APL Photonics

RESEARCH ARTICLE | DECEMBER 29 2021

Machine learning for optical fiber communication systems: An introduction and overview

Special Collection: Photonics and AI in Information Technologies

Josh W. Nevin; Sam Nallaperuma; Nikita A. Shevchenko; Xiang Li; Md. Saifuddin Faruk;
Seb J. Savory

APL Photonics 6, 121101 (2021)
https://doi.org/10.1063/5.0070838

Articles You May Be Interested In

Intelligent performance inference: A graph neural network approach to modeling maximum achievable throughput in optical networks

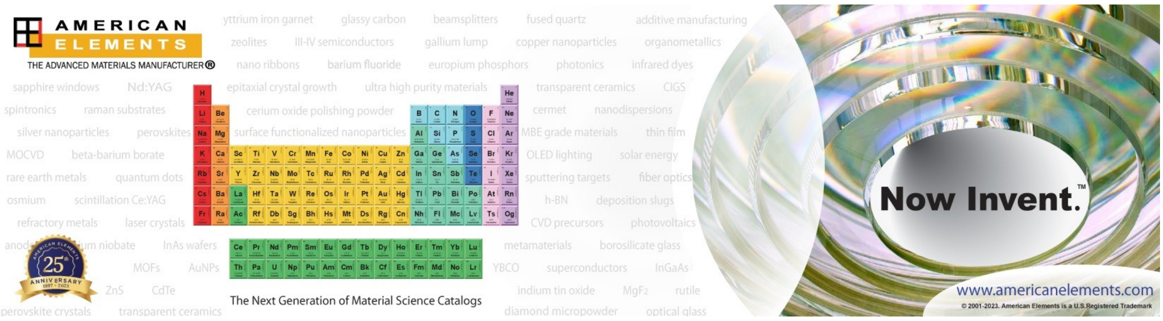
APL Machine Learning (May 2023)

AI boosts photonics and vice versa

APL Photonics (July 2020)

Real-time machine learning based fiber-induced nonlinearity compensation in energy-efficient coherent optical networks

APL Photonics (April 2020)


AMERICAN ELEMENTS
THE ADVANCED MATERIALS MANUFACTURER®
The Next Generation of Material Science Catalogs
Now Invent.™
www.americanelements.com
© 2001-2023. American Elements is a U.S. Registered Trademark

AIP Publishing

Machine learning for optical fiber communication systems: An introduction and overview

Cite as: APL Photon. 6, 121101 (2021); doi: 10.1063/5.0070838
Submitted: 9 September 2021 • Accepted: 6 December 2021 •
Published Online: 29 December 2021

View Online Export Citation CrossMark

Josh W. Nevin, Sam Nallaperuma, Nikita A. Shevchenko, Xiang Li, Md. Saifuddin Faruk, and Seb J. Savory[a)]

AFFILIATIONS

Fibre Optic Communication Systems Laboratory, Electrical Engineering Division, Department of Engineering, University of Cambridge, 9 JJ Thomson Avenue, Cambridge CB3 0FA, United Kingdom

Note: This paper is part of the APL Photonics Special Topic on Photonics and AI in Information Technologies.
[a)]**Author to whom correspondence should be addressed:** sjs1001@cam.ac.uk

ABSTRACT

Optical networks generate a vast amount of diagnostic, control, and performance monitoring data. When information is extracted from these data, reconfigurable network elements and reconfigurable transceivers allow the network to adapt not only to changes in the physical infrastructure but also to changing traffic conditions. Machine learning is emerging as a disruptive technology for extracting useful information from these raw data to enable enhanced planning, monitoring, and dynamic control. We provide a survey of the recent literature and highlight numerous promising avenues for machine learning applied to optical networks, including explainable machine learning, digital twins, and approaches in which we embed our knowledge into machine learning such as physics-informed machine learning for the physical layer and graph-based machine learning for the networking layer.

I. INTRODUCTION

Machine learning (ML) is the study of computer algorithms that can learn to achieve a given task via experience and data without being explicitly programmed.[1] ML has been a topic of research within statistics and computer science since at least the 1950s, with early iterations of many algorithms used today invented in the last 30 years.[2] However, as a result of the increase in the availability of data and computing power over time, the use of ML has recently become ubiquitous across all disciplines of science and engineering. Optical fiber communications is no exception—there are now a great many works utilizing a range of ML techniques to solve a range of problems within the domain. This is reflected in a large number of review and tutorial papers that have been published on the subject of ML applied to optical networks.[3–8] However, given the rapid acceleration of the usage of ML within optical networks, there have been many works published in the domain that leverage ML since these reviews were conducted. Moreover, certain ML applications have recently begun to increase in popularity for optical networks problems, which we address in this Tutorial. Thus, in this Tutorial, we introduce the reader to ML, highlight the key ML techniques being deployed within optical fiber communication systems presently, and outline recent impactful works within each application sub-domain.

