Neural Networks-Based Equalizers for Coherent Optical Transmission: Caveats and Pitfalls

Pedro J. Freire, *Graduate Student Member, IEEE*, Antonio Napoli, Bernhard Spinnler, Nelson Costa, Sergei K. Turitsyn, *Senior Member, IEEE*, and Jaroslaw E. Prilepsky

(Invited Paper)

***Abstract*—This paper performs a detailed, multi-faceted analysis of key challenges and common design caveats related to the development of efficient neural networks (NN) based nonlinear channel equalizers in coherent optical communication systems. The goal of this study is to guide researchers and engineers working in this field. We start by clarifying the metrics used to evaluate the equalizers' performance, relating them to the loss functions employed in the training of the NN equalizers. The relationships between the channel propagation model's accuracy and the performance of the equalizers are addressed and quantified. Next, we assess the impact of the order of the pseudo-random bit sequence used to generate the – numerical and experimental – data as well as of the DAC memory limitations on the operation of the NN equalizers both during the training and validation phases. Finally, we examine the critical issues of overfitting limitations, the difference between using classification instead of regression, and batch-size-related peculiarities. We conclude by providing analytical expressions for the equalizers' complexity evaluation in the digital signal processing (DSP) terms and relate the metrics to the processing latency.**

***Index Terms*—Neural network, nonlinear equalizer, overfitting, classification, regression, coherent detection, optical communications, pitfalls.**

I. Introduction

MACHINE learning techniques and, more specifically, deep artificial neural networks (NN), are rapidly finding their way into the telecommunication sector, in particular due to their ability to efficiently mitigate transmission and device impairments (see, e.g., [1]–[12]). Indeed, a large number of NN-based techniques have already been proposed and tested for the channel equalization in coherent optical communication systems [9]–[11], [13]–[15].

The NNs are known to be universal approximators that can mimic almost any complex function, provided that the NN structure possesses enough computational capacity and is trained on a sufficient dataset. NNs are especially useful in solving complex nonlinear problems, where the results rendered by more conventional approaches often have limited applicability. The capability of NNs to learn from data, and their inherent adaptability to diverse operating conditions, make them a natural tool for the equalization of fiber-optic channels where the data-carrying signal experiences nonlinear interactions, signal-noise interference, and memory effects. The considerable speed of optical data transmission results in large datasets obtained in a short time, making the optical channel a suitable playground for machine learning techniques. Despite various acknowledged advantages and benefits, there are still challenges and pitfalls that hinder the use of machine learning and NNs, particularly in optical transmission-related tasks. In this paper, we discuss common misunderstandings and misinterpretations that occur when using ML approaches for channel equalization in coherent optical communications.

When applying known NN techniques to optical fiber transmission, the customization and adaptation of these algorithms can be necessary for reaping the full benefits of machine learning.[1] For example, when using NN approaches in image recognition tasks, an accuracy of 99% is typically regarded as "more than superb". Alternatively, this outcome may be viewed with skepticism as being considered too good to be true. In contrast, in optical transmission-related problems, the 99% accuracy – when deciding which bit was transmitted – is typically the minimum required by state-of-the-art transponders operating at pre-forward error correction (FEC) bit error rate (BER) ranging between 10^{-3}–10^{-2}. In modern transponders, this threshold is even higher, e.g., $2 \cdot 10^{-2}$–$4 \cdot 10^{-2}$, and we ought to push the BER tolerance to higher values to improve the operating margin of the system performance. Consequently, we inherently have to impose stricter conditions on our NN architectures: in the case of optical communications, the learning quality must be higher than in the other fields. Unfortunately, efficient NN structures

Manuscript received October 30, 2021; revised April 1, 2022 and April 20, 2022; accepted April 24, 2022. Date of publication May 11, 2022; date of current version June 14, 2022. The work of Jaroslaw E. Prilepsky was supported by Leverhulme Trust under Grant RP-2018-063. The work of Sergei K. Turitsyn was supported by EPSRC Project TRANSNET. This work was supported by the EU Horizon 2020 Program under the Marie Sklodowska-Curie Grant agreement 813144 (REAL-NET). *(Corresponding author: Pedro J. Freire.)*

Pedro J. Freire, Sergei K. Turitsyn, and Jaroslaw E. Prilepsky are with the Aston Institute of Photonic Technologies, Aston University, B47ET Birmingham, U.K. (e-mail: p.freiredecarvalhosourza@aston.ac.uk; s.k.turitsyn@aston.ac.uk; y.prylepskiy1@aston.ac.uk).

Antonio Napoli and Bernhard Spinnler are with Infinera R&D, 81541 Munich, Germany (e-mail: ANapoli@infinera.com; bspinnler@infinera.com).

Nelson Costa is with Infinera Unipessoal, 2790-078 Carnaxide, Portugal (e-mail: NCosta@infinera.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSTQE.2022.3174268.

Digital Object Identifier 10.1109/JSTQE.2022.3174268

[1]Similarly, when the telecommunication industry began to use DSP techniques – already employed in wireless – in optics, adaptions were also required.

that meet such ultra-high accuracy requirements are often more vulnerable to various problems and pitfalls.

The severe overfitting that is frequently encountered while dealing with optical channel equalization is a critical element addressed in this work. Furthermore, because we are dealing with a high-accuracy problem, local minima can play an important role, since the NN can stop learning if the current local minima yield an almost zero gradient. Arguably, the overfitting and local minima are the main issues capping the equalization capacity of the NN-based DSP elements (when we neglect other technical issues such as the high complexity of NN-based components). Generally, overfitting occurs when models or procedures that violate a "parsimony principle" are used. For example, this happens when more elements than required by the task at hand or more complicated approaches than necessary are employed. In other words, it occurs "when using a model that includes irrelevant components (excessive degrees of freedom)" [16].

Overfitting in optical channel equalization is the state in which an NN starts to interpret the structure of the training set with such detail that it effectively models the noise or some spurious periodicity present in the transmission data instead of identifying the true inverse transfer function (the deterministic part) of the channel. In this context, the "barriers" that preclude the NN learning process should be considered with a wider meaning than just the well-known performance gap between the training and validation curves. Here we show that learning the noise characteristics in the training dataset, typically produces the "jail window" pattern in equalized 2D constellations, and this effect degrades the learning *while we can still have a small gap between the training and testing loss curves*. This result indicates that the manifestation of overfitting and local minima in our problems can be intricate and unusual. Additionally, a small gap (between training and validation) can be observed when the NN learns the possible periodicity in the dataset, but it is a clear deviation from the "true purpose" of the NN-based equalizer.

Another major issue that is frequently identified in the literature, is the overestimation of the performance and computational complexity (CC) of the NN equalizers. We devote the entire Sec. VIII to discuss this. Therein, we show that the pseudo-random bit sequence (PRBS) order and digital-to-analog converter (DAC) memory may lead to an overestimation of the system performance. Furthermore, we also show that CC can be misinterpreted and exaggerated if certain points are overlooked. In this case, we demonstrate that the number of NN parameters is not a precise indicator of the true CC. In addition, when pruning and quantizing an NN-based equalizer, special care must be taken to how the NN is pruned and quantized; otherwise, overestimation can occur in the CC analysis.

In summary, we investigate typical caveats and pitfalls that may occur when using machine learning-based methods for impairment compensation in coherent optical communication systems. We introduce and discuss the most widely used metrics to quantify the performance and complexity of NN-based equalizers. Our main goal is to present some form of a guide, providing the reader with an intuitive understanding of the key pitfalls when using NNs in optical communications. We illustrate the potential issues that can occur when using some current practices proposed in the literature in this field. We also hope that this paper can serve as an introduction to this fast-growing field of high practical importance.

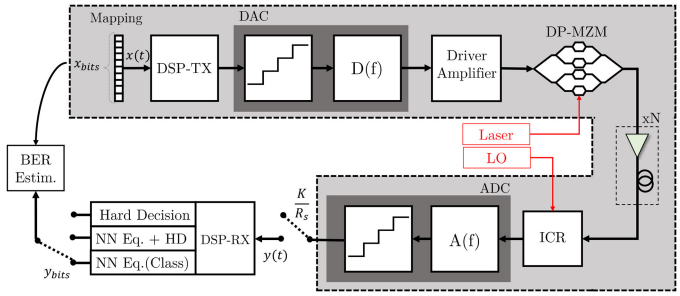


Fig. 1. Experimental setup used to analyze the performance of NN equalizers; $D(f)$ is the DAC electrical transfer function, $A(f)$ is the ADC electrical transfer function, and $x(t)$ and $y(t)$ are the transmitted/received time sequences of symbols, respectively. The input of the NN is the soft output of the regular DSP just before the decision module.

II. Experimental and Numerical Setup and NN-Based Equalizer Characteristics

Our results will be further predominantly demonstrated and confirmed by the data obtained in the extensive numerical modelling of rather general transmission systems. However, we also verified key conclusions using the (relatively) smaller set of the experimental data for specific set-ups. We would like to stress that our conclusions, findings, and design propositions are quite general and apply to various similar optical transmission systems.

The setup used in our experiment is depicted in Fig. 1. At the transmitter (TX) side, a dual-polarization (DP) 16-quadrature amplitude modulation (16-QAM) 34.4 GBd symbol sequence was mapped out of data bits generated by a $2^{32}-1$ PRBS. Then a digital root-raised cosine (RRC) filter with a roll-off factor of 0.1 was applied to limit the channel bandwidth to 37.5 GHz. The resulting filtered digital samples were resampled and uploaded to a DAC operating at 88 GSamples/s. DAC outputs were amplified by a four-channel electrical amplifier that drove a Mach-Zehnder modulator in the phase/quadrature of DP, modulating the continuous waveform carrier produced by an external cavity laser at $\lambda = 1.55\ \mu$m. The resulting optical signal was transmitted over the 5×50 km spans of standard single-mode fiber (SSMF) with an erbium-doped fiber amplifier (EDFA) only. The EDFA noise figure was in the 4.5 to 5 dB range. The parameters of the SSMF at $\lambda = 1.55\ \mu$m, are: attenuation coefficient $\alpha = 0.21$ dB/km, dispersion coefficient $D = 16.8$ ps/(nm $\cdot$ km), and effective nonlinear coefficient $\gamma = 1.2\ (\text{W} \cdot \text{km})^{-1}$.

At the receiver (RX) side, the optical signal was converted into the electrical domain using an integrated coherent receiver. The obtained signal was sampled at 50 Gsamples/s by a digital sampling oscilloscope and processed by an offline DSP based on the algorithms described in [17]. First, the bulk accumulated chromatic dispersion (CD) was compensated using a frequency domain equalizer, which was followed by the removal of the carrier frequency offset. Then a constant-amplitude zero

autocorrelation-based training sequence was located in the received frames and the equalizer transfer matrix was estimated from it. This equalizer transfer matrix was applied to the signal to remove the remaining linear distortions (due to polarization mode dispersion (PMD), residual CD, filtering), and to demultiplex the signal polarization. The processed signal was corrected for clock frequency and phase offsets. The carrier phase estimation was then achieved with the help of pilot symbols. Thereafter, the resulting soft symbols were used as input for the NN equalizers. Finally, the pre-FEC BER was evaluated from the signal at the NN equalizer output.

A simulator was implemented aiming at mimicking the ideal transmission setup, i.e., using ideal TX/RX components. As a consequence, only impairments resulting from fiber propagation and amplified spontaneous emission (ASE) noise originating in the EDFA are considered in the numerical simulations.[2] The propagation of the optical signal along the optical fiber was simulated by solving the Manakov equations using the split-step Fourier method (with a resolution of 1 km per step) [18]. Each fiber span was followed by an optical amplifier with the noise figure NF $= 4.5$ dB, which fully compensates for fiber losses and adds ASE noise. At the receiver, after full electronic CD compensation (CDC) by the frequency-domain equalizer and downsampling to the symbol rate, the received symbols were normalized to the transmitted ones.

The BER was estimated from the received symbols following three different approaches: i) applying a hard decision strategy only; ii) first, equalizing the signal using an NN equalizer in the regression task and then applying a hard decision, or iii) using an NN classifier (the classification is used only in the section where we compare its performance with the regression). These approaches are schematically depicted in Fig. 1. The NN input mini-batch shape can be defined by three dimensions [14]: $(B, M, 4)$, where B is the mini-batch size, M is the memory size defined through the number of neighbors N as $M = 2N + 1$, and 4 is the number of features for each symbol, referring to the real and imaginary parts of two polarization components. The objective of NN is to recover the real and imaginary parts of the k-th symbol in one of the polarizations, so that the shape of the NN output batch can be expressed as $(B, 2)$.

In general, for the regression task – within the NNs considered in this paper, we incorporate the mean squared error (MSE) loss function estimator and the classical Adam algorithm for the stochastic optimization step with the default learning rate set to 0.001 [19]. Standard training was carried out for up to 1000 epochs with a batch size of 2048, which has proven to be high enough to reach convergence for the considered transmission scenarios. Furthermore, the total dataset used consisted of 2^{18} symbols for the training dataset, 2^{18} symbols for the validation dataset, and 2^{18} independently generated symbols for the testing phase. All three are generated with a different random seed. The training dataset is used to update the weights of our NN model; the validation one is to monitor the overfitting of the learned model and to trigger the early stopping, and the testing one gives us the final measurement with a never-seen dataset after the training has been done. In this paper, all results showing the Q-factor after equalization are obtained by using the testing dataset, and all results that show the Q-factor over the epochs, are obtained by using the validation datasets.

[2]We considered the propagation of a single-channel signal filtered by an RRC filter with 0.1 roll-off factor and with an upsampling rate of 8 samples per symbol over different transmission systems.

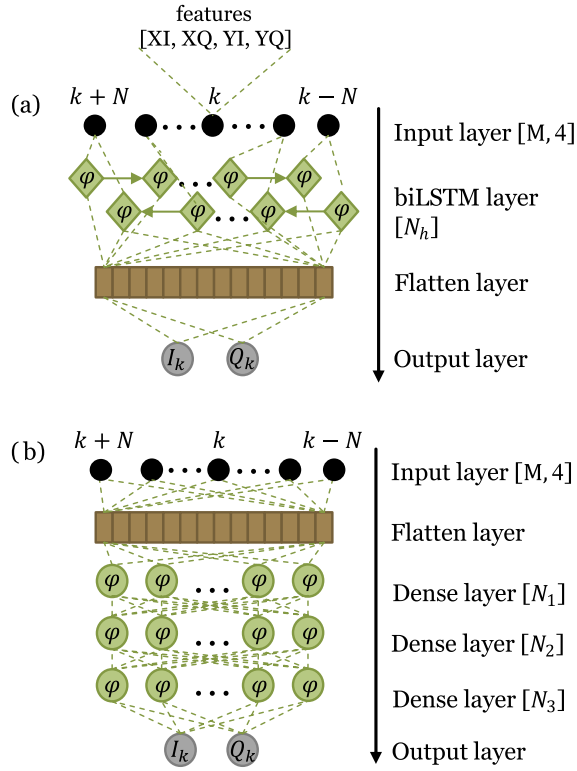


Fig. 2. The schematics of different NN architectures considered in our paper: (a) biLSTM equalizer with N_h hidden units and (b) MLP equalizer having three hidden layers, with N_1, N_2, and N_3 neurons in each consecutive layer, respectively. The input has a memory equal to $M = 2N + 1$ and 4 features representing the real (I) and imaginary (Q) parts of both X and Y polarizations. The function φ represents the activation function which in our case is the "tanh".

The training dataset was shuffled at the beginning of every epoch to avoid overfitting caused by learning the connections between the neighboring training pairs [5]. All simulated datasets were generated using the Mersenne twister generator [20] with different random seeds, which guarantees a cross-correlation below 0.004 between the training and testing datasets, meaning that the symbols are virtually independent.

Finally, since the goal of this work is to demonstrate possible pitfalls and overestimation scenarios of the NN-based equalizers, we tried to use the same architecture and hyperparameters throughout the paper. In some sections, those parameters are altered for some specific purpose that we will be clearly highlighted. In general, when using the multilayer perceptron (MLP), we considered three hidden layers with $[N_1 = 481/N_2 = 31/N_3 = 263]$ neurons in each layer, respectively. When using the bidirectional long-short-term memory (biLSTM) equalizer, the number of hidden units (N_h) was set to 226. Both NN models are illustrated in Fig. 2. The standard number of taps used was $N = 25$: this is the maximal memory

size estimation for the scenarios that we will address. The memory effect is important because, even though we compensate for the effect of CD electronically, we still have to mitigate the impact of the coupling between the nonlinearity and the CD, which occurs along with the fiber transmission. Since the aim of this work is not to propose a new NN equalizer, we do not focus on optimizing the NN structure for different transmission setups [14]. In contrast, we used the same NN structure in different scenarios. Nonetheless, the chose the NN structure to be complex enough to be capable of dealing with the different levels of nonlinearity while allowing us to clearly demonstrate various caveats and pitfalls that can occur in the coherent optical channel equalization task.[3]

III. Quality of Transmission Metrics

There exists a variety of performance indicators that can be used to assess the quality of M-ary QAM transmission systems: bit error rate (BER), mutual information (MI), Q-factor, signal-to-noise ratio (SNR), effective SNR, and error vector magnitude (EVM), to mention the most used examples.

