ML-based Modeling of EDFA Pluggable Modules for OSNR Estimation

1st Mariano Devigili
University of Trento, Trento, Italy
mariano.devigili@unitn.it

2nd Emilio Riccardi
Fibercop S.p.A.
Turin, Italy

3rd Antonino Nespola
LINKS Foundation
Turin, Italy

4th Roberto Morro
Fibercop S.p.A.
Turin, Italy

5th Stefano Straullu
LINKS Foundation
Turin, Italy

6th Rocco D'Ingillo
Politecnico di Torino
Turin, Italy

7th Vittorio Curri
Politecnico di Torino
Turin, Italy

8th Luis Velasco
Universitat Politècnica de Catalunya
Barcelona, Spain

9th Fabrizio Granelli
University of Trento
Trento, Italy

***Abstract*—Erbium doped fiber amplifiers (EDFA) return an output dependent on the target gain, the channel-loading configuration, as well as the input channels' wavelengths and powers. Due to absence of accurate analytical models, in this article we investigate machine learning methods and their use for estimating the output of an EDFA pluggable module. First, we characterize experimentally a preamplifier. Secondly, after time demanding hyperparameter optimizations we evaluate the performances of deep learning and ensemble methods. Then, we evaluate the generalizability of the models and the benefits of transfer learning strategies to reduce the generalization error. Finally, we propose an algorithm for OSNR estimation that we evaluate based on experimental data.**

Keywords—EDFA, Machine Learning, OSNR, IPoWDM

I. Introduction

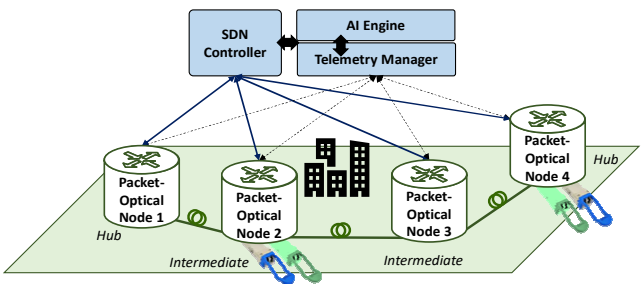
Erbium-doped fiber amplifiers (EDFA) are key components in optical communications systems having a direct impact on the quality of transmission (QoT) of optical connections. Keeping track of the channel power profile and of the amplified spontaneous emission (ASE) noise profile is critical both during network planning and operation to ensure that all the wavelength division multiplexing (WDM) channels have an acceptable optical signal-to-noise ratio (OSNR). Recently, EDFA pluggable modules have become available paving the way towards cost-effective upgrades of IP over WDM (IPoWDM) metro optical networks. These devices can be controlled directly by host packet-optical nodes (e.g., routers or L2 switches equipped with pluggable coherent transceivers) or by external software defined network (SDN) controllers using suitable application programmable interfaces (APIs) exposed by the host device. In addition, such packet-optical nodes can include containerized environments allowing customized APIs and advanced control functionalities to be deployed directly on the host [1,2]. Note that pluggable EDFAs are typically equipped with low-cost power meters measuring the total input and output powers but they do not provide any detail regarding the QoT of the WDM channels.

Unfortunately, accurate explicit analytical physical models describing EDFAs are currently unavailable [3]. For this reason, we propose to integrate an artificial intelligence (AI) engine [4] in the SDN control layer with the aim of building EDFA machine learning (ML)-models providing precise information about the QoT of the WDM channels (e.g., OSNR). In the literature several works on EDFA modeling have been proposed based on physics-driven [5], ML-driven [6-8] and hybrid [9] approaches. A popular analytical model is the so-called center of mass model (CMM) [5], which tends to be accurate when a single channel or all channels are turned on but not for other channel-loading configurations [1,7]. In [7], it was shown that ML-driven models indeed outperform the CMM. Finally, a hybrid model leveraging both the CMM and deep learning was proposed in [9].