Optical fiber communication systems form the backbone of communications, having been deployed across the globe since the early 1980s.[9] At a basic level, the edges of optical fiber networks are composed of optical fibers carrying modulated laser light, with optical amplifiers to combat loss of laser signal power incurred during propagation. The nodes of optical networks are comprised of transmitters, receivers, and switches. Loosely, the job of network operators is to carry messages between these nodes such that the quality of service agreed to customers is met. Different modulated laser signals, known as channels, are assigned different individual wavelengths and can then be transmitted through the same fiber link simultaneously—this is known as wavelength division multiplexing (WDM). Telecommunication systems are split into conceptual layers defined by the open systems interconnection model,[10] and in this Tutorial, we reference applications of ML in layers one and two, which we refer to as the physical layer and the network layer, respectively. In short, the physical layer concerns how raw bits are transmitted across a link between two nodes, also known as a light path. Contrastingly, network layer applications concern how to transfer data across the physical layer between given nodes. As an example, one can control aspects such as the route taken through the network, meaning the sequence of edges and nodes traversed, and the chosen wavelength channel that is used to carry the information between two nodes. Additionally, optical networks research is commonly carried out on specific network types, which are primarily defined by their scale. In ascending order of transmission length, these network types are access networks, which connect individual users to other users and data centers, metro networks at city scale, backbone networks at the scale of large countries and continents, and submarine systems, for connecting continents. Each of these network types has different constraints, for example, access networks have stringent monetary cost and complexity limits, whereas submarine systems have very strong power constraints. There are also data center networks, which are significantly different from all these network types due to their highly configurable topologies and extremely short reach links. In this Tutorial, we discuss works considering backbone, metro, and access networks.

Optical fiber communication systems facilitate the transfer of information at high data rates, currently 10–100 s (and in some cases, greater than 1000) of Mb/s,[11] enabling many data-hungry applications. In fact, Cisco predicts that there will be 5.3 × 10^9 internet users by 2023, an increase from 3.9 × 10^9 in 2018.[11] Moreover, the average connection speed is expected to rise from 45.9 Mb/s in 2018 to 110.4 Mb/s by 2023.[11] The optical fiber communication domain faces a number of key challenges that must be overcome to bring about this growth. First, optical fibers exhibit nonlinear behavior, governed by the optical Kerr effect.[12,13] This means that the refractive index seen by a given wavelength of laser light propagating through the fiber is dependent on the electric field strength in the fiber. As a result, channels interfere in a nonlinear way both with other channels on the same fiber and with themselves. These nonlinear noise-like distortions due to channel interference are power-dependent, meaning that there exists a trade-off between the optical power of the signal and the strength of these nonlinear interactions.[14] This introduces a level of complexity that makes physics-based modeling challenging in practical systems, making ML approaches look promising. Estimating the strength of nonlinear interaction and mitigating its effects form the basis of much of the research in optical fiber communication systems, including a large amount of works in which ML is applied. Furthermore, attempts to extend the range of wavelengths used to carry information beyond the traditional *C*−band, known as wide-band systems, require one to deal with some extra physical effects. Among them are the wavelength dependencies of fiber parameters, such as fiber loss (mainly, the elastic Rayleigh scattering[15]), higher-order fiber dispersion effects, and the influence of the frequency-dependent fiber effective mode area.[16] In addition, higher-order Kerr-type nonlinearities manifesting themselves as stimulated inelastic light scattering effects, i.e., stimulated Raman scattering (for very short optical pulses)[17–19] and stimulated Brillouin scattering (for very large launch powers),[20,21] should also be taken into consideration. ML approaches have shown potential in helping to deal with such effects, which may facilitate the use of wide-band systems in future networks.