The ultimate quality of transmission (QoT) metric in digital communications is the post-FEC BER or the Q-factor,[4] because all real transmissions occur with close to 0 post-FEC BER. The MI can also be a highly valuable metric because it estimates the achievable spectral efficiency when matched with the post-FEC BER. When the decoder is not available (e.g. when measuring channel equalization using received soft symbols), the post-FEC metric is often omitted in favor of the pre-FEC metric, which is a common performance measure for an uncoded system.

The pre-FEC BER can vary depending on the decision technique (hard or soft decision, HD and SD) utilized after the equalization of the soft symbols. This metric accurately predicts the post-FEC BER for HD-FEC with optimal interleaving. Adopting such a metric, however, can result in an incorrect spectral efficiency estimate, which is especially evident at low code rates (see [22] for details). However, when dealing with NN equalizers, the pre-FEC BER/Q is commonly used because obtaining the post-FEC BER/Q would require integrating the equalized symbols into the rest of the DSP chain, which is not worth the work for initial performance evaluations. The SNR and EVM are the least accurate QoT measurements. However, these two measurements can also be utilized to provide a qualitative comparison of different transmission regimes. It is worth noting that these metrics are related to channels with known statistics, such as the additive white Gaussian noise (AWGN) channel. The correlations between the different QoT metrics in nonlinear channels, on the other hand, are not always known, and adopting extrapolations based on linear theories should be done with caution.

Although the pre-FEC BER is an important QoT metric, it does require the transmission of a known pattern, such as a training sequence, through the system for continuous performance monitoring.[5] As a result, the effective SNR (ESNR) and EVM gained popularity insofar as they lent themselves well to the study of unknown symbol sequences. Furthermore, as previously stated, these QoT metrics can still provide an accurate estimation of the BER when the system errors are primarily caused by optical AWGN, i.e., when fiber transmission is close to the linear regime and nonlinear effects (arising in TX/RX components and non-additive noise) are almost negligible. For such metrics, it is inherently assumed that the reception is non-data-aided and that a quadratic-QAM signal constellation is used. The ESNR, EVM, and Q-factor can be calculated using the following expressions that are valid for a Gaussian-distributed signal [23]–[26]:

$$\mathrm{EVM}_{RMS} = \left[\frac{1/N\sum_{n=1}^{N}|y_n - x_n|^2}{1/N\sum_{n=1}^{N}|x_n|^2}\right]^{\frac{1}{2}}, \tag{1}$$

$$\mathrm{SNR} \approx \left[\frac{1}{\mathrm{EVM}_{RMS}}\right]^2, \tag{2}$$

$$Q = \sqrt{2}\,\mathrm{erfc}^{-1}(2\mathrm{BER}), \tag{3}$$

where y_n is the normalized n-th symbol in the stream of measured symbols, x_n is the ideal normalized constellation point of the n-th symbol (i.e. a symbol from the M-QAM alphabet), N is the total number of symbols in the constellation, and erfc^{-1} is the inverse complementary error function. We typically consider these metrics in dB, using the relation: $T[\mathrm{dB}] = 20\log_{10}(T)$, where T is the QoT metric under investigation.

From a statistical perspective, the BER depends on the particular decision mechanism utilized at the receiver side, whereas the MI gives the effective transmission capacity regardless of the decision process. However, because we work with the received soft symbols, the MI cannot be conveyed explicitly. One option to derive the MI value is to estimate it by assuming a single-input single-output AWGN channel, which yields a suboptimal estimate giving out the MI's lower bound. This lower bound to the MI, $I(X;Y)$, can be expressed as [27]–[29]:

$$I(X;Y) = \mathbb{E}\left[\log_2\left(\frac{p(y|x_k)}{\sum_{i=0}^{MF-1}p(i)p(y|x_k)}\right)\right], \tag{4}$$

where $p(i)$ is the probability distribution of each k-th QAM alphabet symbol, and $p(y|x_k)$ defines the conditional probability of the received constellations given the k-th QAM input symbol. Then we can use the multivariate Gaussian distribution estimator [30], [31] to calculate $p(y|x_k)$ of the transmitted-received complex symbols, which, ultimately, gives as the lower bound for the MI via (4).

Now, as we have established the framework for the most relevant QoT metrics and addressed the assumptions used in their computation, we will look at how those metrics may be used to evaluate the performance of the NN equalizer. The main purpose of this section is to raise awareness of the fact that,

[3]As discussed in [11], [14], the step of hyperparameter optimization is crucial on the process of designing a NN equalizer with good performance, and to tackle this we use and suggest the usage of the Bayesian Optimization tool in [21], which is more efficient than other traditional power-hunger techniques.

[4]BER entirely defines the Q-factor in this case, and we use it to employ dB-s instead of a linear scale.

[5]Note that in practical terms, the pre-FEC BER is usually derived from the post-FEC BER assuming that the FEC can correct all errors.

TABLE I
SUMMARY OF THE BEST PERFORMANCE AFTER 1000 TRAINING EPOCHS FOR DIFFERENT QOT METRICS AND TRANSMISSION SETUPS. (SIMULATION RESULTS OF SINGLE-CHANNEL TRANSMISSION)

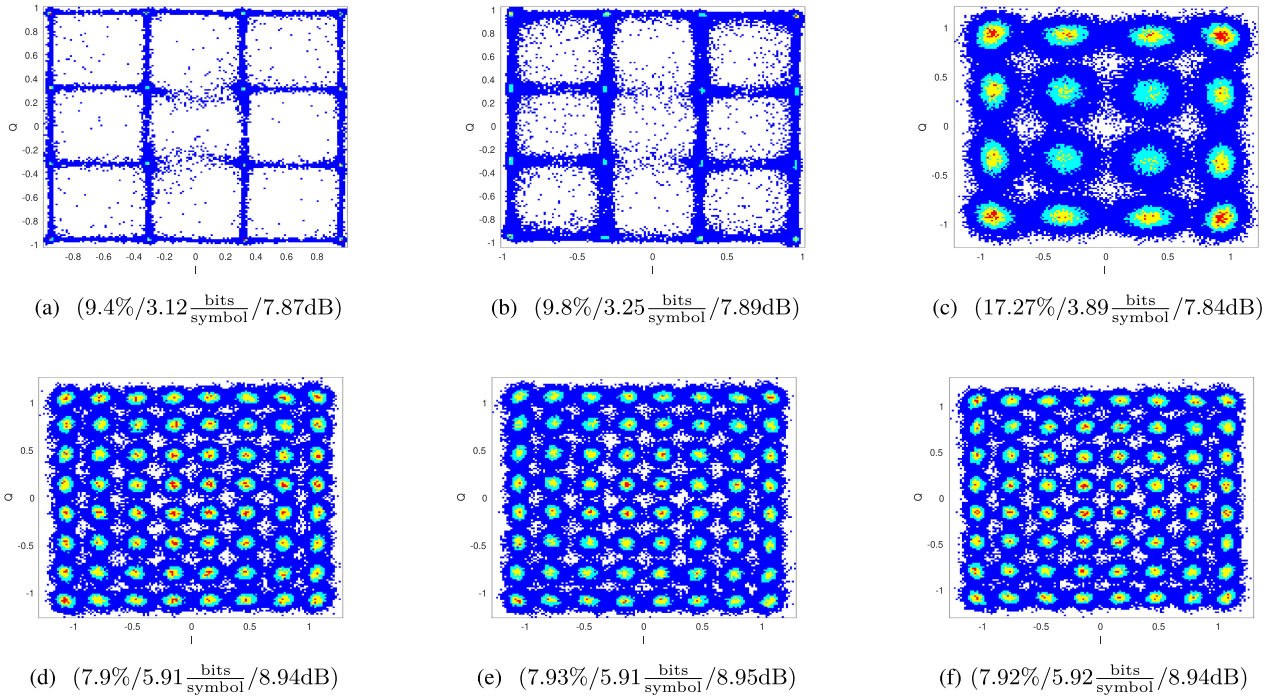
Scenarios	MI [bits/symbol]			EVM [%]			Q-factor [dB]		
	Ref	biLSTM	MLP	Ref	biLSTM	MLP	Ref	biLSTM	MLP
Case i) 20×80km; SSMF; 5dBm; 8-QAM; 34.4GBd	2.93	2.99	2.90	22.7	5.9	8.3	7.82	10.72	8.37
Case ii) 15×100km; SSMF; 4dBm; 16-QAM; 28GBd	3.83	3.99	3.89	19.7	7.4	9.4	7.42	10.48	7.89
Case iii) 30×50km; SSMF; 5dBm; 16-QAM; 64GBd	3.84	3.97	3.87	19.4	8.1	10	7.48	9.15	7.65
Case iv) 10×60km; SSMF; 3dBm; 64-QAM; 30GBd	5.68	5.98	5.92	10.4	5	7.9	7.1	12.75	8.95

depending on the modulation format, the NN structure, and the level of noise, the NN can produce the effect of *squeezing the constellations:* we name this effect the "jail window" pattern. We notice that numerous other works [1], [32]–[38] have also reported the "jail window" type constellations after the NN-based equalization, often without paying attention to the true meaning and consequences of this phenomenon. We clarify this effect and explain possible related misinterpretations. The effect of "jail window" results from the regression task carried out by NN equalizers based on the MSE loss; the effect occurs because the ultimate goal of such NNs is to minimize the Euclidean distance between recovered and transmitted symbols. However, we emphasize that the "jail window" constellation forms *violates the Gaussian channel assumption used in the computation of some metrics mentioned above*. Indeed, this effect reduces the accuracy of all Gaussian-assumption-based metrics (e.g., the ESNR) besides the Q-factor calculated directly from the BER obtained via direct error counting. Thus, when using these inaccurate Gaussian-assumption metrics, we can obtain false results indicating that the NN performs well while, in reality, the true gain provided by the "jail window" constellation is highly overestimated. The "jail window" effect can also be explained by the mismatch between the true transmission performance metric (the BER) and the metric that is minimized by the NN training (the MSE loss), such that we have a disagreement between the objective function and the actual NN result; however, the BER itself cannot be used as a NN loss function inasmuch as it is non-differentiable. Further investigation of the effect of the "jail window" is given further in Section VII.

To illustrate the metric-related problem, we have tested two types of equalizer, biLSTM and MLP, in four different numerical transmission setups: i) 20×80 km SSMF when transmitting 8-QAM at 34.4 GBd; ii) 15×100 km SSMF when transmitting 16-QAM at 28 GBd; iii) 30×50 km SSMF when transmitting 16-QAM at 64 GBd; iv) 10×60 km SSMF when transmitting 64-QAM at 30 GBd; the launch power was set to 3 dB higher than the power level for the best performance without NN equalizer. Clearly, when a nonlinear equalizer is employed, the optimal launch power should increase. The scenarios and NN-equalization results for different metrics are summarized in Table. I.

Here, we note that Table I shows the best value of each QoT metric after running the process over 1000 epochs, but the best value did not occur at the same epoch number for all metrics. For example, in case ii), 15×100 km SSMF with 16-QAM at 28 GBd, we observed that, for the epoch corresponding to the lowest EVM (9.4%), the MI and Q-factor values were, respectively, 3.12 bits/symbol and 7.87 dB; for the epoch corresponding to the highest MI (3.89 bits/symbol), the EVM and Q-factor values were, respectively, 17.27% and 7.84 dB; and for the epoch leading to the highest Q-factor (7.89 dB), the MI and EVM values were, respectively, 3.25 bits/symbol and 9.8%. This result clearly highlights that the particular QoT metric selected for performance optimization *does impact the eventual result for the system with equalizer* and that the differences can be even more accentuated if another transmission setup/ NN architecture is used. Indeed, depending on the quality metric, the constellation after the equalization changes. Fig. 3 shows the constellation corresponding to the maximum of each metric for cases ii) and iv), where, in the first case, the "jail window" pattern does appear, but in the second it does not. For case ii), we can see the "jail window" with thin lines connecting the constellation points for the best EVM. The same "jail window," but with somewhat thicker lines, can be observed in the case leading to the best Q-factor. Contrarily, the traditional Gaussian-type constellations are observed in the case leading to the best MI. For case iv), the NN did not generate the "jail window" pattern. In this case, the minimum in the EVM and the maximum in the MI and Q-factor were achieved at approximately the same epoch. It is important to note that the Q-factor in all three cases did not change significantly, but the improvement in the EVM with the "jail window" caused a high decrease in the MI estimation from 3.89 bits/symbol (the epoch leading to the best MI) down to 3.12 bits/symbol (the epoch leading to the best EVM). This result is the consequence of the non-Gaussian shape of the constellation obtained, which violates the applicability conditions of (4) used to measure the MI. Furthermore, Fig. 3 points to one of the repercussions of the "jail window" pattern: the NN continued to minimize the Euclidean distance between the prediction and the labels, but this process no longer decreases the BER, and so we departed from the main goal of equalization. Our observation was that after the epoch of the highest MI (Fig. 3(c)), the NN began to converge to the "jail window" and stopped further improving the Q-factor, which is why Fig. 3(a) and (c) have roughly the same BER / Q-factor but Fig. 3(a) has a much smaller EVM than Fig. 3(c).

This effect shows that the usage of QoT metrics mentioned above for the regression task can be misleading, resulting in an underestimation of the true achievable MI and converging to a non-optimal equalizer's structure. Ultimately, in Fig 4 we present the evolution of those three metrics over the epochs for



(a) $(9.4\%/3.12\frac{\text{bits}}{\text{symbol}}/7.87\text{dB})$ (b) $(9.8\%/3.25\frac{\text{bits}}{\text{symbol}}/7.89\text{dB})$ (c) $(17.27\%/3.89\frac{\text{bits}}{\text{symbol}}/7.84\text{dB})$

(d) $(7.9\%/5.91\frac{\text{bits}}{\text{symbol}}/8.94\text{dB})$ (e) $(7.93\%/5.91\frac{\text{bits}}{\text{symbol}}/8.95\text{dB})$ (f) $(7.92\%/5.92\frac{\text{bits}}{\text{symbol}}/8.94\text{dB})$

Fig. 3. Constellation diagrams after the MLP-based equalization, corresponding to the best performance when using the different QoT metrics. (a) and (d) lowest EVM; (b) and (e) best Q-factor; (c) and (f) highest MI. The simulated transmission scenarios are: 15×100 km, 4 dBm, 28 GBd 16-QAM (a)–(c); 10×60 km, 3 dBm, 30 GBd, 64-QAM (d)–(f). In each constellation, the QoT metrics were presented as (EVM/ MI/ Q-factor).

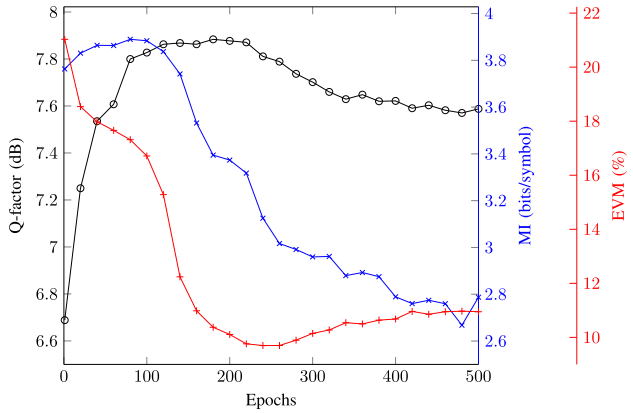


Fig. 4. Evolution of Q-factor (simulations), MI, and EVM over the training epochs when using the MLP equalizer in the 15×100 km, 4 dBm, 28 GBd 16-QAM system.

case ii), to visualize the previous statements, where we can see that the overall behavior and the best values are different for different metrics.

Table I also demonstrates that the NN equalizer may lead to significant improvement of the EVM when compared to the linear equalization only, but without rendering the same corresponding improvement in the Q-factor. To better illustrate this effect, we can use the following expression to estimate the BER from EVM after the equalization [39]:

$$\text{BER} = \kappa \frac{1 - M^{-1/2}}{1/2 \log_2(M)} \text{erfc}\left[\sqrt{\frac{3/2}{(M-1)\text{EVM}_{RMS}^2}}\right], \quad (5)$$

where M is the cardinality of the modulation format and κ is the correction factor. Now, we estimate how much the result obtained through (5) deviates from the one calculated through the direct bit error counting. We start by using the EVM and BER before the NN to calculate the correction factor κ: for case ii), the correction factor is $\kappa = 1.076$, which means that the BER calculated via (5) using the reference value of the EVM (before the NN equalization) is a suitable QoT estimator, almost matching the true BER value after the respective conversion. However, this equation shows how overestimated the EVM can be if the "jail window" is present after the NN equalization. For the MLP equalizer, by using (5) for case ii), we obtain a Q-factor estimate of 13.62 dB while, in reality, it is only 7.89 dB. Additionally, in case iii), the Q-factor estimated using (5) is 13 dB, whereas the true Q-factor is just 7.65 dB. However, in case iv), where the "jail window" is absent (see Fig. 3) the Q-factor estimated through the EVM is equal to 9.4 dB, whereas the true one is 9.2 dB, showing a good match. Interestingly, even in cases where the "jail window" is absent, the Q-factor calculated through the EVM can be considerably overestimated. For instance, in case ii) and using the biLSTM equalizer, the Q-factor estimated through the EVM is 15.6 dB, whereas the true value is 10.7 dB.

