link state especially in FMF case wherein this assumption should be consistent for all modes.

Fig. 9 describes scatterplots of reference GSNR and predicted GSNR by DNN-based regressor, for combinations a) C1, b) C2, c) C3, and d) C4. As seen, the scatterplots are propagated along $y = x$ line which means that the DNN-based regressor achieves a good performance. In C1 and C2 we experience denser plots compared with C3 and C4, as FMF scenario is a more complex problem. In C3, the GSNR estimation has more bias than C4, as we train the DNN based on D3 and test on D4, this in turn results in higher RMSE values for C3 compared with C4.

VII. Conclusion

In this paper, we have proposed a DNN structure for QoT estimation of optical communication links. We have presented a comprehensive investigation considering different ML-based regression methods for estimating GSNR in partial-load SMF and FMF links. Synthetic datasets have been generated based on EGN model. Results have shown that the DNN-based regressor can provide higher accuracy in terms of RMSE compared with other state-of-the-art methods such as XGB regressor and analytical approaches as the CF-EGN. In 99% of cases, the GSNR estimation error obtained by DNN-based regressor is lower than 0.3 dB for SMF and and 1 dB for FMF. Moreover, the DNN-based regressor requires much less computation complexity compared with CF-EGN and is candidate solution for real time QoT estimation applications needed in control plane of dynamically reconfigurable optical networks.

Acknowledgement

This work was supported by the Italian Ministry for University and Research (PRIN 2017, project FIRST). Computational resources were provided by HPC@POLITO (http://www.hpc.polito.it). Farhad Arpanaei acknowledges support from the CONEX-Plus programme funded by Universidad Carlos III de Madrid and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 801538.

References

[1] R. J. Essiambre, "Nonlinear capacity limit to optical communications", *In Nonlinear Optics*, pp. NTu2A-3, July, 2015.

[2] G. Rademacher, K. Petermann, "Nonlinear Gaussian noise model for multimode fibers with space-division multiplexing", *Journal of Lightwave Technology*, Vol. 34, No. 9, pp. 2280-2287, 2016.

[3] G. Rademacher, R. S. Luis, B. Puttnam, R. Ryf, S.Van der Heide, T. A. Eriksson, N.K. Fontaine, H. Chen, R.J. Essiambre, Y. Awaji, H. Furukawa, "A Comparative Study of Few-Mode Fiber and Coupled-Core Multi-Core Fiber Transmission", *Journal of Lightwave Technology*, 2022

[4] A. Carena, G. Bosco, V. Curri, Y. Jiang, P. Poggiolini, F. Forghieri, "The EGN model of non-linear fiber propagation", *Optics Express*, Vol. 22, No. 13, pp. 16335–16362, 2014.

[5] M. A. Amirabadi, M. H. Kahaei, S. A. Nezamalhosseini, L. R. Chen "Joint Power and Gain Allocation in MDM-WDM Optical Communication Networks Based on Enhanced Gaussian Noise Model", *IEEE Access*, Vol. 10, pp. 23122-23139, 2022.

[6] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, F. Forghieri, "A simple and effective closed-form GN model correction formula accounting for signal non-Gaussian distribution", *Journal of Lightwave Technology*, Vol. 33, No. 2, pp. 459-473, 2015

[7] M. A. Amirabadi, M. H. Kahaei, S. A. Nezamalhosseini, F. Arepanaei, A. Carena "Closed-Form EGN Model for FMF Systems", *In ACP*, 2021.

[8] 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", *Journal of Optical Communication and Networking*, Vol. 9, pp. 98–108, 2017.

[9] R. M. Morais, B. Pereira, and J. Pedro, "Fast and high-precision optical performance evaluation for cognitive optical networks", *In Optical Fiber Communication Conference and Exhibition (OFC)*, 2020.

[10] X. Liu, H. Lun, M. Fu, Y. Fan, L. Yi, W. Hu, and Q. Zhuge, "A three stage training framework for customizing link models for optical networks", *In Optical Fiber Communication Conference and Exhibition (OFC)*, 2020.

[11] A. Mahajan, K. Christodoulopoulos, R. Martínez, S. Spadaro, and R. Muñoz, "Modeling EDFA gain ripple and filter penalties with ML for accurate QoT estimation", *Journal of Lightwave Technology*, Vol. 38, pp. 2616–2629 2020.