In this work, we concentrate on ML-models to estimate the channel power, the ASE noise power and the OSNR of all the WDM channels at the output of an EDFA pluggable module. From a ML perspective, EDFA modeling is a multi-regression problem based on tabular data. In this context, deep learning algorithms are attracting interest for this data type [10]. Nonetheless, ensemble methods have repeatedly outperformed deep models and extreme gradient boosting (XGB) remains one of the preferred algorithms for tabular data [11]. To the best of our knowledge, the ML-driven approaches for EDFA modeling proposed so far in the literature are based on multi-layer perceptron (MLP) [6,7], convolutional neural networks (CNN) [8] and deep transfer learning (TL) [7] while the use of XGB remains unexplored. Based on these remarks, in this article we compare the performances of deep learning models for EDFA modeling to the ones of ensemble methods. Moreover, we evaluate the benefits of TL to reduce the generalization error and finally we propose and evaluate an algorithm which exploits the proposed ML-models for estimating the OSNR.

II. Reference Network Scenario

Fig. 1 illustrates the envisioned optical network scenario which represents a typical IPoWDM transport context for metro-regional scope. The topology is horseshoe with dual hub nodes as service points of presence or to interconnect to higher hierarchy segments of the network. In the horseshoe, both hub and intermediate nodes, are packet-optical nodes



978-3-903176-67-6 © 2025

Fig. 1 Reference network scenario.

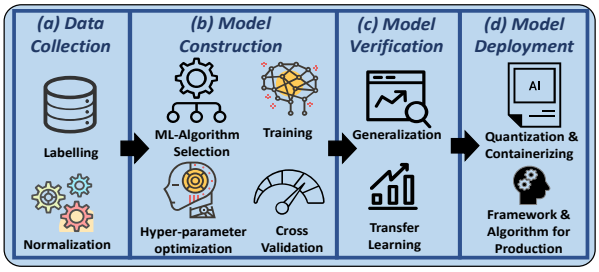


Fig. 2 AI engine phases toward ML-models in production.

Algorithm I OSNR Estimation.

	INPUT: *rq, hl, ml* **OUTPUT:** *Output*
1	$A[] \leftarrow ml$.amp_of_interest(rq.lightpath)
2	**for** a **in** $A[]$:
3	$P_{ch,IN}[], S_{ch}[] \leftarrow ml$.RetrieveInPwProfile($rq$,$hl$)
4	$P_{tot,IN}, P_{tot,OUT}, G_t, C_p \leftarrow ml$.readEDFA()
5	$\hat{y}[] \leftarrow ml$.regr($P_{tot,IN}, P_{tot,OUT}, G_t, C_p, P_{ch,IN}[], S_{ch}[]$)
6	$\hat{N}[], \widehat{S_{ch}}[] \leftarrow ml$.distinguish($\hat{y}[], S_{ch}[]$)
7	$\widehat{N_{ch}}[] \leftarrow ml$.interpolate($\hat{N}[]$)
8	$\widehat{OSNR}[] \leftarrow ml$.computeOSNR($\widehat{S_{ch}}[], \widehat{N_{ch}}[]$)
9	$Output \leftarrow ml$.append($\widehat{OSNR}[]$)
10	**return** *Output*

equipped with an Ethernet switch and hosting pluggable coherent transceivers and pluggable EDFAs. A suitable network operating system (e.g., SONiC) allows to control both packet and optical functionalities. The pluggable modules are the only active elements in this optical network, where the traffic aggregation and disaggregation (i.e., add-drop) and the transport is managed by passive elements.

A control layer is envisioned on top of this joint packet optical domain, where an SDN controller coordinates the packet and optical domains and enables a distributed telemetry system. We assume that the switches include some containerized SDN agent (e.g., instantiating a NETCONF server) that can access pluggable modules to monitor and/or configure their parameters. Moreover, the SDN controller is assisted by both a telemetry manager that retrieves a historical database (DB) and by an AI engine that builds ML-models of the EDFAs to estimate its output in the operating conditions of interest.

Fig. 2 summarizes the main phases carried out by an AI engine to produce a ML-model: *a*) *data collection*, in which datasets (DS) are collected to build and evaluate the ML-models. Note the data need to be properly labelled and normalized before being fed to the ML models; *b*) *model construction*: finding the best performing ML method and its optimal hyperparameters through cross validation and hyperparameter optimizations (HPO); *c*) *model verification,* where the generalizability of the models is evaluated and TL techniques are applied to transfer knowledge learned in a base task to target ones; and finally, *d*) *model deployment*, where the ML-models are deployed by algorithms for specific applications.