Finally, the results in Table I also show that when using the biLSTM equalizer, the "jail window" phenomenon occurs rarer compared to when we use the MLP equalizer. However, it can still persist for the biLSTM in some scenarios, as will be pointed out later in section VII. Therefore, we can conclude that to avoid the overestimation of the system's performance, the results should be presented in terms of BER or Q-factor derived

through (3). Otherwise, the constellations after equalization should be assessed to ensure that it is still approximately Gaussian so that the other QoT metrics provide a good approximation of the true performance.

IV. PRBS Order Impact in Systems' Performance

There has been a plethora of empirical evidence that deep enough NNs can memorize random labels, even when using considerably large datasets [40]. Thus, in principle, by sufficiently increasing the size of a NN, we can always reduce the training error to small enough values, even though the task of learning a completely random sequence is meaningless.

Unfortunately, when transmitting the commonly-used PRBS (say, of orders 7, 9, 11, 15, 20, and 23), the benefits rendered by the NN-based equalization can be overestimated. Indeed, the NN may learn the PRBS generation rules themselves, instead of estimating the inverse of the transmission channel model, and this naturally results in a sharp equalizer's performance degradation when truly random data (say, obtained from live traffic) is transmitted. This overfitting effect is particularly relevant because the NN-based equalizers are often trained using PRBS-based datasets, even in experiments.

When dealing with NNs, two key PRBS-related issues must be addressed. First, the PRBS has a periodicity that is determined in terms of symbols by the PRBS order and the modulation format cardinality. The order 20 PRBS, for example, has a period of $2^{20}-1$ bits, and if we use a 64-QAM signal with 6 bits per symbol representation, the symbol periodicity will be around 174 k symbols.[6] To avoid NN learning such a pattern in this circumstance, the size of the training dataset used to train the NN must be less than the aforementioned quantity. This topic was investigated in [5], where an MLP classifier was used to check the overestimation of PRBSs of orders 7 and 15 in the 4-pulse amplitude modulation (PAM) transmission system using a large training dataset of size 2^{19}. According to the findings of that work, the NN was able to produce a reasonable result when tested on another PRBS sequence, but when tested on a fully random signal, the system's performance degraded dramatically because the NN did not learn the channel equalization but instead learned the PRBS periodicity.

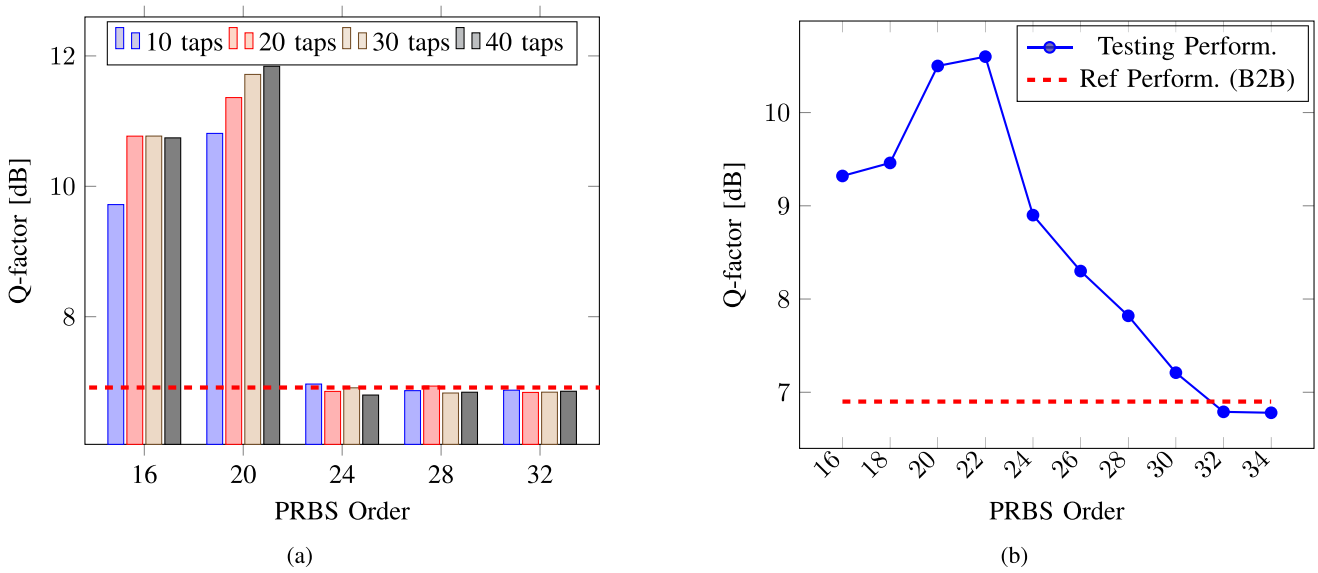
A very simple and popular approach for the generation of PRBS is to use a linear feedback shift register (LFSR) with particular initialization and feedback. This approach can lead to a serious limitation since, if the NN's input is broad enough to catch the whole inputs used by the LFSR to construct the current symbol (typically its size is equal to the memory), the NN will be able to learn the PRBS model properties instead of performing the genuine nonlinearity mitigation task. More recently, in [41], this issue was mathematically studied and tested for an MLP classifier for both on-off keying (OOK) and 4-PAM transmission. In that Ref., the authors used the PRBS of order 20 and generated 2^{19} bits, a sequence that is shorter than the generator periodicity. The bits were converted into symbols and fed into the NN classifier. The authors then created an input with the neighbor symbols but removed the current symbol to be classified. If the current symbol cannot be learned from adjacent symbols, the NN cannot correctly decide which bit was transmitted, and this situation corresponds to BER $= 0.5$. Otherwise, the BER should be significantly lower than 0.5, indicating that the NN can understand the symbol generation and mapping rules. In [41], the authors found that, in the case of OOK signals, the input memory length of 17 was enough to recover the symbols. In the case of transmission of 4-PAM signals, the same behavior was observed once the input had 9 taps. In the case of 4-PAM, some fewer taps were needed because each symbol carries 2 bits, whereas the OOK signals carry only 1 b per symbol. In order to avoid the issues resulting from the limited PRBS length, instead of LSFR, the Mersenne Twister random sequence (MTRS) generator should be used, which can provide the sequences with a much longer period than that of the LFSR.

In this section, differently from the two previous works [5], [41], we evaluate both aforementioned issues using the recurrent equalizer (biLSTM). The biLSTM's likelihood of learning the deterministic time correlation due to the PRBS order is by far superior to that of the MLP equalizer. Additionally, we consider the transmission of a 64-QAM modulation format, which increases the number of bits per symbol, therefore enhancing the PRBS-related problems in the NN-based equalization.

We generated data from 10 transmission runs with PRBS orders 16, 18, 20, 22, 24, 26, 28, 30, 32, and 34. AWGN was added to the RX input in a back-to-back scenario, so that all data sets after the hard decision had the same Q-factor (6.9 dB). Since the only source of signal degradation is a random noise coming from the AWGN added, the NN should not provide any performance improvement, as the NN is a nonlinear deterministic function. Hence, any Q-factor improvement when employing the NN results from the NN's learning of the PRBS generation rule.

We perform two tests to evaluate the impact of the PRBS order on the performance of NN. First, we study the impact of the training data set and PRBS symbol periodicity. Fig. 5 shows the Q-factor as a function of the PRBS order when 2^{18} symbols are used to train the NN. A different seed is used to generate another sequence of 2^{18} symbols (with the same PRBS order), which are used to test the NN. The analysis of Fig. 5 shows a clear improvement in the Q-factor for PRBS orders 16 and 20. This result confirms that the NN can learn the PRBS periodicity when the training data sets (2^{18} symbols) are larger than the symbol periodicity. Moreover, Fig. 5 also shows that increasing the number of taps enables achieving an even better Q-factor (the NN could train faster); this behavior was also observed in [41]. For the PRBS orders higher than 24, the training data set becomes smaller than the symbol periodicity, and, consequently, the NN is no longer capable of learning the PRBS generation rules: the values drop below the threshold (the dashed line).

[6]The true period of symbols will be 6 times longer in this case, because this is when the bit sequence is aligned to the symbol boundaries. However, because the NN maps the input symbols onto a multidimensional space, the NN may also trace the information on bit periodicity. As a result, we define symbol periodicity as the number of symbols that contain the entire bit periodicity.



(a) (b)

Fig. 5. Q-factor's dependencies on the PRBS order for (a) the different values of input memory (the different number of taps is highlighted with different bar's color) and different random seeds for training and testing with dataset sizes of 2^{18} symbols; (b) for the same random seed for training and testing, but taken from different chunks, with using 40 taps and different training dataset sizes per PRBS order as indicated in Table II. The red dashed lines for both panels show the threshold, over which we arrive at the overestimation issue. (Simulation results)

TABLE II
SUMMARY OF PARAMETERS SUCH AS PERIODICITY AND TRAINING/TESTING DATASET SIZE PER PRBS ORDER USED IN OUR STUDY WITH 64-QAM

PRBS Order	Periodicity Bits	Periodicity Symbols	Training Dataset Size	Testing Dataset Size
16	$2^{16}-1$	$\approx 10k$	$5k$	$5k$
18	$2^{18}-1$	$\approx 43k$	$20k$	$20k$
20	$2^{20}-1$	$\approx 174k$	$87k$	$87k$
22	$2^{22}-1$	$\approx 699k$	$262k$	$262k$
24	$2^{24}-1$	$\approx 2.79M$	$1M$	$1M$
26	$2^{26}-1$	$\approx 11.18M$	$1M$	$1M$
28	$2^{28}-1$	$\approx 44.7M$	$1M$	$1M$
30	$2^{30}-1$	$\approx 178.9M$	$1M$	$1M$
32	$2^{32}-1$	$\approx 715M$	$1M$	$1M$
34	$2^{34}-1$	$\approx 2.8B$	$1M$	$1M$

At this point, it is worthwhile to question for which dataset size the NN would learn the data generation rule for the higher-order PRBS. To address this issue, we have conducted a second round of tests, where we trained the NN using up to $1M$ symbols for the PRBS orders ranging from 16 to 34. The main results of this study are reported in Fig. 5(b). Note that to guarantee that the NN learns the PRBS generation rule instead of the symbol periodicity, we generated the training and testing data sets with the same seed, but for training and testing, we selected different data blocks with lengths shorter than the PRBS periodicity. Table II provides the periodicity for each PRBS order, and the training/testing data set sizes for our study. The number of taps was set equal to 40 and the NN equalizer was trained for more than 5000 epochs. As before, if the Q-factor increases above the reference (red line in Fig. 5(b) is the B2B level), this indicates that the NN was able to learn the data generation rule. Fig. 5(b) shows that the Q-factor increases with PRBS order up to 22. This behavior can be explained by the fact that PRBS orders 16 and 18 were trained with 5 k and 20 k symbols, respectively. This training data set is too small to fully train the NN, but we cannot increase it further because of the periodicity constraint given in the third column of Table II.

When the PRBS order is increased, the amount of training data required to learn the respective PRBS generation rule also increases. However, increasing the size of the training data beyond $1M$ becomes impractical (the training takes too long). Thus, the Q-factor curve in Fig. 5(b) starts decreasing for PRBS orders above 22, indicating that the NN's capacity to recover the PRBS generation rule becomes progressively worse. In this case, the $1M$ training data size is insufficient (and/or the NN complexity is insufficient to learn the PRBS generation rule of the highest order). Thus, when increasing the PRBS order from 24 to 30, we observe a decrease in the Q-factor until the point where the NN completely ceases to learn the PRBS generation rule (for the PRBS orders equal to or higher than 32).

We emphasize again that the MTRS should always be used in simulations, instead of the LSFR, because it renders virtually infinite PRBS lengths. Therefore, an undesirable system performance overestimation can be avoided. When dealing with experimental data, even larger PRBS orders (e.g. ≥ 32) should be used with caution, since, depending on the training dataset size, modulation format, and input memory, overestimation can still occur therein.

V. DAC/ADC MEMORY IMPACT ON TRAINING NN EQUALIZERS

As described in the previous section, the two most influential factors related to the data quality are the dataset size and its variability (the absence of spurious periodicity, bias, etc). Fig. 6 shows the typical layout of a lab-style optical transmitter and a coherent receiver. The Tx-DSP writes the signal samples into

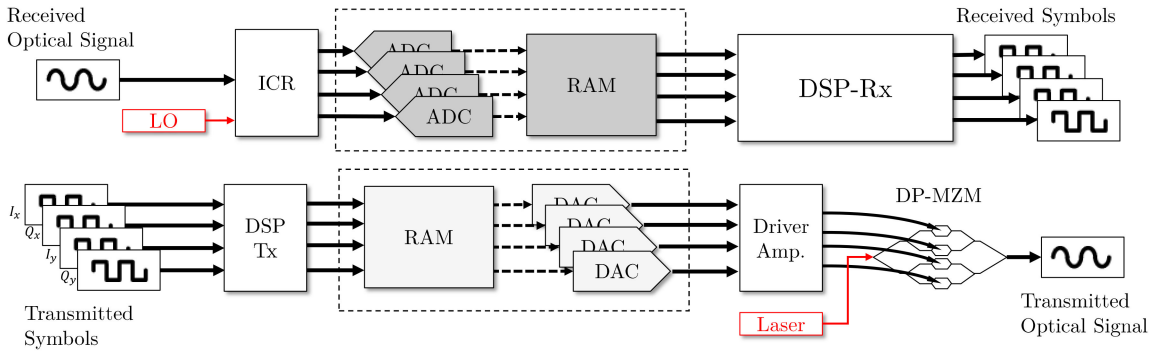


Fig. 6. A simplified block diagram illustrating the components of a typical transmitter and receiver. The DAC/ADC RAM memory is highlighted to emphasise its role in the transmission chain.

the DAC memory, and the signal is transmitted cyclically. On the Rx side, the output of the ADC is written into memory and processed subsequently by the Rx-DSP. When dealing with experimental data, there are several factors to take into account in order to ensure proper data quality for the NN training. First, the DAC and ADC memory is clearly limited in size, which puts an upper limit on the length of the signal to be transmitted. This applies primarily to capturing the buffers built into real-time transponders, where the memory resource is very precious and, hence, the available memory is usually rather limited. However, the same argument applies, in principle, to lab equipment, where, however, the limits are not so stringent. The other factors to consider are the sampling frequency of DAC and ADC devices and the symbol rate of the transmitted signal. For the NN training, we are interested in the sequences of symbols or bits; for a given DAC memory, the number of symbols the DAC can hold depends on the sampling frequency and symbol rate. Finally, an additional constraint can emerge due to the Rx DSP architecture. Often, to achieve synchronization, the DSP implementation assumes that the data are composed of frames of equal length. Furthermore, all frames may be assumed to contain the same payload data to facilitate the alignment of received and reference sequences. In this case, only the data obtained from a single received frame can be taken up for the NN's training and testing. In this section, we will look into how the aforementioned properties impact the training of NN-based equalizers and data variability quality.

Consider an exemplary scenario where we have a DAC/ADC with a memory equal to 512 k samples per channel, operating at the sampling frequency of 80 GSample/s. Assume that the DSP requires about 10 frames holding identical data to achieve a proper synchronization and evaluate BER. Then, the number of samples per frame is around 52 k samples. Let us further assume that our transmission symbol rate is 34.4 GBd. This leads to the number of effective symbols 52 k/(80/34.4) $\approx$ 22 k, which are available for NN training. This can be easily verified by applying the autocorrelation function for the received symbols: Fig. 7 shows the autocorrelation of the experimental data that was produced with a DAC specification close to the ones mentioned above. As it can be seen, the difference in the peaks is around 22 k, which is the same value that we calculated.