Another critical problem in optical fiber communications is the high complexity of optical networks, which poses a significant operational challenge.[22] As networks have evolved over time to carry a higher information throughput, the modeling of the optical communication channel has become more difficult due to the increased number of adjustable design and operational parameters.[3] Perhaps the biggest driving force behind this has been the introduction of coherent technologies,[23] which increased the complexity of transmitters and receivers significantly. Moreover, the configurability of the network layer has increased due to advances such as software defined networking (SDN).[24] In addition, future optical networks will be more dynamic, requiring automation as requests must be satisfied on shorter time scales.[25] As a result, investigating the extent to which ML can help with modeling and network control has been the subject of a large volume of research. In this Tutorial, we focus on introducing the ML techniques that appear in the works we outline. Furthermore, we introduce a classification of algorithms in order to clarify the relationship between these techniques as well as outlining trends within optical communications such as which algorithm classes are used within each optical communication sub-domain.

The rest of this Tutorial is organized as follows. In Sec. II A, we introduce the general concept and nomenclature of ML, followed by a description of the specific techniques utilized by the works discussed in this Tutorial in Sec. II B. We then outline key research problems and selected interesting work within the physical layer in Sec. III, followed by an equivalent survey for network layer problems in Sec. IV. Selected opportunities for future research across both physical and network layer problems are highlighted in Sec. V, and concluding remarks are included in Sec. VI.

II. INTRODUCTION TO SELECTED MACHINE LEARNING TECHNIQUES

A. Categorization of machine learning

First, algorithms can be categorized based on the type of problem that is being solved, i.e., whether it is a regression or classification problem.[26] Regression algorithms make continuous predictions, such as the signal-to-noise ratio (SNR) of a light path in an optical network, and may have continuous or discrete inputs, also known as features. Classifier algorithms, instead, predict the class associated with a given set of inputs, for example, whether a request to connect two nodes in a network can be satisfied or rejected. A second distinction can be made based on whether the data are labeled or unlabeled.[26] Algorithms requiring labeled data are known as supervised, for instance, a dataset of SNR as a function of the signal power for an optical channel. Each datum in this set has a label, the measured SNR, which the algorithm can use as a target when learning. Contrastingly, unsupervised algorithms involve learning from unlabeled data. This can be done by attempting to group these data based on similarity—known as clustering, or compressing the data by finding the features that are most important for distinguishing between examples and removing the remaining features—known as principal component analysis.[26] An example of unlabeled data might be traffic flows in a network, which can be grouped into classes that are not pre-determined, but rather determined by the algorithm based on similarities in various features. There also exists another formulation of ML that is distinct from supervised and unsupervised learning, known as reinforcement learning (RL).[27] In RL, the goal is to learn a policy for achieving a given task by interacting with the environment. Every action taken affects the environment and returns a reward, the value of which quantifies how successful the given action was in the context of the overall goal. Formulations of RL and various algorithms are discussed in Sec. II B 4. An example of an application of RL in optical fiber networking might be an agent that learns an optimal routing policy, which maximizes the total throughput of the network, given a series of requests. Here, the environment may consist of the current network state and outstanding requests, and the action space (the set of allowed agent actions) may consist of a set of candidate routes and channel wavelengths to choose from.

A categorization of different ML techniques discussed in this Tutorial is outlined in Fig. 1. This diagram reflects the fact that supervised learning is more commonly used within optical communications than RL and unsupervised learning and that unsupervised learning is the least-used class of algorithms. Moreover, for physical layer applications, regression is more popular than classification, as we are often interested in predicting continuous target signals. Classifier algorithms are predominantly used in network layer applications where we are often interested in distinguishing candidate light paths that are suitably high quality from those that are not and predicting source and destination nodes of network traffic. Similarly, the majority of works applying RL in optical communications address network layer applications, which are often formulated as dynamic control problems. However, these are not absolute rules and there are exceptions. For example, Generative adversarial networks (GANs) and graph neural networks (GNNs) are commonly used in a regression formulation to tackle problems in network traffic prediction and generation. Rather, these are the general trends seen in the literature by the authors.

In the broader applied ML community, the types of data used can be categorized as structured tabular data, text data for natural language processing, image data consisting of sets of pixels, and time series data. The structure within tabular data may include spatial information, such as a graph, which can be represented as a matrix of edges and weights. Within optical fiber communications, the most common data types used are tabular and time series data. Furthermore, a further distinction can be drawn between batch and online learning. The more traditional batch learning approach involves learning from the whole training dataset, before deploying this model on new examples. Alternatively, online learning involves learning as data become available, updating the current model with information obtained from new examples.[28] In the case of a NN model, for instance, online learning would involve adapting the weights of a trained model based on a small volume of data. One could therefore train the NN initially on a large historical dataset before fine-tuning the weights using new data from monitors via online learning. In the supervised case, the new data will be labeled with an example of a label being the SNR for a given set of operating parameters. Unsupervised online learning is also possible, and online algorithms for principal component analysis and clustering using neural networks are available.[29] Here, the basic idea is to begin with a dataset that has been compressed in the case of principal component analysis or grouped in the case of clustering and modify the compression or grouping based on a new datum as it becomes available, rather than for all the data at once. Thus, the new datum is also compressed or grouped, which may, in turn, also change how the other data are compressed or grouped. A related approach to online learning is transfer learning, where we utilize information obtained from training a model for one task in order to reduce the computational effort required in training a model to perform another similar task.[30] In other words, transfer learning involves starting with a trained model for an old task and adapting it for the new target task, rather than starting from scratch. For example, one can modify the weights of a NN that has been trained for another task, rather than starting with untrained weights, reducing the computational requirements of training.