This article focuses mainly on the first three phases with the goals of identifying the ML method more suitable to model a preamplifier pluggable module and of verifying the models' generalizability and performance to estimate the OSNR.

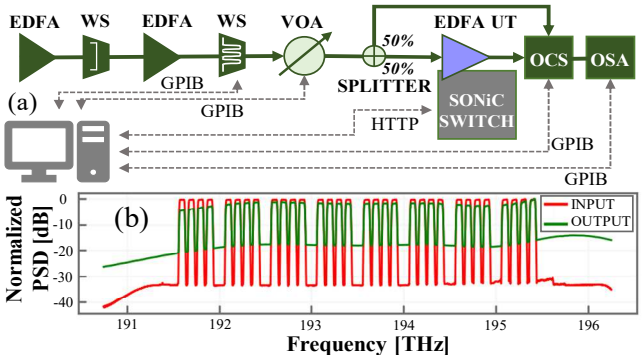


Fig. 3 Setup (a) and an example of collected spectra (b).

III. ML-Models and OSNR Estimation Algorithm

In this article we consider three ML methods: MLP, one dimensional CNN (1D-CNN) and XGB. For the task of estimating the output powers of all the WDM channels $\hat{y}$, the following input features are considered: the input power $P_{ch,IN}$ and the status S_{ch} (i.e., on/off) of all the WDM channels; the total input power $P_{tot,IN}$; the total output power $P_{tot,OUT}$; the target gain G_t; and the pump current C_p. As for the 1D-CNN, an architecture inspired to the one proposed in [12] is considered where a fully connected layer is used to create a larger set of features with locality characteristics, then followed by several 1D-Conv layers with shortcut-like connections for feature extraction and average pooling layers and finally by another fully connected layer to predict targets. To construct MLP and 1D-CNN models, the open-source Keras API from TensorFlow is employed. As for the XGB model, in this paper we employ an open access implementation [13]. Finally, the hyperparameters of the ML-models are carefully chosen through a Bayesian HPO implemented using the open-access Optuna library [14].

Furthermore, we propose an algorithm to estimate the OSNR of the active channels based on the proposed EDFA ML-models. Algorithm I shows the pseudo-code for OSNR estimation which receives as an input a request *rq*, a model DB *ml* and an historical DB *hl*. Once a request to establish or decommission a lightpath is received, the algorithm acts on the EDFA interested by the lightpath (lines 1-2). Then, the expected channel powers at the input of the amplified are retrieved from the request and from the historical DB (line 3). Furthermore, physical parameters can be read directly from the EDFA and used as input features when executing the ML-model (lines 4,5). Then, the signal powers of the active WDM channels $\widehat{P_{ch}}$ are distinguished by the ASE noises $\hat{N}$ of the inactive ones (line 6). At this point, the ASE noises of the active channels $\widehat{N_{ch}}$ are obtained by interpolating over the estimated ASE noise data points and the OSNRs of the active channels are computed (lines 7,8). Finally, the estimated OSNR profiles estimated at the output of the amplifiers are returned (line 9,10).

IV. Illustrative Results

A. Experimental Characterization

Fig. 3 (a) details the testbed employed during the experimental characterization of an EDFA pre-amplifier pluggable module operating in automatic gain control

Table 1 Collected datasets (DS).

DS id	$P_{tot,\,IN}$ [dBm]	G_t [dB]	#Ch Active	Samples	Tilt [dB]
(i)	-26:-10	8:26:2	1:32:1	4,978	/
(ii)	-19:-10	20:29:1	1:32:1	1,045	-3:3:1.5
(iii)	-17.9:-10	17.2:24	4:32:4	760	/

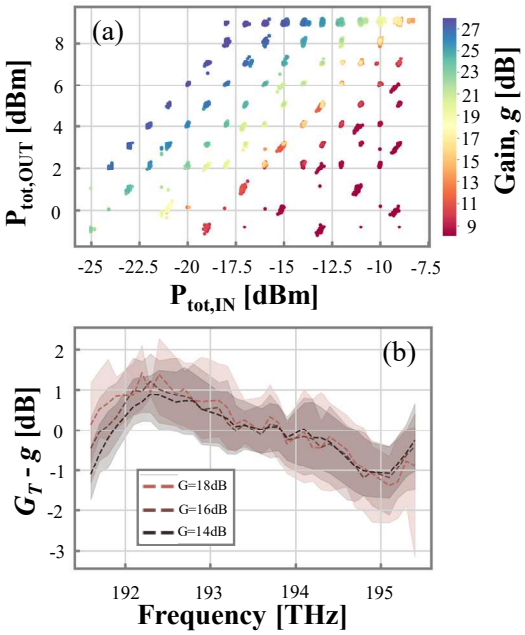


Fig. 4 Power mask (a) and gain ripple (b) of the EDFA UT.