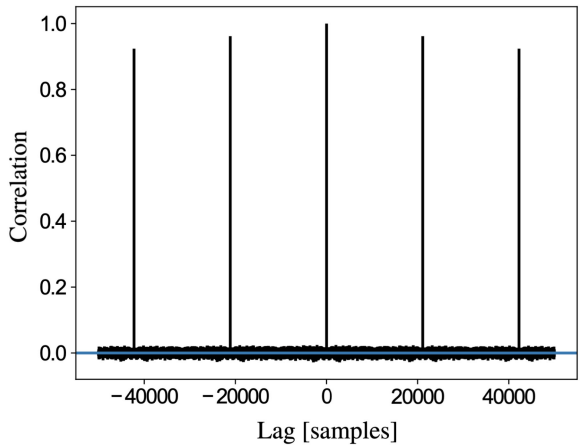
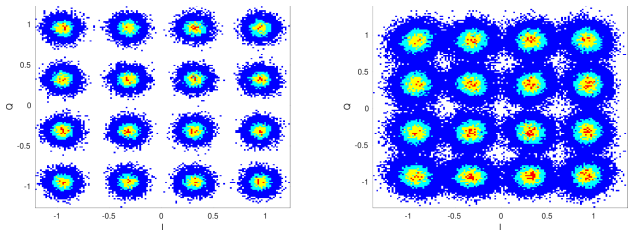


Fig. 7. The auto-correlation diagram of the received experimental signal shows the impact of the DAC memory on the signal's periodicity.

With this information, two main concerns can be raised in terms of the use of machine learning in systems employing DAC/ADC. First, our having a system that is unintentionally biased towards one subset of symbols can result in poor model performance when the latter is validated on a different subset. This fact can ultimately lead to the overfitting of the NN-based equalizer. Second, as shown in Fig. 7, even when we use a PRBS of order 32 or even completely random data, we cannot remove the periodicity in the data since the DAC will repeat just a portion of the PRBS. Because of this, *the same transmission trace cannot be used for training and testing* even if we select non-overlapping chunks. In Fig. 8, we show the constellations after NN equalization using different chunks of the same transmission trace, Fig. 8(a), and when using the transmission traces with different random seeds, Fig. 8(b). From these constellations, it is evident that using the same random seed to train and test the NN provides a constellation of outstanding quality (the respective Q-factor is equal to 13 dB). Nevertheless, the NN trained with the same random seed (the one that produced the picture in Fig. 8(a)) cannot generalize to perform the channel equalization: when we use it with another random seed, we see that the resulting equalized constellation is noticeably degraded, and the NN's true performance in terms of Q-factor is just 8.66 dB, well below



(a) Same random seed (b) Different random seed

Fig. 8. 16-QAM signal constellation after the NN-based equalization when (a) training and testing are carried out using different datasets but generated with the same random seed, and when (b) training and testing are run using datasets generated with different random seeds.

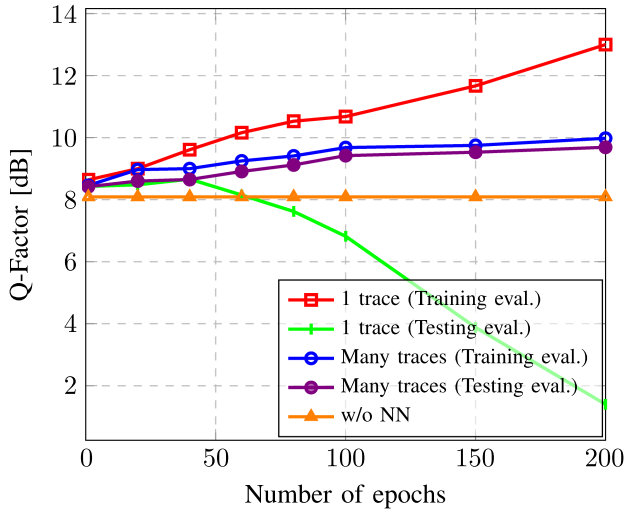


Fig. 9. Q-factor versus training epochs for the experiment with 16-QAM 5×50 km SSMF, 34.4 GBd, 6 dBm power, using the biLSTM equalizer. These curves refer to the training and testing performance of the NN when using one trace or when using multiple traces (our solution is described in the main text) for training.

the previous overestimated value. This fact clearly demonstrates that, in this scenario, the system has overfitted over the training sequence pattern. Therefore, when reporting results using the same random seed for training and testing, you may overestimate the performance of your equalizer.

To avoid overfitting, we have generated our NN inputs with the following procedure. First, we record 60 measurement traces, each with 2^{18} symbols, using a different random seed to generate each trace. After that, we split the entire data set into two parts: 50 traces were taken for training purposes and the remaining 10 traces were saved for testing. From these traces, we then generated the window vector inputs for each symbol to be recovered. Afterward, we concatenated all these window vectors to generate our training and testing datasets. Finally, after having $\approx$ 13M window vectors in the training and $\approx$ 2M window vectors in the testing datasets, we select 2^{20} random input vectors from the overall $13M$ in each epoch while training the NN. To show the impact of not having enough variability in the training dataset, we compared the NN equalizer performance trained using our aforementioned dataset generation solution with the case where we trained the NN with 2^{20} vectors generated by just one random seed. To evaluate the performance of the equalizers, we tested both trained models with 2^{18} input vector taken from $2M$ (the testing set) that were never used in the training.

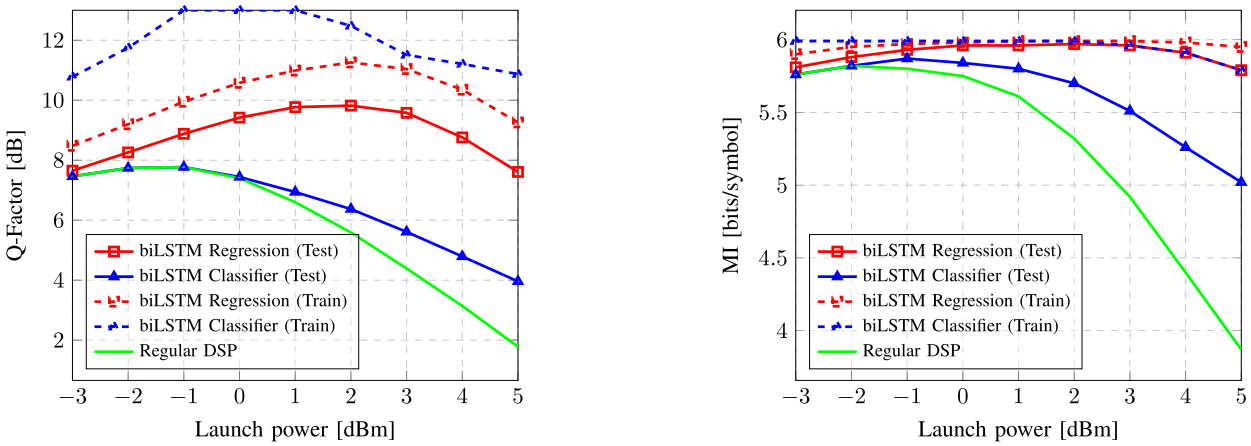
The results of our comparison are summarized in Fig. 9, and several conclusions can be readily drawn from that figure. First, for the case where we trained the NN with only one trace, the traditional overfitting appears: the training curve keeps growing while the testing curve bends down after some point as the model does not generalize. Because the model's variability was only for 22 k points, it overfitted quickly, and the maximum Q-factor of the testing dataset was just 8.66 dB. On the other hand, when we used our multiple trace training solution, we see that both the training and testing datasets' curves grow simultaneously, indicating the generalization capability of the equalizer. Using our method, we were able to reach a maximum Q-factor of 9.69 dB using the testing dataset, which is almost 1 dB higher than the value obtained when training the NN with one single trace. Note that our solution uses the different parts of the training dataset, which benefits not only from the diversity of different symbols picked from different random seeds but also from the fact that noise (which is different for each trace) adds diversity to the dataset as well. Heuristically, we expect this noise to "smear out" each data point, making it difficult for the network to properly match individual data points, and therefore reducing the overfitting [42]. However, as it has been observed in several previous works [43]–[49], the noise injection to various parts of the NN during the back-propagation training can remarkably improve NN's generalization capability, and the latter observation fully complies with the result achieved with our solution, Fig. 9.

VI. Regression or Classification (Soft Demapping) NN Equalizers: The Design Dilemma

First, we mention that the regression versus classification question in designing the most efficient NN equalizer structures[7] was covered exhaustively in [50]. Thus, in this section, we do not expose any specific pitfalls attributed to the types of predictive modeling used in the equalizers, referring an interested reader to the aforementioned Ref., but, instead, we review the drawbacks of using specifically the regression or the multi-class classification in the context of coherent optical channel equalization task.

For regression, the MSE loss function can be treated as a simplification of the true likelihood measurement for the case of optical channel equalization: using the MSE optimization, we assume that the channel noise distribution is additive Gaussian; if the noise distribution is non-Gaussian or signal-dependent, then the MSE-based optimization fails to capture all information in the distorted sequences [51], and may not learn the model parameters that optimally maximize the negative log-likelihood that is the cross-entropy between the empirical distribution defined by the training set and the probability distribution

[7]The receiver NN-based classifiers can be understood as soft-demappers implemented with the NNs.



(a) Q-factor Metric (b) MI Metric

Fig. 10. Performance metrics' comparison for the regression- and classification-based equalizers, showing the impact of overfitting that can be seen when comparing the training and testing learning curves. The analyzed setup considered a 20×50 km SSMF link and a single-carrier-DP 64-QAM signal with 30 GBd and 0.1 RRC pulse shape (simulation results).

defined by the model [52, Chapter 5.5]. In practice, when using MSE-based regression, we try to recover the deterministic part by assuming that the stochastic part has a Gaussian distribution with a signal-independent variance. This approximation is often reasonable, as in many systems the noise-signal interaction is smaller than the transmitter-induced and additive optical amplifier signal-independent noises, and the latter two can often be well approximated as a Gaussian process [53].

For the classification, the cross-entropy loss (CEL) – the most common loss function used in the classification tasks [54] – is the most suitable loss function concerning its meaning in information theory [55], effectively representing any type of noise statistics. However, there are two major drawbacks associated with this loss function, emerging specifically from the machine learning-related perspective, which can make the training of such a classifier a troublesome task. First, regardless of the corresponding inaccuracy in the target space, the CEL penalizes the misclassification between the two classes (i.e. between any two constellation points in our problem) with the same "cost" value: the penalization ignores the spatial proximity of the labels, reflecting only the fact that the constellation point has been misclassified. However, typically, the account of the misclassification "type," i.e. when the cost of all errors is not equal, can be (and typically is) quite beneficial for the efficient NN's training. The cost of making a mistake can be determined by the projected and actual classes of an example [56], [57]. Each class represents a distinct notion that can be identified using a NN in the conventional classification task, for example, when we classify different kinds of coordinate objects. However, when it comes to the problem of optical equalization, each class contains more information than just a label. In other words, the different classes correspond to different point positions in the constellation, and each class has its nonlinear distortion level. The additional information about each label can be obtained by using the physical nature of each nonlinear level, i.e., the aforementioned distortion level that depends on the constellation point's power.

To better understand this question, consider the task of classifying symbols in, say, a 16-QAM constellation. In this problem, the misclassification between classes that share the same decision boundary should cost less than the misclassification of the ones that do not share any decision boundary: in the first case, one naturally shares some symbols due to the noise-induced clouds' spreading and overlap, while the second case is a "serious error". However, using the CEL, we will only capture errors on the target class: it discards any notion of errors that you might consider "false positive" and does not care how predicted probabilities are distributed other than the predicted probability of the true class (since we deal with one-hot encoded vectors), implying that only the predicted probability associated with the label influences the value of the CEL. Thus, from the standpoint of machine learning, we can say that there is a natural ordering among the labels of the target variable. The training difficulties that the standard CEL can bring in classification with natural ordering problems are discussed in more detail in [58].

The second classification disadvantage is the magnitude of the gradients that arise during the training process with the CEL. The CEL surfaces, according to [59], present fewer local minima than the square error-based losses (SEL), the type to which the MSE loss belongs. The CEL, on the other hand, has stronger gradients than the SEL, which leads to a stronger tendency of overfitting in CEL-trained systems, leading to the SEL's having a better generalization property in almost all scenarios examined in [59]. This result was explained thereby assuming that the CEL loss surface is more prone to sharp minima (narrow valleys) than the SEL's surface, making the overfitting considerably "easier" to happen for the former systems. Additionally, it was also shown that such classification-based systems can suffer from the gradient vanishing problem. For instance, [60] shows the gradient vanishing when using the softmax with categorical-CEL, and the same for the sigmoid with the binary-CEL was reported in [61]. We observed the same tendency in our coherent channel equalization problem, when the MSE-based system generalized better than the categorical CEL systems, due to overfitting, sharp

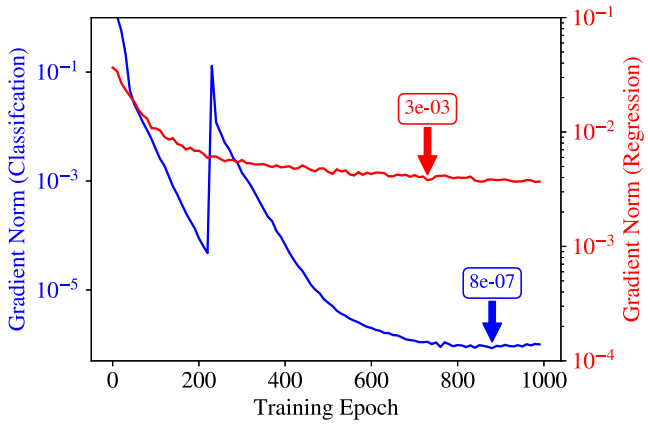


Fig. 11. Gradient norm over the training epochs for the Regression model (red) and Classifier model (blue), to highlight the gradient vanishing problem and sharp loss-landscape local minima.

local minimal, and ultimately the gradient vanishing problem that is observed in the CEL-based learning.

Finally, we can point out one more disadvantage of classification-based equalizers, which is more relevant to their use in real transmission systems. When utilizing the classification, the equalization must have a predefined number of outputs that correspond to the constellation's cardinality. This means that the classifier model's operation is dependent on the modulation format on which it was trained. In other words, a classifier's practical implementation (e.g., in hardware) would limit the device's applicability to a particular modulation format only, which greatly reduces the device's flexibility and adaptability. However, as shown in Refs. [15], when we use regression, we can easily adapt the model to work with different modulation formats. As a result, regression-based models are much more flexible (reconfigurable) than classification models, allowing us to use the former under conditions different from the ones in which the regression models were trained.

Now, since we have covered the key points related to the regression and classification tasks, we will look at how an equalizer (or a soft-demapper) with the same architecture (applied to the data from the same transmission setup) performs, and how the result depends on whether the regression or classification task is employed. In this test, a single channel DP signal of 30 GBd is transmitted over 20×50 km employing 64-QAM (used in Figs. 10 and 11). The optical launch power is varied from −3 dBm to 5 dBm. We used a biLSTM equalizer with 208 hidden units and 66 neighboring symbols for the input (the memory). Note that both regression and classification architectures have the same number of layers, inputs, hidden units, and they were trained on the same data sequences. However, for fairness purposes, we have optimized the mini-batch size and learning rate for each equalizer individually, because we observed that using the same ones in regression and classification was causing even stronger overfitting after a few epochs for the latter, which is one of the classification drawbacks.

Fig. 10 shows the Q-factor and MI dependencies for training and test datasets when we use the same biLSTM equalizer for regression and classification (of course, the output layers of NN structures are different for the two tasks). The Q-factor curves for the training and testing of the classifiers exhibit a considerable discrepancy, as shown in Fig. 10(a), indicating that the classification model is overfitted. This noticeable difference in the classification task persisted even after we had optimized the learning rate and mini-batch size. However, in the case of regression, the training and testing output curves behave almost identically, in contrast to the classifier's results. It follows that for our test scenarios, the regression model based on the MSE generalizes considerably better, which is consistent with the findings from [59], where the strong gradients in the classification task loss function impact learning further: the learning state gets "trapped" in a sharp local minimum of loss landscape. Furthermore, we point out that after the equalization, the regression equalizers resulted in a higher Q-factor than the achieved value for the classifiers, owing to the better generalization of the regression-based NN structures on the optical datasets. The maximum regression-based NN's Q-factor on the testing dataset was 9.82 dB at 2 dBm, while the maximum Q-factor for the classification was 7.74 dB at −1 dBm. Besides the Q-factor, it is also instructive to show the MI for both tasks as well. The reason for this is that the classification loss function, the CEL, is directly related to the MI metric, therefore, displaying Q-factor after a hard decision can limit the entire potential of the model [15]. Looking at the MI values in Fig. 10(b), we can see that the classifier's training performance was overfitted, yielding nearly the maximum MI attainable for each power when the training dataset was used. However, in the case of regression, the training and testing curves followed the same trend, indicating a much better generalization capability of the equalizers. The maximum MI on the testing dataset of the regression NN was 5.97 bits/symbol (note that this value is only a MI's lower bound), and the one for the classification was 5.87 bits/symbol (but this MI value is exact). We can then conclude that in typical optical transmission tasks, the machine learning drawbacks associated with conventional classification "overpower" the regression-related shortcomings related to statistical limitations.