[12] I. Sartzetakis, K. K. Christodoulopoulos, and E. M. Varvarigos, "Accurate quality of transmission estimation with ML", *J. Opt. Commun. Netw.*, Vol. 11, pp. 140–150, 2019.

[13] C. Rottondi, L. Barletta, A. Giusti, M. Tornatore, "Machine-learning method for quality of transmission prediction of unestablished lightpaths", *Journal of Optical Communications and Networking*, Vol. 10, No. 2, pp. A286-A297, 2018.

[14] S. Aladin, A. V. S. Tran, S. Allogba, C. Tremblay, "Quality of Transmission Estimation and Short-Term Performance Forecast of Lightpaths", *Journal of Lightwave Technology*, Vol. 38, No. 10, 2020.

[15] A. A. Diaz-Montiel, S. Aladin, C. Tremblay, M. Ruffini, "Active wavelength load as a feature for QoT estimation based on support vector machine", *In ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1-6, May 2019.

[16] R. M. Morais, J. Pedro, "Machine Learning Models for Estimating Quality of Transmission in DWDM Networks", *Journal of Optical Communications and Networking*, Vol. 10, No. 10, 2018.

[17] J. Müller, S. K. Patri, T. Fehenberger, C. Mas-Machuca, H. Griesser, J. P. Elbers, "A QoT Estimation Method using EGN-assisted Machine Learning for Network Planning Applications", *In 2021 European Conference on Optical Communication (ECOC)*, pp. 1-4, Sep. 2021.

[18] J. Pesic, M. Lonardi, N. Rossi, T. Zami, E. Seve, and Y. Pointurier, "How uncertainty on the fiber span lengths influences QoT estimation using ML in WDM networks", *In Optical Fiber Communication Conference and Exhibition (OFC)*, 2020.

[19] J. Müller, T. Fehenberger, S. K. Patri, K. Kaeval, H. Griesser, M. Tikas, J. P. Elbers, "Estimating Quality of Transmission in a Live Production Network using Machine Learning", *In Optical Fiber Communication Conference and Exhibition (OFC)*, pp. 1-3, 2021.

[20] I. Khan, M. Bilal, V. Curri, "Assessment of cross-train machine learning techniques for QoT-estimation in agnostic optical networks", *OSA Continuum*, Vol. 3, No. 10, pp. 2690-2706, 2020.

[21] M. Martin, "On-line support vector machine regression", *In European Conference on Machine Learning*, pp. 282-294, Aug. 2002.

[22] Y. Song, J. Liang, J. Lu, X. Zhao, "An efficient instance selection algorithm for k nearest neighbor regression", *Neurocomputing*, Vol. 251, pp. 26-34, 2017.

[23] A. Suárez, J. F. Lutsko, "Globally optimal fuzzy decision trees for classification and regression", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1297-1311, 1999.

[24] M. R. Segal, "ML benchmarks and random forest regression", 2004.

[25] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, "Xgboost: extreme gradient boosting", *R package version 0.4-2*, Vol. 1, No. 4, pp. 1-4, 2015.

[26] G. A. Seber, A. J. Lee, "Linear regression analysis", *John Wiley & Sons*, Vol. 329, 2012.

[27] A. E. Hoerl, R. W. Kannard, K. F. Baldwin, "Ridge regression: some simulations", *Communications in Statistics-Theory and Methods*, Vol. 4, No. 2, pp. 105-123, 1975.

[28] J. Ranstam, J. A. Cook, "LASSO regression", *Journal of British Surgery*, Vol. 105, No.10, pp. 1348-1348, 2018.

[29] S. Mumtaz, R. J. Essiambre, G. P. Agrawal, "Nonlinear Propagation in Multimode and Multicore Fibers: Generalization of the Manakov Equations", *Journal of Lightwave Technology*, Vol. 31, No. 3, pp. 398-406, 2012.

[30] A. M. Rosa Brusin, U. C. de Moura, V. Curri, D. Zibar, A. Carena, "Introducing load aware neural networks for accurate predictions of Raman amplifiers", *Journal of Lightwave Technology*, Vol. 38, No. 23, pp. 6481-6491, 2020.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, "Scikit-learn: ML in Python", *the Journal of ML research*, Vol. 12, pp. 2825-2830, 2011.