(AGC), that hereafter will be referred as EDFA under test (UT). To create the input channels comb, an EDFA-generated ASE noise was applied to a first Finisar waveshaper (WS) acting as a pass band filter to select the spectrum range of interest, that is then amplified by another EDFA. A second WS was used to shape the noise into WDM channels with 100 GHz channel spacing. Up to 32 active WDM channels were considered. Fig. 3 (b) shows for illustrative purposes the normalized power spectral densities (PSD) collected both at the input (in red) and at the output (in green) of the EDFA UT. As shown in the figure, the channels were organized in 8 sub-bands spaced 100 GHz one from the other assuming the use of band filters to cancel the out-of-band noise generated by coherent transponders. The desired channel-loading configurations were created acting on the WS, while the optical power of the active channels at the input of the EDFA UT was adjusted by a variable optical attenuator (VOA) remotely controlled by GPIB interface. A packet switching device running the SONiC network operating systems was used to host and control the EDFA UT. In particular, the switch was equipped with a docker exposing an HTTP interface that upon GET request retrieved the physical parameters and configured the target gain on the EDFA UT. An optical circuit switch (OCS) enabled to employ a single OSA to capture the spectra both at the input and at the output of the EDFA UT. Several operating conditions were explored in the tri-dimensional hyper-space function of the channel-loading configurations, the input powers and the target gains.

Table 1 details the datasets (DS) that were generated. DS *(i)* considers a wide range of operating conditions and is composed by a large number of samples (i.e., 4978). In

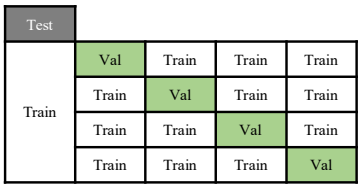


Fig. 5 Hold out 5-fold cross validation scheme.

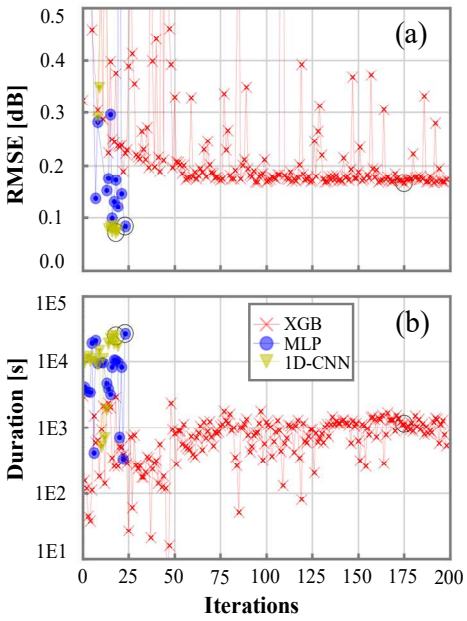


Fig. 6 HPO's iterations as function of RMSE (a) and duration (b).

Table 2 Most import optimal hyperparameters values.

Hyper-parameters	Meanings	Algorithm values		
		XGB	MLP	CNN
Learning rate	step size at each iteration	2.4E-2	4.4E-5	2.9E-4
Estimators	decision trees	9100	N/A	N/A
Gamma	minimum loss reduction to further partition a leaf node	1.9E-7	N/A	N/A
Subsample	subsample ratio of the training instances.	0.95	N/A	N/A
Colsample by tree, level or node	subsample ratio of columns for each tree, level or node	0.92, 0.97, 0.97	N/A	N/A
Min child weight	minimum sum of instance weight needed in a child.	190	N/A	N/A
Layers	number of hidden layers	N/A	3	4
Neurons L	number of neurons in each hidden layer	N/A	256,128 ,256	128,128, 128,128
Kernel Size	Conv layer filter window size	N/A	N/A	4,3,3,3
Optimizer	optimization algorithm	N/A	Nadam	Adamax
Activation	activation function	N/A	'tanh'	'selu'