Aside from the overfitting, we also want to highlight the consequence of the second classification drawback, regarding the gradient vanishing in the training process. According to [52, Chapter 8.2.2], a practical way to demonstrate that the local minima is potentially the cause of the learning problem is to verify that the gradient norm shrinks to some "insignificant" values along with the training. As in [50], we depicted in Fig. 11 the gradient norm of the last layer for the biLSTM equalizer over the epochs to show the effects of the loss function on the behavior of the gradient norm. Unlike [50] that presented this plot for the MLP architecture, we now show it for the biLSTM equalizer, basically observing the same trend as reported for the MLP equalizer in a completely different transmission setup. From Fig. 11, it is clear that after just 200 epochs, the gradient norm for the categorical-CEL case drops from 1.25 to 7×10^{-5} and, with continued training, goes even lower to 8×10^{-7}. For the regression case using MSE, the gradient continues to stably decrease over the epochs, reaching a minimum of around 0.003. Therefore, we can affirm that we potentially have sharp local

TABLE III
SUMMARY OF BILSTM AND MLP PERFORMANCE DEPENDENCE ON THE MINI-BATCH SIZE FOR THE DIFFERENT CONSTELLATION CARDINALITIES. SIMULATION RESULTS

QAM	Q-factor Ref	MLP Equalizer Q-factor (Different Batch Sizes)						biLSTM Equalizer Q-factor (Different Batch Sizes)					
		8	16	32	64	128	2048	8	16	32	64	128	2048
8	8.65 dB	8.62 dB	8.70 dB	8.65 dB	8.74 dB	8.82 dB	9.44 dB	10.08 dB	10.25 dB	10.51 dB	10.71 dB	11.28 dB	13 dB
16	7.22 dB	6.91 dB	7.02 dB	7.09 dB	7.17 dB	7.28 dB	7.72 dB	8.58 dB	9.14 dB	9.72 dB	10.10 dB	10.43 dB	11.57 dB
32	5.05 dB	4.47 dB	4.57 dB	4.69 dB	4.80 dB	5.01 dB	5.58 dB	7.02 dB	7.66 dB	8.36 dB	8.69 dB	8.75 dB	8.79 dB
64	3.15 dB	2.67 dB	2.73 dB	2.95 dB	2.96 dB	3.06 dB	3.79 dB	5.78 dB	5.87 dB	6.29 dB	6.47 dB	6.51 dB	6.78 dB
128	1.53 dB	0.77 dB	0.90 dB	1.10 dB	1.30 dB	1.36 dB	2.09 dB	3.54 dB	3.92 dB	4.09 dB	4.13 dB	4.14 dB	4.27 dB

minima in categorical CEL for the optical channel equalization task, since, as described in [52, Chapter 8.2.2], the gradient norm shrinks to a negligible value (on the order of 10^{-7}) during training.

Finally, we conclude this section with a brief discussion of the loss function study in the task of optical channel equalization/soft demapping. Even though the MSE is a good fit when we deal with a Gaussian noise model making the NN learn effectively and deliver attractive Q-actor gains, in some scenarios, the stochastic gradient descent (SGD) algorithm continues to decrease the MSE, but this does not translate into the BER improvement, which means that we fall into a local minimum. This is visually observed when the "jail window" constellation pattern appears, which indicates a mismatch between the MSE loss and the BER metric. We believe that the "jail window" appears for local minima in learning, and although it can produce seemingly good BER results, it is not the ultimate equalization. In the next section, we present a scenario where we clearly show that when the "jail window" disappears due to optimization of the mini-batch size, the final Q-factor performance increases for the model without the "jail window".[8] To match the QoT metric of our transmission and the loss function of our learning algorithm, we can claim that the CEL is the most suitable loss function for communication applications [55] since by minimizing cross-entropy, we maximize the mutual information of the system, and therefore no mismatch between the loss function and the QoT metric appears. However, as presented in this section, we observed other machine learning-related problems attributed to the learning process, namely SGD learning. We note that the landscape of such a loss function has very sharp local minima, and the gradients tend to vanish in the early learning stage due to our high accuracy requirements in optical channel equalization tasks. Therefore, even though the CEL does not have a statistical limitation for the noise likelihood, it poses many learning difficulties that can ultimately make its performance worse than that of the regression NNs based on MSE. Here, we also stress that several works have also acknowledged that both regression and classification have disadvantages [62]–[68].

So we can confidently say that, for optical communication, we still do not have the ultimate answer to the question of what would be the best loss function, as each candidate has drawbacks. We believe that by looking at different fields we can potentially find some other possible candidates, but this required further investigation. In computer vision, it was discovered that when an image is processed, the majority of the pixies describe the image background, and only a few pixels express the objects in the image. This resulted in inefficient training because most parts of the image correspond to "an easy prediction" (which means that they can be easily labeled as background by the detector) and therefore offer little relevant learning. Although individually they provide tiny contributions to the loss value, when we combine those contributions, they can overwhelm the loss and computed gradients, resulting in a degraded model's prediction performance. It happens because easy predictions (detections with high probabilities or, in our context, the correct classifications following a simple hard decision) account for a large share of inputs. To address this issue, in [68] Facebook A.I. developed a new modified approach named focal loss (FL), by adding a weighting factor to the CEL function. The FL gives a higher weight to cases that are hardly misclassified: in communications, it would correspond to the cases that has been misclassified after the HD process. We believe that the difficulty of the "dataset imbalance" (meaning that just a small fraction of the dataset corresponds to the wrong HD predictions) exists in virtually all high-accuracy communication-related equalization/demapping problems. As a toy example, consider a system where an initial SER after HD is equal to 10^{-3}. Training the NN-classifier with 100 K symbols, in this case, means that only 100 symbols (0.1%) are the errors that persist after HD and that the model needs to learn from them to mitigate the impairments, while the rest 99900 symbols (99.9%) of the training dataset corresponds to "an easy prediction". Therefore, to improve the performance of classifiers, we expect that a similar focal loss function, as in computer vision, must be created for the communications application. This can be an interesting topic for future research in the field of optical channel equalization and efficient NN-based soft symbol demapping.

VII. INTERRELATION BETWEEN BATCH SIZE, CONSTELLATION CARDINALITY, AND EQUALIZATION QUALITY

The NN training presupposes the use of optimization algorithms, such as SGD or Adam [69], to update the NN parameters (weights) based on the value of the loss function. Traditionally, two training methods are utilized: batch training, in which the algorithm updates the weights after the entire training dataset, and online learning, which is executed after each training sample.

In practice, however, stochastic optimization methods use mini-batches, i.e., the portions of all training examples with the

[8]Note that this is not always the case, increasing the mini-batch size will not necessarily eliminate the "jail window" pattern.

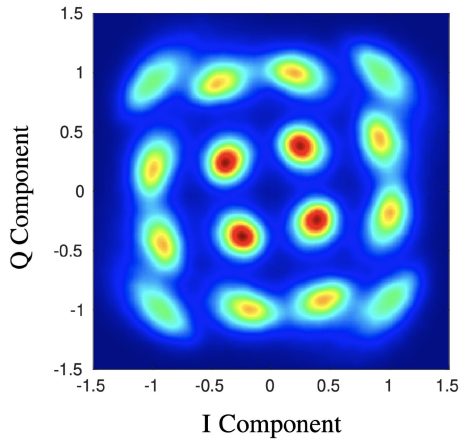


Fig. 12. Signal constellation of 16-QAM after metro-link (450 km) simulated optical transmission using a Nonzero Dispersion Fiber [73]. We used this highly-nonlinear exemplary system to visually demonstrate the distinct distortion levels at different constellation points: the constellation points at the outer layer are clearly more distorted than the ones closer to the origin.

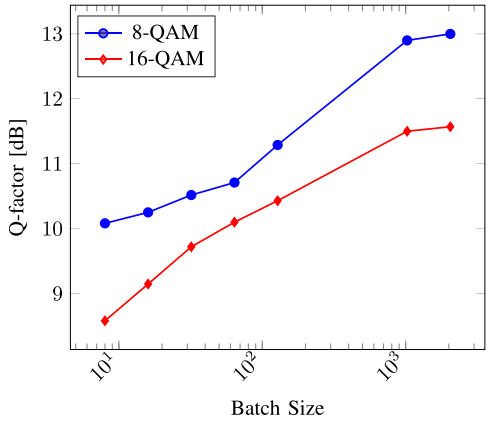


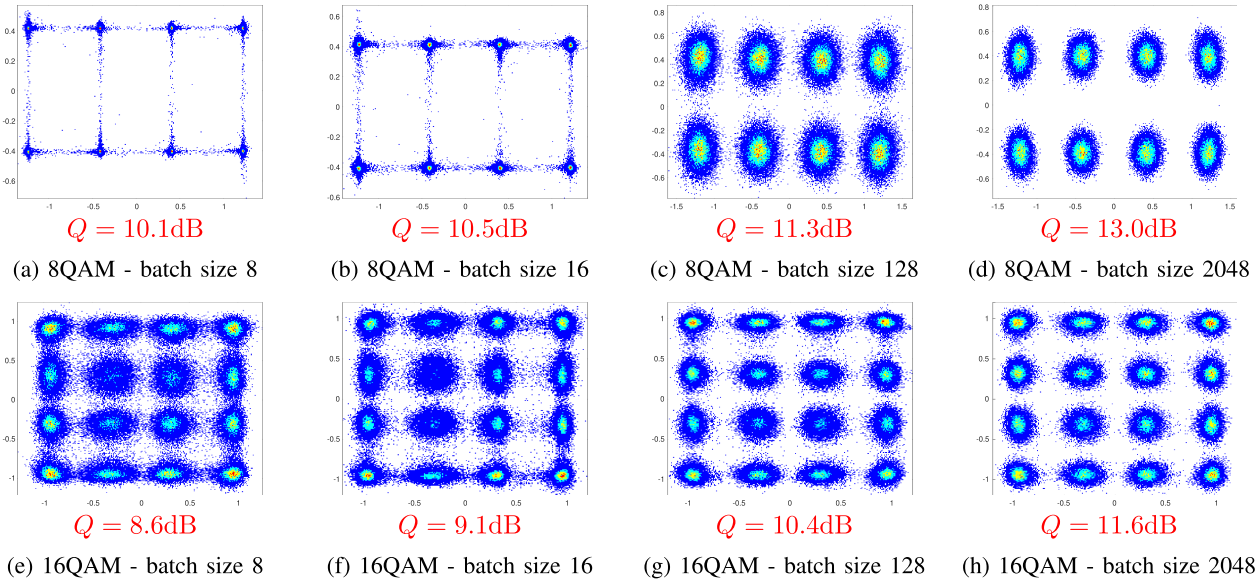
Fig. 13. Q-factor of 8- and 16-QAM simulated signals equalized by the biLSTM trained with different mini-batch sizes: 8, 16, 32, 64, 128, 1024, and 2048. The simulated transmission setup is described in the main text.

size greater than one, but less than the entire training dataset, to compute the gradients using less memory and update the parameters after each mini-batch [52]. One reason for this is that modern NNs demand such a large amount of data that the computer dynamic memory cannot keep the entire dataset. Notably, the size of a mini-batch is regarded as one of the most important hyperparameters to consider when training an NN. Large mini-batches are well known in deep learning for their ability to significantly speed up the training process, while also providing an accurate approximation of the gradient [52]. Small mini-batches, on the other hand, have been shown to have a regularizing impact, preventing overfitting [52], [70]. The latter phenomenon can be explained by the fact that, while the large mini-batches can indicate the gradient's direction, the algorithm cannot forecast how long this trend would continue, usually causing the training process to take a large update step forward. This leads to unstable learning or falling into local minima. However, small mini-batches, while carrying added noise due to their smaller sampling across the entire dataset, result in small steps and the system can easily converge to a true direction [70]. Several studies have been carried out to investigate the effects of mini-batches size on the NNs performing traditional deep learning tasks. In [71] proposed using small mini-batches to introduce noise in the gradient estimation and push the gradient away from sharp local minima; the authors of that Ref. also demonstrated that the optimal batch size for the CIFAR-10 dataset is 80. In [72], it was empirically demonstrated that the large mini-batch sizes lead to the convergence at sharp local minima, resulting in a poor NN's generalization, whereas the smaller mini-batches lead to flatter local minima, allowing the NN to achieve a better generalization. However, to the best of our knowledge, no research has been conducted to study the effect of mini-batch size on the performance of NN-based regression equalizers in optical transmission.

The motivation for this study is that the mini-batch in the NN performing optical channel equalization has more physical meaning than it does in other "traditional" machine learning tasks, such as computer vision. Recall that different transmitted symbols in the constellation correspond to different optical field intensities, depending on the modulation format. This can be seen by looking at the 16-QAM received constellation diagram, Fig. 12, where the outermost points are the most distorted ones. This happens because the fiber nonlinearity is proportional to the cube of the optical field amplitude, so the most distant points (having the largest amplitude) experience the most nonlinear effects. Because the NN's parameters are updated after each mini-batch, the training process can be more efficient if each mini-batch contains training samples that cover the whole range of possible constellation amplitudes, to encompass different distortion levels. If this hypothesis is correct, there should be a relationship between the mini-batch size and the modulation format's cardinality, also affecting the NN's performance. This section reports tests and simulations to explain the aforementioned connections.

Table III shows the Q-factor of optical signals equalized by the MLP and biLSTM MSE-regression equalizers for a range of modulation formats: 8-, 16-, 32-, 64, 128-QAM[9] and mini-batch sizes: size 8, 16, 32, 64, 128, and 2048. In this section, simulations were performed for the SSMF 16×60 km link at 5 dBm launch power and 34.4 GBd symbol rate. When the size of the mini-batch increases, we observe a progressive growth in the Q-factor after equalization. Let us consider the case of biLSTM, which gives the best equalization quality. When the mini-batch size is increased for the 8- and 16-QAM, the Q-factor after the biLSTM-based equalization dramatically improves. For example, comparing the post-equalization Q-factors in the 16-QAM scenario, for mini-batch sizes 8 and 2048, the 3 dB improvement can be seen for the largest mini-batch over the smallest one. This behavior of mini-batch vs. Q-factor gain for biLSTM is illustrated in Fig. 13 for 8-QAM and 16-QAM. Note that increasing the mini-batch size above 2048 does not provide any further improvement: the Q-factor reaches some kind of

[9]Note that to generate the 8-QAM constellation we used the standard Matlab function `qammod`; the same constellation shape was used in [74]–[77].



(a) 8QAM - batch size 8 (b) 8QAM - batch size 16 (c) 8QAM - batch size 128 (d) 8QAM - batch size 2048

(e) 16QAM - batch size 8 (f) 16QAM - batch size 16 (g) 16QAM - batch size 128 (h) 16QAM - batch size 2048

Fig. 14. Signal constellations (simulation results) for different batch sizes after equalization using the biLSTM equalizer for 8- and 16-QAM. The transmission system is described in the text. The Q-factor of each constellation is also shown to highlight the improvement in QoT.

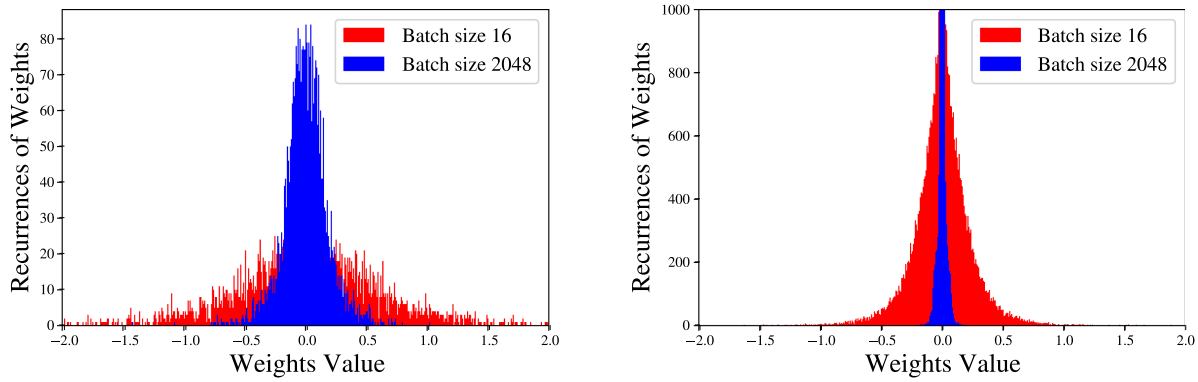
plateau (saturates) after the batch size of approximately 1000. When using a higher constellation cardinality, 128-QAM, the improvement rendered by the biLSTM equalizer increases from 3.54 to 4.27 dB when changing from the mini-batch size from 8 to 2048. We note that, from Table III, for the 128-QAM with the MLP equalization, the difference between the Q-factors corresponding to the lowest and highest mini-batch sizes is approximately 1.3 dB, which is bigger than the difference for the biLSTM. The increase in Q-factor backs up the claim that the NNs can learn more efficiently when given the training samples covering all nonlinearity levels in the constellation.