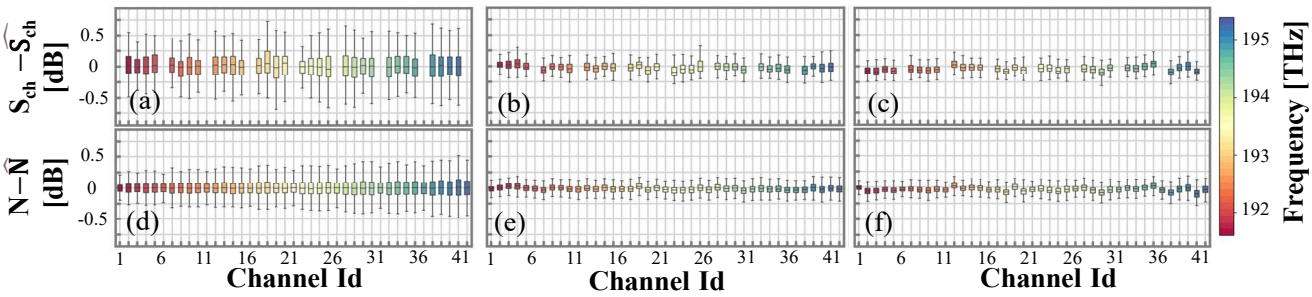


Fig. 7 Channel Power and ASE noise error distributions for XGB(a,d), MLP(b,e) and 1D-CNN(c,f).

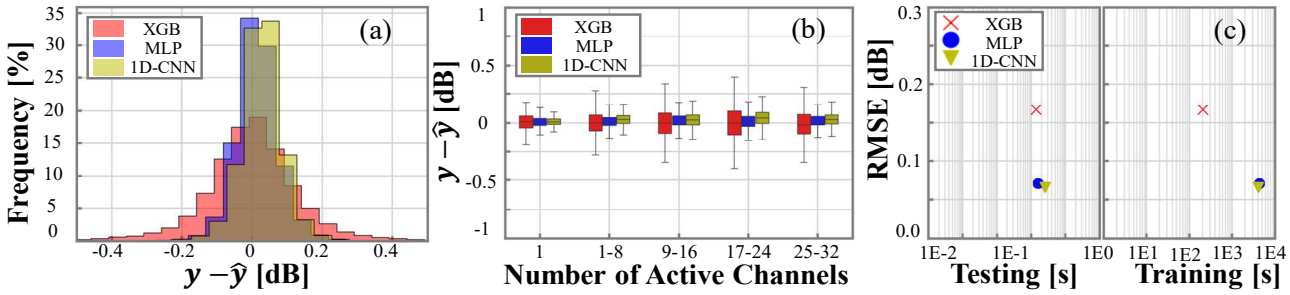


Fig. 8 Error histograms (a), error distribution as function of the number of loaded channels (b) and computation time (c).

particular, 438 channel-loading configurations were explored: 63 were generated deterministically activating and deactivating all the channels one by one while the other 375 were generated randomly with a geometric distribution prioritizing configurations with a low number of active channels. As for total input powers and target gains, we considered those leading to expected channel output powers higher than -3 dBm and lower than 9 dBm. In particular, we considered total input powers ranging from -26 to-10 dBm and target gain ranging from 8 to 26 dB in steps of 2 dB. DS *(ii)* provides fewer samples (1045) and introduces controlled power variations to assess the model's robustness to power imbalances. We examined three specific channel-loading configurations (full, alternating and half empty) and total input powers and target gains obtained as in *(i)* but introducing a tilt in the channel input power ranging from -3 to +3 dB with 1.5 dB increments. Finally, DS *(iii)* has the fewest samples number (760) and simplifies the channel-loading configuration by grouping the channels into sub-bands. In particular, we assumed that the channels of each sub-band are turned on or off jointly and we explored input powers ranging from -17. 9 dBm to -10 dBm and gain targets ranging from 17.2 dB to 24 dB. Fig. 4 (a) shows the power mask defined as the input and output power plane limited by the EDFA maximum and minimum gains and maximum output and minimum input total powers [15]. Note that the considered EDFA pluggable module saturates at a total output power equal to approximately 9 dBm. Moreover, Fig. 4 (b) shows for illustrative purposes the gain ripple profiles (i.e., the difference between the target gain profiles and the measured ones) observed for three target gains. The dashed lines represent the mean gain ripple averaged across all channel-loading configurations, while the shaded areas represent the full range of the measured gain ripple, including the minimum and maximum values.