Together with the improvement in Q-factor (up to some limiting value) following the growth of the mini-batch size, we can see the interrelation between the constellation distribution after equalization and the mini-batch size used in training. The constellations for the 8- and 16-QAM systems for different mini-batch sizes utilizing the biLSTM equalization, are given in Fig. 14. In this figure, we can observe that utilizing a small batch size for 8-QAM drives the NN-based equalizer to fall into the previously discussed "jail window" pattern rather than into the Gaussian-type distribution, but the latter can be obtained when we use larger mini-batch sizes. In the 16-QAM case, the signal constellations after NNs with small mini-batches became noticeably distorted, the "jail window" elements are clearly seen. At the same time, for the larger mini-batches, the 16-QAM constellations show little distortion with clear concentrations at the center of each constellation point.

The degradation of constellations from circular clusters into the "jail window" pattern when training with small mini-batch sizes, is actually a new and intriguing phenomenon: to our knowledge, there have been no studies relating the equalizer's performance deterioration to the size of the training mini-batches. Generally, as we have already seen, several factors can contribute to the patterning effect of the "jail window". In this section, we addressed the specific situation, where the "jail window" occurred solely due to the batch size effect, and so we were able to understand how and why the batch size relates to the "jail window" patterning. In a more general case, different contributions amalgamate, resulting in the "jail window" pattern; the latter is always an indication that something is not good with the training or with the NN system itself. This is actually evident from the results in Fig. 14(a) to (d). In case (a), where the learned weights resulted in the "jail window," the Q-factor is around 10 dB, while in case (d), where the learned weights did not generate the "jail window," we measured the Q-factor to be around 13 dB. Therefore, we expect that when the "jail window" pattern is present, even though the performance metric value can be rated as "satisfactory," in the training we have most likely reached just a local minimum, and some better equalization results can be achieved with more appropriate training or by using a modification of the NN architecture.

To explain further the degradation of signal constellations further, when training the biLSTM equalizer with small mini-batch sizes, we investigated the weight distribution inside the NNs. More precisely, we compared the weight distributions in the last linear layer (output layer) of the biLSTM, and the distribution in the forward LSTM layer, for two values of the training mini-batch size: 16 and 2048, using an 8-QAM system as an example. Fig. 15 shows that the weights, when training with small mini-batches, range over a considerably larger interval than we have when training the NN with larger mini-batches. From Fig. 15(b), we can see that the final layer weights were significantly saturated when the NNs were trained with small mini-batches[10]. The saturation is indicated by the presence of large value weights and typically degrades the performance

[10]Well trained NNs usually have the weights' distribution being close to the Gaussian distribution with a small variance. When this variance is too large, say more than one, as we show in Fig. 15, it usually indicates a saturation of weights, which, in the NN terms, means our disregarding some important input features.



(a) Weight distribution at the forward LSTM layer

(b) Weight distribution at the final linear layer

Fig. 15. Weight distributions of the biLSTM equalizer's forward and output layers, corresponding to the mini-batch size of 16 (red) and 2048 (blue).

of the NN. At this point, we hypothesize that the presence of large weights makes the output of the NN strongly dominated by a few weights with a large value. This is a widely known phenomenon in deep learning when training NNs to perform regression tasks [52], [78]. The NNs featuring large weights tend to have the regression output converging to a few values that are hard-coded from the input. In the case of optical signal equalization, this effect deteriorates the signal constellations into a "jail window".

To carry out the analysis further, let us examine the number of outlier weights in the linear layer. As it can be seen in Fig. 15, the weights greater than one can be considered outliers. We found that in the case of 8-QAM, the real parts of the recovered symbols have twice as many outlier (larger than 1) weights as the imaginary parts (298 for the latter case in comparison with 153 for the former)[11] When we increased the outlier weight threshold value criteria from 1 to 4, we found just four weights on the neuron representing the real value of the constellation, while the neuron for the imaginary part had just two. This is the exact ratio between the number of continuous straight "lines" (each "line" is made up of equalized constellation points) in the amplitudes of the in-phase and quadrature, as seen in Fig. 14(a).

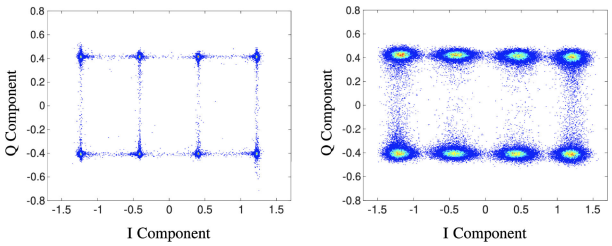
We propose the following explanation for the aforementioned patterning effects. When employing small mini-batch sizes with a small cardinality constellation, there exists a noticeable probability that many points in the batch belong to just a single amplitude level. As a result, the NN tends to learn the inverse channel function at this particular nonlinearity level and is more likely to hard-code some inputs to this obvious amplitude output. This issue is less likely to occur with high-cardinality modulation formats since it is less probable that the batch would contain a lot of samples belonging to the same amplitude level. However, even in high modulation formats, we could see the "jail window" pattern when using the MLP equalizer, but now with less intensity[12]. One possible explanation for why the optical signals recovered by the MLP equalizer are more prone to constellation degeneration than in the case of the biLSTM equalizer, is that the feed-forward structure (e.g. the MLP) makes it easier to hard-code input data to the output. At the same time, the recurrent-type structures (e.g. the LSTM type) allow parameters to be shared across the NN model [52], making the hard-coding effect more difficult to emerge. Here, however, we want to remark on the performance of the MLP and LSTM equalizers. Both Sections III and VII demonstrated that the LSTM equalization produces a better result than the MLP equalizer. The LSTM, which can be deemed as a nonlinear infinite impulse response (IIR) filter, can represent the inverse of a nonlinear channel more accurately than equalizers that use an MLP structure; the latter is actually a non-linear finite impulse response (FIR) filter [79]. IIR filters, unlike FIR filters, do not require that the system has finite memory, giving the LSTM-type equalization an advantage. Theoretically, the MLP may also achieve such a high degree of performance when the input accounts for large enough memory; but, due to the overfitting of the MLP structure[13], this turns out to be extremely difficult to achieve. On the other hand, because the large-memory handling LSTM structures are simpler than MLPs, the overfitting there does not occur, or at least, is much less pronounced.

After considering the deteriorating effects attributed to the presence of large-value weights, we advocate the use of a well-known regularization technique, L^2 regularization, as a feasible way of minimizing the degradation in the equalized signal constellations. The idea behind L^2 regularization is to penalize large weights and favor smaller weights throughout the model [78]. From a Bayesian statistics standpoint, the addition of L^2 regularization is equivalent to performing a maximum a posteriori estimation with a Gaussian prior; the traditional MSE loss function corresponds to a maximum likelihood estimation when the likelihood has a Gaussian distribution. This means that after the learning, our equalization transfer function (likelihood) has been re-weighted to reflect a prior, making it much more difficult to have hard-coded correlations between the input and

[11]This effect persists in the case of symmetric 16-QAM and other constellations when the "jail window" occurs, such that it is not relevant to the asymmetry of the 8-QAM constellations used in our study.

[12]We observed that the "jail window" pattern is modulation format dependent. An interesting discussion on the fact that non-linear MSE equalizers realize a stair function can be found in [55]

[13]To recover the nonlinear channel, the MLP would need a complicated structure with large memory. However, such a redundant architecture is equally susceptible to learning the training dataset, which leads to overfitting [14].



(a) Without regularisation (b) With regularization

Fig. 16. 8-QAM signal constellations after the NN-based simulated channel equalization when training with mini-batch size equal to 16 without (a) and with (b) L^2 regularization.

output (recall that the hard-coding is one of the reasons causing the "jail window" pattern). The following equation represents the L^2 contribution to the regularized loss function $L(y, \hat{y})$:

$$L(y, \hat{y}) = \text{MSE}(y, \hat{y}) + \lambda \sum_{i=1}^{n} w_i^2, \tag{6}$$

where the first addend is the initial MSE loss function, λ is the regularization parameter that must be optimized,[14] n is the total number of weights in the NN topology, and w_i is the i-th weight in the NN topology.

Fig. 16 shows the signal constellations of the 8-QAM signal equalized by the NNs trained with the mini-batch size of 16, with and without regularization. We observe that when using the regularization, the degradation of signal constellations can be prevented. It is worth pointing out that although regularization can render better constellations, the equalizing performance is still badly affected by the small mini-batch sizes: even with regularization, we cannot match the performance of the NNs trained with large mini-batches. Thus, while the large weights contribute to the degeneration of constellations, this is not the only detrimental effect caused by a small mini-batch size. This observation suggests that the most important problem with small mini-batch sizes can relate to the insufficient number of unique amplitude levels that are present in each mini-batch, as we suggested above.

Fig. 16(a) and 16(b) demonstrate the signal constellations at the epoch with the highest Q-factor. It was observed that when training the NNs after this level, the performance degeneration took place even with the regularization. Although most of the weights after the regularization turned out to have a relatively low value, there still exist significantly larger outliers. We surmise that these outliers contribute to the distortion of the signal's constellations. Interestingly, using large mini-batch results in a narrower distribution of the weights, which does not feature any significant outliers even when the regularization is not applied. This observation strongly supports our claim that training with an insufficiently large mini-batch brings about the performance degradation of NN-based coherent optical channel equalizers.

[14]We have tested the regularization parameter from the range between 0.01 and 0.0001, but no drastic difference was observed in terms of Q-factor improvement. In the plots presented in this section we used the parameter value $\lambda = 0.001$.

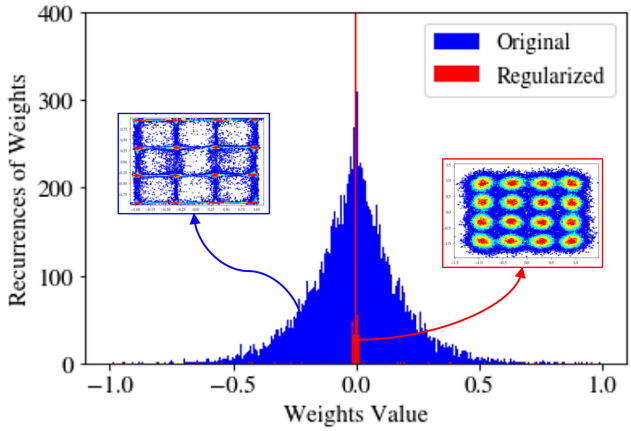


Fig. 17. Distribution of MLP weights for the entire model: non-regularized (Original) and regularized cases. The insets show the respective equalized constellations for each case.

Finally, we applied the L^2 regularization to all hidden layers of the MLP equalizer in another transmission case where the "jail window" was really strong: it was the case (ii) described in Section III and given in Fig. 3(b). The weight distribution of the entire model for the MSE (marked as "Original") and for the MSE with the L^2 regularization (marked as "Regularized") is presented in Fig. 17. From this result we can see that, this time, for a higher modulation format (16-QAM), the regularization makes the original "jail window" constellation be again a set of "Gaussian-like" clusters, see the corresponding insets in Fig. 17. However, even though the regularization produced a visually different constellation output, the original and regularized model revealed almost identical performance in terms of the eventual Q-factor: the original one gave 7.89 dB, regularized −7.61 dB. This fact shows that regularization is not the solution to make the model learn further[15], but it rather helped us to identify which element in the NN-model was responsible for the "jail window" pattern.

VIII. Computational Complexity Analysis: Number of Parameters Versus Number of Multiplications

One of the most critical elements in implementing a deep NN in an end-product is the signal processing latency associated with the signal passing through the NN. The majority of real-world applications demand a really short inference (processing) time [81], [82], ranging from a few milliseconds to a second. However, evaluating a NN's inference time, or latency, accurately and efficiently, can be a difficult problem [83]. The consequences of latency evaluation errors can lead to poor decisions about the implementation of NN. At the same time, the device's power consumption and the required hardware size for the NN implementation are no less important characteristics; the latter is also related to the NN's CC. Additionally, we note

[15]Using the L^2 regularizer can also make the NN model be a single point attractor in the space of its weights, where the attractor is located at the origin. In such a case, any information that was inserted into the NN model dies out exponentially fast. As described in [80], this can hide long-term memory traces that impact the learning process in our task.

TABLE IV
SUMMARY OF LATENCY VALUES FOR DIFFERENT HYPERPARAMETERS SETS

Topology	MLP Equalizer				biLSTM Equalizer			
	Hyperparameters	NN Parameters	RMpS	Latency (s)	Hyperparameters	NN Parameters	RMpS	Latency (s)
1	$N = 35$; $n_{1/2} = 600/518$	482,236	4.82e+5	7.87e-5	$N = 5$; $n_h = 73$	48,180	5.03e+5	1.03e-4
2	$N = 35$; $n_{1/2/3} = 400/460/400$	482,400	4.82e+5	7.90e-5	$N = 10$; $n_h = 52$	27,664	5.00e+5	1.30e-4
3	$N = 35$; $n_{1/2/3/4} = 200/156/553/555$	482,293	4.82e+5	7.06e-5	$N = 14$; $n_h = 44$	22,000	5.03e+5	1.46e-4
4	$N = 45$; $n_{1/2} = 1250/458$	1,028,416	1.02e+6	1.33e-4	$N = 25$; $n_h = 48$	29,760	1.04e+6	2.82e-4
5	$N = 45$; $n_{1/2/3} = 620/564/800$	1,028,160	1.02e+6	1.47e-4	$N = 30$; $n_h = 43$	26,660	1.01e+6	2.51e-4
6	$N = 45$; $n_{1/2/3/4} = 610/500/500/501$	1,028,542	1.02e+6	1.21e-4	$N = 35$; $n_h = 40$	25,440	1.03e+6	3.14e-4
7	$N = 60$; $n_{1/2/3} = 1000/1000/600$	2,085,200	2.08e+6	2.51e-4	$N = 40$; $n_h = 53$	41,340	2.00e+6	5.34e-4
8	$N = 50$; $n_{1/2/3/4} = 2000/500/910/200$	4,087,000	4.08e+6	4.30e-4	$N = 50$; $n_h = 68$	66,640	4.02e+6	9.30e-4

that the latency is actually architecture dependent. For instance, we can implement an MLP in a fully parallel way, which would provide the lowest latency but the highest power consumption. Alternatively, we can implement a single multiplier unit and process the multiplications sequentially. This would result in the lowest power consumption and the highest latency, such that we always ought to seek the desirable balance between the power consumption and latency. In this paper, to better understand the relevance of latency and our CC metrics, we measure the average latency to recover one symbol using a Colab CPU (4vCPU at 2.0 GHz with 26 GB RAM) for sequential NN architectures.[16] In this section, we will analyze the number of real multiplications per recovered symbol (RMpS) as a CC metric. Technically, if well-defined, this CC metric can help in the assessment of a solution's design appropriateness before going to the hardware level. In the following, we will identify three aspects that should be considered when evaluating the CC of any NN-based equalizer.

First, we address two important points. i) We show that the RMpS metric provides a good estimate of the CC by demonstrating its proportionality to the averaged latency (inference time) for one equalized symbol in a pure sequential architecture. ii) We show that the number of weights of the NN-based equalizer is not a good metric for assessing the NN's complexity. To showcase these points, we have tested eight different topologies for MLP and biLSTM equalizers with different levels of RMpS and measured their one-symbol latency algebraically averaged over 1 M recovered symbols. The results of the tests performed in this subsection are summarized in Table IV. To account for the number of parameters, we used the TensorFlow application for each topology, and for the proper RMpS computation, we used the equations introduced in [14]. The expressions for the RMpS of the MLP having 2, 3, and 4 layers, and biLSTM equalizers are, respectively:

$$C_{\text{MLP}_2} = n_s n_i n_1 + n_1 n_2 + n_2 n_o, \tag{7}$$

$$C_{\text{MLP}_3} = n_s n_i n_1 + n_1 n_2 + n_2 n_3 + n_3 n_o, \tag{8}$$

$$C_{\text{MLP}_4} = n_s n_i n_1 + n_1 n_2 + n_2 n_3 + n_3 n_4 + n_4 n_o, \tag{9}$$

$$C_{\text{biLSTM}} = 2 n_s n_h (4 n_i + 4 n_h + 3 + n_o), \tag{10}$$

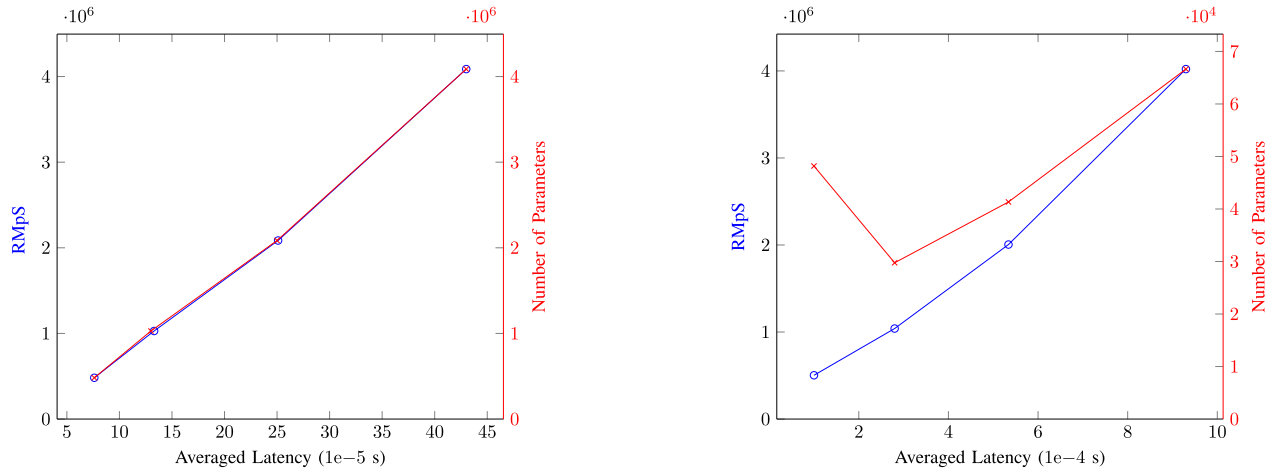
[16]Note that with a different architecture, e.g. on GPU, TPU, different CPU, FPGA, or ASIC, the latency can be drastically different [84].

where n_s is the input time sequence size, with the memory size $n_s = 2N + 1$, with N being the number of neighboring symbols considered, and n_i being the number of input features, which, in our case, is equal to 4 (the number of outputs per symbol); n_o is equal to 2, $n_{1,2,3,4}$ are the number of neurons in each respective hidden layer, and n_h is the number of hidden units in the LSTM cell.