B. ML-model Construction

We considered three ML methods (i.e., XGB, MLP and 1D-CNN) returning the output power of 41 WDM channels while receiving as input a total of 86 features: the input power and the status of 41 WDM channels and four additional features (i.e., $P_{tot,IN}$, $P_{tot,OUT}$, G_t, and C_p). The considered 41 WDM channels had frequencies ranging from 191.5 to 195.5 THz in accordance with the ITU WDM frequency grid [16]. Note that 9 among the 41 WDM channels were not activated as they were located in the guard-bands between sub-bands.

The proposed ML methods were built based on DS *(i)*. The dataset was partitioned into training (70%), validation (17.5%) and testing (12.5%) set. Fig. 5 details the 5-fold cross validation scheme employed to evaluate the generalization performance of the proposed ML-models. In this scheme, to avoid data leakage two layers of cross-validation are adopted: an outer hold out layer to measure model's performance and an inner loop for hyperparameters tuning.

Fig. 6 (a) shows the root mean square error (RMSE) in dB as function of the number of iterations while Fig. 6 (b) details the duration of each iteration. Note that HPOs of XGB, MLP and 1D-CNN models were processed for a total duration of 47, 48 and 64 hours, respectively. The deep learning methods were trained with a batch size of 32 for up to 5,000 epochs. Early stopping was adopted as regularization techniques to avoid over fitting for both ensemble and deep learning methods. Finally, Table 2 reports the values of the most important hyperparameters determined for the three considered ML-models.

Fig. 7 shows the channel power (a,b,c) and ASE noise power (d,e,f) error distribution boxplots as function of the WDM channel identifier. MLP and 1-D CNN models achieve the best performance with an error in the range of ± 0.25 dB for both the ASE noises and channel power profiles. Differently, XGB model is considerably less accurate than the deep learning approaches with errors exceeding ± 0.5 dB in estimating the channel powers. Fig. 8 (a) shows the normalized error histograms achieved by the considered ML methods. Note the maximum errors observed were 2.87, 0.68

Table 3 Generalization RMSE [dB].

DS id	XGB	MLP	1D-CNN
(i)	0.17	0.07	0.07
(ii)	0.44 ± 0.02	0.36 ± 0.01	0.47 ± 0.02
(iii)	0.29 ± 0.02	0.24 ± 0.01	0.39 ± 0.01

Table 4 RMSE [dB] of TL strategies for the MLP model.

DS id	(a) base model	(b) target model	(c) TL: Fine-tuning	(d) TL: Freezing	(e) TL: Progressive
(ii)	0.36 ± 0.01	0.14±0.03	0.10 ± 0.01	0.18 ± 0.02	0.10 ± 0.01
(iii)	0.24 ± 0.01	0.75±0.07	0.13 ± 0.01	0.15 ± 0.03	0.17 ± 0.02

and 0.44 dB for XGB, MLP and 1D-CNN, respectively. Differently, Fig. 8 (b) contains the boxplot of the error distribution obtained to estimate the power profile (both ASE noise and channel power profiles) as function of the loaded channels. We observe that the XGB models have an error distribution increasing with the number of loaded channels. Finally, Fig. 8 (c) details the RMSE of the considered ML methods versus the time required for the training and for the testing (i.e., inferencing of 622 samples). These results were obtained using Python 3.9 on a workstation with 32 GB of RAM and a GPU RTX A500. Notably, the ensemble method compared to the deep learning ones required one order of magnitude less for the training however performing worse in terms of RMSE.

C. ML-model Verification

We evaluated the generalizability of the models (i.e., their capability to react to previously unseen data) when tested on two target DSs, i.e., DSs *(ii)* and *(iii)*. Table 3 shows the average RMSE and its standard deviation obtained after executing the ML-models on the base DS, i.e., DS *(i)*, and on 20 different random splits of the target DSs. Note that all the considered ML-models had a worse performance on the target DSs than on the base one. Particularly, the ML-models had a lower generalizability on DS *(ii)* than on DS *(iii)* presumably due to the presence of power imbalances. In addition, it has to be highlighted that the MLP outperformed the other ML methods.

Furthermore, TL techniques were applied in order to reduce the generalization error of the MLP model. Note that knowledge transfer is paramount to foster the adoption of ML methods in practical cases when only limited training sets are available. Here we assumed the training sets to be composed by the 10% of the target DSs (i.e., 104 and 76 samples for DSs *(ii)* and (*iii*), respectively).

In particular, we considered the following methodologies: (a) *base model*, i.e., directly applying the base model on the target DS; (b) *target model*, where new target models were trained for the specific target; (c) *fine-tuning*, i.e., re-training the base model with a lower learning rate; (d) *freezing*, where the first layers of the base model were frozen while final ones were fine-tuned; and finally, (e) *progressive learning*, i.e., freezing all the layers of the base model and adding new layers to be trained before finally fine-tuning all the layers. Table 4 shows the average RMSE and its standard deviation obtained considering 20 different random splits of the target DSs. The results show that the application of TL techniques on the MLP model enabled to reduce the RMSE in comparison both to the base and to the target models. In particular, the benefit of TL is particularly evident as compared to directly applying the base model. In addition, the knowledge transfer results beneficial also when compared to training new target models. For these reasons, we conclude that TL enables to effectively generalize a ML model of an EDFA preamplifier for specific cases where the training sets are composed by only few samples.

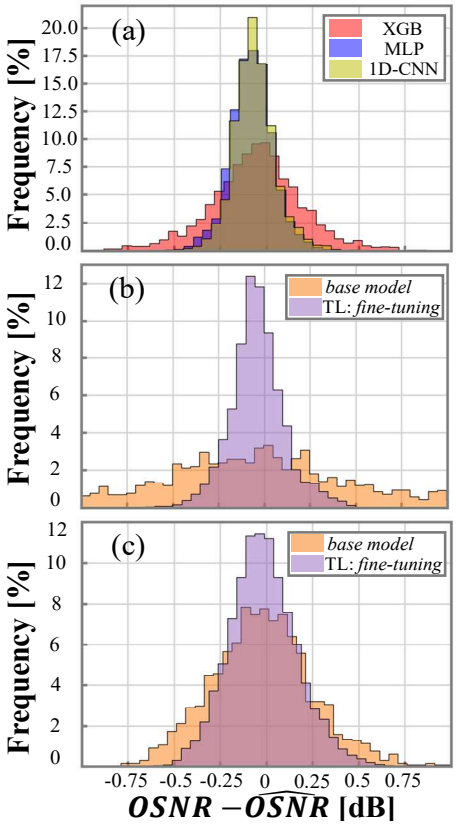


Fig. 9 OSNR estimation error in Algorithm I evaluated offline on DS *(i)* (a), DS *(ii)* (b) and DS *(iii)* (c).

D. OSNR Estimation Algorithm

Finally, we evaluated the performance of the proposed algorithm for OSNR estimation (Algorithm I). Note that the algorithm was evaluated off-line based on DSs *(i)*, *(ii)* and *(iii)*. In this analysis, we considered a request *rq* for a lightpath composed only by one preamplifier module. The input features were loaded directly from the DSs (neglecting lines 3-4 in Algorithm I).

Fig. 9 shows the histogram of the errors obtained estimating the OSNR on the three considered DSs. Specifically, Fig. 9 (a) shows the error distribution obtained on DS *(i)* where a maximum OSNR estimation errors of 2.17, 0.70 and 0.66 dB was observed for XGB, MLP and 1D-CNN, respectively. As expected, deep learning methods exhibited lower errors. Differently, Fig. 9 (b) and (c) compare the error distribution obtained on DSs *(ii)* and *(iii)*, respectively, when adopting the MLP base model (in orange) and when applying fine-tuning on the MLP model (in purple). The use of TL enabled to reduce the OSNR estimation error both on DS *(ii)* and *(iii)*. In particular, the TL benefits are very evident in Fig.