From Table IV, we see that for the same equalizer type and similar RMpS, the averaged latency values are also very similar. For example, for topologies 1, 2, and 3, the latency of the MLP equalizer was around 7.8×10^{-5} s, and for the biLSTM equalizer it was around 1.2×10^{-4} s. Also, we can infer that the number of weights in the MLP is exactly equal to the RMpS. However, this equality is true only when the equalizer is solely composed of dense layers, as it is in the MLP case.

Regarding the number of parameters (trainable weights) in the equalizer, two main conclusions can be drawn from Table IV. First, for the same level of RMpS, the number of parameters for the MLP is much larger than that number for the biLSTM equalizer. This indicates that comparing equalizers with a different structure in terms of their number of parameters may be misleading. Then, we can also see that, for the biLSTM case, the increase in the number of parameters does not necessarily cause the increase in RMpS (in the CC) and the latency. This is an important observation since, theoretically, for our implemented sequential NN models, the latency should increase with the CC growth. For better visualization, in Fig. 18 we show the interrelation between the latency, RMpS, and the number of NN structure parameters. As we can see, the latency grows almost linearly with the RMpS metric, which confirms that our metric is a good estimate for the CC. On the other hand, we claim that the number of parameters does not represent a good estimation for the CC in the case of recurrent and convolutional layers, and it is not a good metric to compare the CC across different NN architectures. Therefore, to address the complexity analyses, we should always consider the RMpS value, while the number of NN parameters is relevant only if we are interested in the memory required by the NN or in the NN's training complexity.

The second point that we mention, refers to the memory access cost. In the case when we have approximately the same RMpS number, the biLSTM equalizer's latency is almost two times higher than that of the MLP equalizer; see Topology 7 from Table IV with the RMpS equal to $\approx 2 \times 10^6$. This effect is the



(a) MLP Equalizer Study

(b) biLSTM Equalizer Study

Fig. 18. Processing time to equalize one symbol vs. RMpS comparison, and vs. the number of trainable parameters in the respective NN.

consequence of Keras's implementation of the LSTM layer, in which the input data are subject to additional pre-processing operations to construct the state of each recurrent cell, and this pre-processing requires additional memory usage.[17] However, in our expression for the CC using RMpS, we do not account for such a memory usage impact. The cost of communication with the memory is usually much larger than the cost of "local" computation [85]. In this context, the term communication refers to the movement of data between the levels of memory or between multiple processors on a network. Therefore, when computing the CC of our equalizers, the proposition of using memory to save/access previous computations to reduce the number of multiplications (RMpS) must be taken with caution, as the cost of moving data (measured in time or energy expenditures) can (and usually does) exceed the cost of an arithmetic operation by orders of magnitude depending on the technology used [86], [87]. To illustrate this problem, we notice that by using modern SRAMs [88], the read and write estimated latencies for a 64 Kb SRAM, used for the NN, are $\sim$1 ns; for the most costly operations in NN inference (processing), the multiplications, [89] shows that a 16$\times$16 bits multiplier can have up to $\sim$10 ns estimated latency. Consider the scenario where the multiplication in our NN model is used 100 times. For a parallel implementation, 100 multipliers could be used and the total latency would be just 10 ns. But, turning to the NN implementation using the SRAM to save the multiplications, we need to consider that SRAMs usually has one read port and one write port, so only one parameter can be read at a time. Therefore, the total value would be 10 ns of the multiplication plus 1 ns to write in the SRAM, plus 100 times we read the values of this multiplication, which eventually gives us the estimate: 111 ns. Thus, we see that SRAM usage, in fact, becomes the processing bottleneck that drastically increases overall latency. This is the infamous latency versus complexity trade-off problem in hardware implementation, where we can use fewer resources of our hardware by sacrificing the latency, and vice versa, more resources allow us to reduce the processing time. Therefore, when going to the level of design for the NN equalizer using some SRAM, it is not completely fair to directly compare the reduced CC (the RMpS number) of such solutions to the CC of some traditional DSP techniques (say, to those of traditional digital back-propagation, the often-used benchmark), since SRAM will also effectively add to the complexity when implementing the particular design in hardware. Instead, we can say that by using advanced compression techniques and hardware technologies, a model that shares the same multiplications/parameters across its architecture and uses the SRAM to repeatedly access them, can potentially be less complex in terms of the hardware implementation at the expense of worse latency.

Finally, one more direction to account for when we assess the complexity refers to NN pruning and quantization techniques applied to the developed equalizer's structure [90], [91]. The purpose of quantization and pruning is to make the NN implementation in hardware more resource-efficient. In this context, pruning refers to the practice of removing weights from the original trained model. We utilize the number of bit operations (BoPs) metric [92] to measure NN's CC because the BoP metric is especially important when evaluating the performance of mixed-precision arithmetic in hardware implementations on FPGAs or ASICs. As described in [92], the BoPs of a dense layer with n neurons and input with m features can be expressed as:

$$\mathrm{BoP}_{\mathrm{dense}} = nm\left(b_w b_i + b_w + b_i + \lceil \log_2(m) \rceil\right), \qquad (11)$$

where, in the context of the NN, b_w is the number of bits used to represent the weights of the NN, b_i is the number of bits used to represent the input/ activation function, and $\lceil x \rceil$ is a ceiling function. From this equation, we observe that the number of multiplications is multiplied by $b_w b_i$, and the number of additions is multiplied by $b_w + b_i + \lceil \log_2(m) \rceil$. The latter is the actual bit width of the accumulator needed in MAC operations (the accumulator is a register in which the results of the intermediate arithmetic logic unit are stored). Also, note that this expression considers a dense layer with weight-matrix multiplication

[17]Note that the latency would naturally change when using GPUs since the implementation of LSTM changes for such hardware.

TABLE V
OVERVIEW OF MAIN ISSUES IN DEVISING EFFICIENT NN-BASED EQUALIZERS

Topic	Problem	Solution
QoT Metrics	The "jail window" effect in the NN-based equalizers. The use of the metrics assuming Gaussian channel statistics, such as EVM and SNR can lead to overestimated results in terms of BER.	To avoid overestimation it is recommended to always report the results in terms of BER or Q-factor calculated directly from BER.
PRBS Order	Depending on the PRBS order, the NN can learn the PRBS periodicity or the generation rule of the PRBS sequence that is based on its linear feedback shift register.	The training and testing datasets must be different fragments within the PRBS periodicity range. This periodicity depends on the PRBS order and the modulation format. Also, high PRBS orders should be used, ($\geq$32) , and the NNs are to be tested with a B2B system to guarantee the absence of overestimation and overfitting effects.
DAC Memory	Depending on the DSP implementation, the DAC memory limits the data variability to a few thousand samples, which makes the NN to overfit quickly into the training dataset statistics.	Increase the variability by, concatenating different traces generated with different random seeds to create a large training dataset of more than 13M symbols, and train the NN with different random 1M symbols from this training dataset for each epoch.
Regression vs Classification	The regression and classification have some statistical and machine learning drawbacks that can impact the learning in the field of nonlinearity mitigation in coherent optical channels.	In all our tests regression has performed better than classification. However, we mention that this check always needs to be done. For fairness reasons, take into account that mini-batch size, learning rate, and the number of epochs need to be individually tuned for each configuration since the gradients are different when using CEL or MSE.
Batch Size Study	The batch size is an important hyperparameter to guarantee a proper learning process of channel equalization. Using small mini-batch sizes, in the regression can make the NN move the learning in a direction that is not the best.	We observed that using a large mini-batch size ($\geq$1000) is good for generalization when using regression for NN-based equalizers. We see that when using small mini-batches, the NN was not generalizing for the different levels of nonlinearity for points in the QAM constellation
Computational Complexity	The computational complexity can be confused with the number of NN parameters. Also, it can be mistakenly underestimated if one uses memory to save some common multiplications. The underestimation can take place as well if quantization and pruning are not accounted for accurately.	We recommend reporting the computational complexity in terms of real multiplications per recovered symbol. In the same way, when using the quantization and pruning, the CC should be reported in terms of BoPs, Eq (11). Also, when reporting the CC values considering that some computations can be saved to be used later, it needs to be clearly stated that latency is not a constraint.

and bias addition, otherwise, the number of additions would be $n(m-1)$. Now, considering unstructured pruning (that is, we remove the least important connections/weights rather than entire layers or neurons), we can include the sparsity impact in (11) as follows:

$$\mathrm{BoP}_{\mathrm{dense}} = nm(b_w b_i \times (1-\eta_s) + b_w + b_i + \lceil \log_2(m) \rceil). \quad (12)$$

According to the equation above, the CC grows quadratically with bit widths and linearly with the pruning/sparsity ratio η_s, which is the percentage of connections erased from the layer. Now, considering the case with multiple layers, let us take, as an example, the MLP structure with 3 hidden layers (with the number of neurons $n_1/n_2/n_3$), input features number m, and output features number n_o. If we assume that the weights have the same quantization characterized by the number of bits b_w throughout the structure, and each input/activation function with b_i bits, as well as each layer is pruned equally with the sparsity ratio η_s, the BoPs for the MLP can be described in terms of the RMpS metric from (8) as:

$$\mathrm{BoP}_{\mathrm{MLP}} = C_{MLP_3}(b_w b_i \times (1-\eta_s) + b_w + b_i) + \mathrm{ACC}, \quad (13)$$

where "ACC" is part of the cost attributed to the accumulator and in this case is equal to $mn_1\lceil \log_2(m) \rceil + n_1 n_2 \lceil \log_2(n_1) \rceil + n_2 n_3 \lceil \log_2(n_2) \rceil + n_3 n_o \lceil \log_2(n_3) \rceil$.

However, regarding (13) we, first, stress that it is not applicable when pruning is non-uniform:[18] If another type of pruning is used, the term $C_{MLP_3} \times (1-\eta_s)$ has to be replaced by the total number of multiplications emerging from a particular non-uniform pruning scheme. Second, when dealing with quantization, we should be aware of all quantization levels which are used in the NN equalizer. As mentioned in [93], different quantization levels can be applied in the activation functions, input, and weights of each layer, and this obviously affects the overall complexity. Therefore, if different quantizations are used in different layers or the input, this must be counted separately, multiplying $b_w b_i$ per layer as in [91]. It is a common error to assume that when quantizing refers to just the weights, the complexity would drop quadratically: the CC would drop quadratically only *if we equally reduce the quantization bit width of both the input and activation function*.

Notice that, theoretically, the signal/activation function quantization has a floor depending on the modulation format used in the transmission. Recall that each M-QAM symbol denotes a single number from a set of M symbols in the QAM alphabet. That is, the real and imaginary components of each M-QAM symbol are represented by a number in the range $\sqrt{M}$. As a result, both real and imaginary parts' resolution should be significantly higher than $\log_2(\sqrt{M})$. For example, in a 64-QAM system, the real/imaginary parts of each symbol must be represented by at least $\log_2(\sqrt{64}) = 3$ quantization levels (bits in a quantized quantity's representation), and the minimal number of levels in the quantized complex symbol's representation should be no less than 3+3 bits in total. However, in the exemplary case of 64-QAM, using a value close to 6 bits to represent the NN's complex-valued input typically results in a substantial performance degradation because the system would virtually always make hard decisions and lose crucial equalization features. Regarding the quantization of the weights, the bit-precision for the weights is more flexible: it can be as low as allowed by the performance level of the model, and such advanced techniques as, e.g., the quantization-aware training [93] can be used to achieve the low-bit quantization of the weights.

IX. CONCLUSION

In this paper, we revealed and studied important hidden caveats and pitfalls that we have observed in recent publications and our own research dealing with the applications of machine

[18]Non-uniform pruning means that different layers of the NN model are pruned with different sparsity ratios.

learning methods and, particularly, the NNs for nonlinear impairments' compensation in coherent optical communications. We have grouped our original results here in such a way that this paper can also serve as a guidance and tutorial in this rapidly-growing field, drawing out the lessons learned and aiming at somewhat navigating the researchers in the area. We note that this work is not purely a traditional review insofar as we presented a number of completely new results regarding the difficulties in designing efficient NN equalizers. We believe that our results can foster new concepts and artificial intelligence techniques, allowing researchers to avoid known design process pitfalls and misinterpretation of results or findings. We underline again the challenges and common misconception errors that often occur in the development of NN-based equalizers applied in high-speed coherent optical communication systems: a poor model generalization, dependent dataset characteristics (periodicity), overfitting behavior and indications, performance overestimation, and inaccuracy in estimating the computational complexity. For convenience, our findings and recommendations are summoned up in Table V. Although this exploration of pitfalls is, without a doubt, far from complete, we believe that we have covered the most common problems that pose a particularly high risk for the design of efficient NN-based equalizers and other machine learning structures used in optical transmission lines.

REFERENCES

[1] M. A. Jarajreh *et al.*, "Artificial neural network nonlinear equalizer for coherent optical OFDM," *IEEE Photon. Technol. Lett.*, vol. 27, no. 4, pp. 387–390, Feb. 2015.

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

[3] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[4] S. Hunt *et al.*, "Adaptive electrical signal post-processing with varying representations in optical communication systems," in *Engineering Applications of Neural Networks*. Berlin, Germany: Springer, 2009, pp. 235–245.

[5] T. A. Eriksson, H. Bülow, and A. Leven, "Applying neural networks in optical communication systems: Possible pitfalls," *IEEE Photon. Technol. Lett.*, vol. 29, no. 23, pp. 2091–2094, Dec. 2017.

[6] D. Wang *et al.*, "System impairment compensation in coherent optical communications by using a bio-inspired detector based on artificial neural network and genetic algorithm," *Opt. Commun.*, vol. 399, pp. 1–12, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401817303346

[7] C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *Proc. IEEE Opt. Fiber Commun. Conf. Expo.*, 2018, pp. 1–3.

[8] S. Zhang *et al.*, "Field and lab experimental demonstration of nonlinear impairment compensation using neural networks," *Nature Commun.*, vol. 10, no. 1, pp. 1–8, 2019.

[9] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," *IEEE Commun. Surv. Tut.*, vol. 21, no. 2, pp. 1383–1408, Jun. 2019.

[10] F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An optical communication's perspective on machine learning and its applications," *J. Lightw. Technol.*, vol. 37, no. 2, pp. 493–516, Jan. 2019.

[11] P. J. Freire *et al.*, "Complex-valued neural network design for mitigation of signal distortions in optical links," *J. Lightw. Technol.*, vol. 39, no. 6, pp. 1696–1705, 2021.

[12] M. Schädler, G. Böcherer, and S. Pachnicke, "Soft-demapping for short reach optical communication: A comparison of deep neural networks and volterra series," *J. Lightw. Technol.*, vol. 39, no. 10, pp. 3095–3105, 2021.

[13] X. Lin *et al.*, "Perturbation theory-aided learned digital back-propagation scheme for optical fiber nonlinearity compensation," *J. Lightw. Technol.*, vol. 40, no. 7, pp. 1981–1988, 2022.

[14] P. J. Freire *et al.*, "Performance versus complexity study of neural network equalizers in coherent optical systems," *J. Lightw. Technol.*, vol. 39, no. 19, pp. 6085–6096, 2021.

[15] P. J. Freire *et al.*, "Transfer learning for neural networks-based equalizers in coherent optical systems," *J. Lightw. Technol.*, vol. 39, no. 21, pp. 6733–6745, 2021.

[16] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.

[17] M. Kuschnerov *et al.*, "Data-aided versus blind single-carrier coherent receivers," *IEEE Photon. J.*, vol. 2, no. 3, pp. 387–403, Jun. 2010.

[18] G. P. Agrawal, *Nonlinear Fiber Optics*, 5th ed., Boston, MA, USA: Academic Press, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123970237000024

[19] A. Gulli and S. Pal, *Deep Learning With Keras*. Birmingham, U.K.: Packt Publishing, 2017.

[20] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[21] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian optimization of machine learning hyperparameters on large datasets," in *Proc. Artif. Intell. Statist.*, 2017, pp. 528–536.

[22] A. Alvarado, E. Agrell, D. Lavery, R. Maher, and P. Bayvel, "Replacing the soft-decision FEC limit paradigm in the design of optical communication systems," *J. Lightw. Technol.*, vol. 33, no. 20, pp. 4338–4352, 2015.

[23] M. D. McKinley *et al.*, "EVM calculation for broadband modulated signals," in *Proc. 64th ARFTG Conf. Dig.*, Orlando, FL, USA, 2004, pp. 45–52.

[24] R. Schmogrow *et al.*, "Error vector magnitude as a performance measure for advanced modulation formats," *IEEE Photon. Technol. Lett.*, vol. 24, no. 1, pp. 61–63, Jan. 2012.

[25] W. Freude *et al.*, "Quality metrics for optical transmission," in *Proc. IEEE Int. Conf. Photon. Switching*, 2012, pp. 1–4.

[26] R. A. Shafik, M. S. Rahman, and A. R. Islam, "On the extended relationships among EVM, BER and SNR as performance metrics," in *Proc. IEEE Int. Conf. Elect. Comput. Eng.*, 2006, pp. 408–411.

[27] T. A. Eriksson, T. Fehenberger, and W. Idler, "Characterization of nonlinear fiber interactions using multidimensional mutual information over time and polarization," *J. Lightw. Technol.*, vol. 35, no. 6, pp. 1204–1210, 2017.

[28] T. A. Eriksson *et al.*, "Four-dimensional estimates of mutual information in coherent optical communication experiments," in *Proc. IEEE Eur. Conf. Opt. Commun.*, 2015, pp. 1–3.

[29] T. Catuogno, M. R. Camara, and M. Secondini, "Non-parametric estimation of mutual information with application to nonlinear optical fibers," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 736–740.

[30] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[31] O. Kramer, "Scikit-learn," in *Machine Learning for Evolution Strategies*. Berlin, Germany: Springer, 2016, pp. 45–53.

[32] M. Schaedler *et al.*, "Deep neural network equalization for optical short reach communication," *Appl. Sci.*, vol. 9, no. 21, 2019, Art. no. 4675.

[33] J. Zhang *et al.*, "Functional-link neural network for nonlinear equalizer in coherent optical fiber communications," *IEEE Access*, vol. 7, pp. 149900–149907, 2019.

[34] L. Liu *et al.*, "OLS-based RBF neural network for nonlinear and linear impairments compensation in the CO-OFDM system," *IEEE Photon. J.*, vol. 10, no. 2, pp. 1–8, Apr. 2018.

[35] S. Liu *et al.*, "An effective artificial neural network equalizer with s-shape activation function for high-speed 16-QAM transmissions using low-cost directly modulated laser," in *Proc. Int. Conf. Netw. Infrastructure Digit. Content*, 2018, pp. 269–273.

[36] J. L. Julus, D. Manimegalai, A. C. Beaula, J. J. Athanesious, and A. A. Roobert, "Advanced nonlinear equalizer for filter bank multicarrier-based long reach-passive optical network system," *Int. J. Commun. Syst.*, vol. 34, no. 14, 2021, Art. no. e4921, doi: 10.1002/dac.4921.

[37] O. Kotlyar *et al.*, "Convolutional long short-term memory neural network equalizer for nonlinear Fourier transform-based optical transmission systems," *Opt. Exp.*, vol. 29, no. 7, pp. 11254–11267, Mar. 2021. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-29--7-11254

[38] H. Ming *et al.*, "Ultralow complexity long short-term memory network for fiber nonlinearity mitigation in coherent optical communication systems," *J. Lightw. Technol.*, vol. 40, no. 8, pp. 2427–2434, Apr. 2022.

[39] I. Fatadin, "Estimation of BER from error vector magnitude for optical coherent systems," *Photonics*, vol. 3, no. 2. 2016, Art. no. 21.

[40] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Commun. ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[41] Z. Yang, F. Gao, S. Fu, M. Tang, and D. Liu, "Overfitting effect of artificial neural network based nonlinear equalizer: From mathematical origin to transmission evolution," *Sci. China Inf. Sci.*, vol. 63, pp. 1–13, 2020.

[42] C. M. Bishop *et al.*, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 1995.

[43] G. An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural Comput.*, vol. 8, no. 3, pp. 643–674, 1996.

[44] A. Neelakantan *et al.*, "Adding gradient noise improves learning for very deep networks," 2015, *arXiv:1511.06807*.

[45] C. Wang and J. Principe, "Training neural networks with additive noise in the desired signal," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1511–1517, Nov. 1999.

[46] Y. Grandvalet, S. Canu, and S. Boucheron, "Noise injection: Theoretical prospects," *Neural Comput.*, vol. 9, no. 5, pp. 1093–1108, 1997.

[47] S. Yin *et al.*, "Noisy training for deep neural networks in speech recognition," *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, no. 1, pp. 1–14, 2015.

[48] S. Rifai, X. Glorot, Y. Bengio, and P. Vincent, "Adding noise to the input of a model trained with a regularized objective," 2011, *arXiv:1104.3250*.

[49] W. M. Brown, T. D. Gedeon, and D. I. Groves, "Use of noise to augment training data: A neural network method of mineral–potential mapping in regions of limited known deposit examples," *Natural Resour. Res.*, vol. 12, no. 2, pp. 141–152, 2003.

[50] P. J. Freire, J. E. Prilepsky, Y. Osadchuk, S. K. Turitsyn, and V. Aref, "Deep neural network-aided soft-demapping in optical coherent systems: Regression versus classification," 2021, *arXiv:2109.13843*.

[51] H. A. K. Rady, "Shannon entropy and mean square errors for speeding the convergence of multilayer neural networks: A comparative approach," *Egyptian Informat. J.*, vol. 12, no. 3, pp. 197–209, 2011.

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[53] L. N. Binh, *Noises in Optical Communications and Photonic Systems*. Boca Raton, FL, USA: CRC Press, 2019.

[54] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8792–8802.

[55] G. Bocherer, "Lecture notes on machine learning for communications," 2021. [Online]. Available: http://georg-boecherer.de/mlcomm-v0.pdf

[56] M. Pazzani *et al.*, "Reducing misclassification costs," in *Proc. Mach. Learn. Proc.*, 1994, pp. 217–225.

[57] P. A. Gutiérrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez, "Ordinal regression methods: Survey and experimental study," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 127–146, Jan. 2016.

[58] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2065–2081, Sep. 2020.

[59] A. S. Bosman, A. Engelbrecht, and M. Helbig, "Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions," *Neurocomputing*, vol. 400, pp. 113–136, 2020.

[60] S. Wang, F. Liu, and B. Liu, "Escaping the gradient vanishing: Periodic alternatives of softmax in attention mechanism," *IEEE Access*, vol. 9, pp. 168749–168759, 2021.

[61] Z. Niu *et al.*, "End-to-end deep learning for long-haul fiber transmission using differentiable surrogate channel," *J. Lightw. Technol.*, vol. 40, no. 9, pp. 2807–2822, 2022.

[62] Y. Li *et al.*, "Online human action detection using joint classification-regression recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 203–220.

[63] L. Mukherjee, M. A. K. Sagar, J. N. Ouellette, J. J. Watters, and K. W. Eliceiri, "Joint regression-classification deep learning framework for analyzing fluorescence lifetime images using NADH and FAD," *Biomed. Opt. Exp.*, vol. 12, no. 5, pp. 2703–2719, 2021.

[64] M. Liu, J. Zhang, E. Adeli, and D. Shen, "Joint classification and regression via deep multi-task multi-channel learning for Alzheimer's disease diagnosis," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1195–1206, May 2019.

[65] M. Liu, J. Zhang, E. Adeli, and D. Shen, "Deep multi-task multi-channel learning for joint classification and regression of brain status," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, Springer, 2017, pp. 3–11.

[66] J.-Y. Wu, M. Wu, Z. Chen, X. Li, and R. Yan, "A joint classification-regression method for multi-stage remaining useful life prediction," *J. Manuf. Syst.*, vol. 58, pp. 109–119, 2021.

[67] J. Chen *et al.*, "Joint learning with both classification and regression models for age prediction," in *Proc. J. Phys., Conf. Ser.*, IOP Publishing, 2019, vol. 1168, no. 3, Art. no. 032016.

[68] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[69] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[70] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Netw.*, vol. 16, no. 10, pp. 1429–1451, 2003.

[71] S. L. Smith and Q. V. Le, "A Bayesian perspective on generalization and stochastic gradient descent," 2017. [Online]. Available: https://arxiv.org/abs/1710.06451

[72] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2016. [Online]. Available: https://arxiv.org/abs/1609.04836

[73] M. Wandel *et al.*, "Dispersion compensating fibers for non-zero dispersion fibers," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2002, pp. 327–329.

[74] L. Peng, M. Hélard, and S. Haese, "On bit-loading for discrete multi-tone transmission over short range POF systems," *J. Lightw. Technol.*, vol. 31, no. 24, pp. 4155–4165, 2013.

[75] J.-W. Kim and C.-H. Lee, "Modulation format identification of square and non-square M-QAM signals based on amplitude variance and OSNR," *Opt. Commun.*, vol. 474, 2020, Art. no. 126084.

[76] L. Nadal, J. M. Fàbrega, J. Vílchez, and M. S. Moreolo, "Experimental analysis of 8-QAM constellations for adaptive optical OFDM systems," *IEEE Photon. Technol. Lett.*, vol. 28, no. 4, pp. 445–448, Feb. 2016.

[77] J. Wang *et al.*, "Capacity of 60 GHz wireless communications based on QAM," *J. Appl. Math.*, vol. 2014, no. SI03, 2014, doi: 10.1155/2014/815617.

[78] J. Brownlee, *Better Deep Learning. Train Faster, Reduce Overfitting, and Make Better Predictions*. vol. 1. Machine Learning Mastery Python, 2018.

[79] S. Ong, C. You, S. Choi, and D. Hong, "A decision feedback recurrent neural equalizer as an infinite impulse response filter," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2851–2858, Nov. 1997.

[80] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[81] M. Mikami and H. Yoshino, "Field trial on 5G low latency radio communication system towards application to truck platooning," *IEICE Trans. Commun.*, 2019, Art. no. 2018TTP0021.

[82] S. R. Pokhrel, N. Kumar, and A. Walid, "Towards ultra reliable low latency multipath TCP for connected autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8175–8185, Aug. 2021.

[83] E. Ponomarev, S. Matveev, I. Oseledets, and V. Glukhov, "Latency estimation tool and investigation of neural networks inference on mobile GPU," *Computers*, vol. 10, no. 8, 2021, Art. no. 104.

[84] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Architecture*, 2017, pp. 1–12.

[85] P. Pacheco, *An Introduction to Parallel Programming*. New York, NY, USA: Elsevier, 2011.

[86] J. Demmel, "Communication avoiding algorithms," in *Proc. IEEE SC Companion: High Perform. Comput., Netw. Storage Anal.*, 2012, pp. 1942–2000.

[87] G. Ballard *et al.*, "Communication lower bounds and optimal algorithms for numerical linear algebra," *Acta Numerica*, vol. 23, pp. 1–155, 2014.

[88] M. Imani, S. Patil, and T. Rosing, "Low power data-aware STT-RAM based hybrid cache architecture," in *Proc. 17th Int. Symp. Qual. Electron. Des.*, 2016, pp. 88–94.

[89] S. Ullah *et al.*, "Area-optimized low-latency approximate multipliers for FPGA-based hardware accelerators," in *Proc. 55th Annu. Des. Automat. Conf.*, 2018, pp. 1–6.

[90] S. Fujisawa *et al.*, "Weight pruning techniques towards photonic implementation of nonlinear impairment compensation using neural networks," *J. Lightw. Technol.*, vol. 40, no. 5, pp. 1273–1282, 2022.
[91] D. R. Argüello *et al.*, "Experimental implementation of a neural network optical channel equalizer in restricted hardware using pruning and quantization," 2022. [Online]. Available: https://doi.org/10.21203/rs.3.rs-1236203/v1
[92] B. Hawks *et al.*, "PS and QS: Quantization-aware pruning for efficient low latency neural network inference," *Front. Artif. Intell.* vol. 4, 2021, Art. no. 676564.
[93] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2704–2713.

Pedro J. Freire (Graduate Student Member, IEEE) received the bachelor's and master's degrees in electronic engineering from the Federal University of Pernambuco, Recife, Brazil. He had a one-year and a-half year tenure with the State University of New York and the State University of San Francisco, San Francisco, CA, USA, respectively. He is currently an Early Stage Researcher with Aston University, Birmingham, U.K., and Infinera. His master's thesis in the photonics area was dealt with a new metaheuristic using evolutionary algorithms to solve the RMLSA problem. His research interests include advanced digital signal processing and coding, network monitoring and planning, artificial intelligence, machine learning, and hardware DSP realization. He currently holds a Marie-Curie (MSCA) Doctoral Fellowship.

Antonio Napoli received the Ph.D. degree from the Polytechnic University of Turin, Turin, Italy, in 2006. In 2018, he joined Infinera and working on XR Optics. He currently represents Infinera at OIF. He is the author or co-author of seven patents, 193 articles, and a book chapter. His research interests include DSP, wide-band systems, modeling, and ML/AI. He led three special issues on JOCN/JLT. Dr. Napoli was an OFC TPC member and co-organized four OFC workshops. He is also the Technical Coordinator of three H2020 projects.

Bernhard Spinnler received the Dipl.-Ing. degree in communications engineering and the Dr.-Ing. degree from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1994 and 1997, respectively. Since 1997, he has worked on low-complexity modem design of wireless radio relay systems at Siemens AG, and information and communication networks. In 2002, he joined the Optical Networks Group, Siemens Corporate Technology, which became Nokia Siemens Networks and later Coriant. He is currently with Infinera on the design and modeling of robust, tolerant, and automated optical communications systems. His research interests include advanced digital signal processing and coding, network automation, artificial intelligence, and machine learning.

Nelson Costa was born in Tomar, Portugal, in 1983. He received the Licenciatura and Ph.D. degrees in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon Technical University, Lisbon, Portugal, in 2006 and 2012, respectively. He is currently with Infinera Portugal, Carnaxide, Portugal. He has authored or coauthored more than 50 publications in international conferences and journals. His current research interests include advanced coherent optical modulation formats, nonlinear fiber transmission effects, and network optimization.

Sergei K. Turitsyn (Senior Member, IEEE) graduated from the Department of Physics, Novosibirsk University, Novosibirsk, Russia, in 1982, and received the Ph.D. degree in theoretical and mathematical physics from the Budker Institute of Nuclear Physics, Novosibirsk, Russia, in 1986. In 1992, he moved to Germany, first as a Humboldt Fellow and then working in the collaborative projects with Deutsche Telekom. Since 2012, he has been the Director of the Aston Institute of Photonic Technologies, Aston University, Birmingham, U.K., which is a world-known photonics research center, with a strong track record of academic achievements, a range of developed technologies, and industrial collaborations. Prof. Turitsyn is the Originator of several key concepts in the fields of nonlinear science, optical fiber communications, and fiber lasers. He was/is a Principal Investigator in 67 national and international, research, and industrial projects. He is a member of the Editorial Board (Electronics, Photonics, and Device Physics) of the Scientific Reports, Nature Publishing Group, and an Associate Editor for JLT. He was the recipient of the Royal Society Wolfson Research Merit Award in 2005, European Research Council, Advanced Grant in 2011, Lebedev Medal from the Rozhdestvensky Optical Society in 2014, Aston 50th Anniversary Chair Medal in 2016, and Chancellor's Medal in 2018. He is a Fellow of the Optical Society of America and the Institute of Physics.

Jaroslaw E. Prilepsky received the M.E. degree (Hons.) in theoretical physic from the National University of Kharkiv, Kharkiv, Ukraine, in 1999, and the Ph.D. degree in theoretical physics from the B. Verkin Institute for Low-Temperature Physics and Engineering, Kharkiv, Ukraine, in 2003, with a focus on nonlinear excitation in low-dimensional systems. From 2003 to 2010, he was a Research Fellow with the B. Verkin Institute for Low-Temperature Physics and Engineering. From 2010 to 2012, he was a Research Associate with the Nonlinearity and Complexity Research Group, Aston University, Birmingham, U.K., and since 2012, he has been a Research Fellow with the Aston Institute of Photonics Technologies, Aston University. He has authored more than 80 journal papers and conference contributions in the fields of nonlinear physics and mathematics, solitons, nonlinear signal-noise interaction, optical transmission, and signal processing. His current research interests include optical transmission systems and networks, nonlinearity mitigation methods, neural networks and machine learning, nonlinear Fourier-based optical transmission methods, and optical signal processing.