9 (b) due to the high contrast with the base model which exhibited a low generalizability on DS *(ii)*.

V. Conclusions

A preamplifier pluggable module operating in AGC was characterized experimentally for a wide range of input powers, target gains and channel-loading configurations. Based on the collected data, it was possible to evaluate the performances of three ML methods (i.e., XGB, MLP and 1D-CNN) in estimating the power of all the WDM channels at the output of the EDFA. The deep learning methods achieved a higher accuracy compared to the ensemble method however requiring more time for the training.

Moreover, we evaluated the generalization error achieved by the considered ML methods when adopted on unseen data. MLP was shown to achieve a lower generalization error compared to the two other methods. Furthermore, TL techniques were applied on the MLP model to reduce the generalization error showing considerable accuracy improvements on the target datasets.

Finally, we proposed an algorithm for OSNR estimation based on the proposed ML-models which was evaluated offline on the experimental data. Further research is needed to explore the use cases and applications for deploying ML-models in production in IPoWDM systems.

Acknowledgement

The research leading to these results has received funding from the European Community through the MSCA MENTOR (G.A. 956713), the European Union's Horizon Europe research and innovation programmes SEASON (G.A. 101096120) and ALLEGRO (G.A. 101092766) and from the ICREA Institution. Moreover, this research has received partial support from the European Union's project HORSE (G.A. 101096342) and by the NextGenerationEU projects PRIN 2022 (CUP E53D23000760006, identification code 2022MWBFEE001) and SOW (CUP E63C22000970007, identification code CN00000013).

References

1. A. Giorgetti et al., "Enabling hierarchical control of coherent pluggable transceivers in SONiC packet–optical nodes," J. Opt. Commun. Netw., **15**, 163–173, (2023).
2. E. Riccardi, et al., "An Operator view on the Introduction of White Boxes into Optical Networks," J. Light. Technol., **36**, 3062–3072, (2018).
3. D. Wang et al., "Digital Twin of Optical Networks: A Review of Recent Advances and Future Trends," J. Light. Technol. **42**, 4233–4259, (2024).
4. P. Paudyal, et al., "Toward Deployments of ML Applications in Optical Networks," IEEE Photon. Technol. L., **33**, 537–540, (2021).
5. K. Ishii, et al., "Experimental Investigation of Gain Offset Behavior of Feedforward-Controlled WDM AGC EDFA Under Various Dynamic Wavelength Allocations," IEEE Photon. J., **8**, 1-13, (2016).
6. F. da Ros, et al., "Machine learning-based EDFA Gain Model Generalizable to Multiple Physical Devices," ECOC, (2020).
7. Z. Wang, et al., "Open EDFA gain spectrum dataset and its applications in data-driven EDFA gain modeling," J. Opt. Commun. Netw., **15**, 588, (2023).
8. H. Chen et al., "Accurate EDFA gain modelling using convolutional neural network with denoising layers enabled by soft-thresholding," in Proc. ECOC (2023).
9. J. Yu, et al., "Machine learning-based EDFA gain estimation [Invited]," J. Opt. Commun. Netw., **13**, B83, (2021).
10. V. Borisov, et al., "Deep Neural Networks and Tabular Data: A Survey," IEEE Trans. Neural Netw. Learning Syst., **35**, 7499–7519, (2024).
11. R. Shwartz-Ziv, et al., "Tabular data: Deep learning is not all you need," Information Fusion, **81**, 84–90, (2022).
12. Baosenguo, *baosenguo/Kaggle-MoA-2nd-Place-Solution*.", 2021, Available: https://github.com/baosenguo/Kaggle-MoA-2nd-Place-Solution?tab=readme-ov-file
13. Distributed (Deep) Machine Learning Community, *dmlc/xgboost* Available: https://github.com/dmlc/xgboost
14. Optuna, "Optuna - A hyperparameter optimization framework," Available: https://optuna.org/
15. U. Moura et al., "Cognitive Methodology for Optical Amplifier Gain Adjustment in Dynamic DWDM Networks," J. Light. Technol., **34**, 1971–1979, (2016).
16. ITU-T, "Spectral grids for WDM applications: DWDM frequency grid", Available: https://www.itu.int/rec/T-REC-G.694.1-202010-I/en