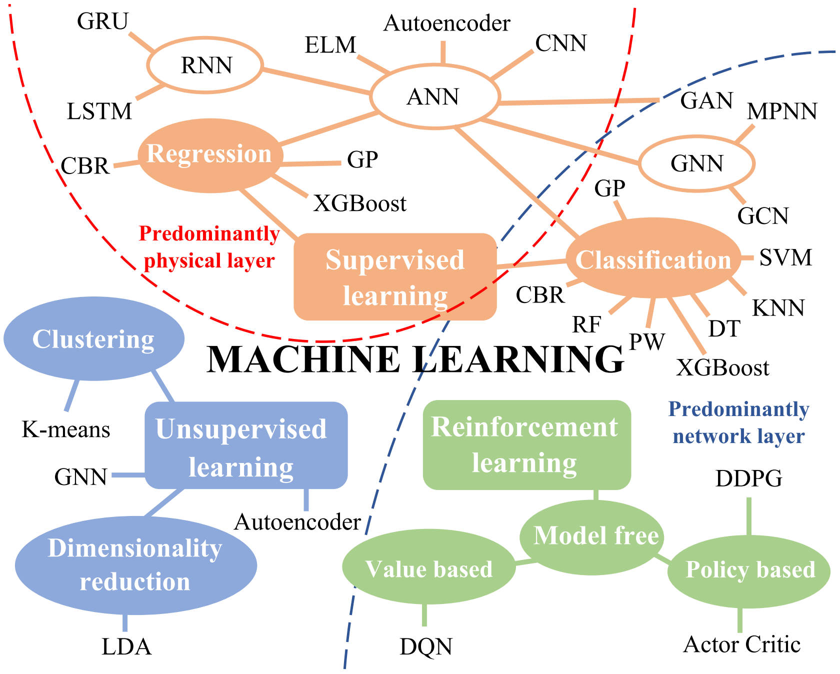


FIG. 1. Categorization of ML techniques discussed in this Tutorial. In general, supervised regression algorithms are more common in physical layer applications, whereas supervised classifiers and RL are more popular for network layer problems. Some techniques appear more than once as they can be formulated for different problem types (ANN: artificial neural network, ELM: extreme learning machine, CNN: convolutional neural network, GNN: graphical neural network, RNN: recurrent neural network, LSTM: long-short term memory, GRU: gated recurrent unit, GAN: generative adversarial network, MPNN: message-passing neural network, GP: Gaussian process, CBR: case-based reasoning, GCN: graph convolutional network, SVM: support vector machine, DT: decision tree, RF: random forest, KNN: K-nearest neighbor, PW: Parzen window, LDA: linear discriminant analysis, DQN: deep Q-network, and DDPG: deep deterministic policy gradient).

Finally, explainable ML is a growing field of ML that is crucial for ML applications as explainability increases confidence in ML systems.[31] In this work, we follow the definition of explainability given by Roscher *et al.*[32] Specifically, explainable ML is transparent and interpretable and leverages domain knowledge. In this context, transparency means that the design of the ML model can be justified beyond empirical performance on the testing dataset; interpretability means that the ML model output is human understandable—we can reason as to why the model makes a given prediction for a given input; and domain knowledge broadly encompasses all the knowledge of the problem we possess before we have seen the data. A black box is a model for which the decision processes are not interpretable by humans and the design cannot be easily justified.[33] There are two main approaches to explainability. First, there are those that accept that the underlying model is a black box and analyze the model's input–output relationship, in order to explain how it makes decisions and infer its internal structure.[34–36] Alternatively, there are those that try to replace the black box with a more simplistic or more mathematically principled model that is inherently more understandable. The former are commonly known as post-hoc techniques. Thus, a black box method can be made more explainable using extra add-on techniques or one can design the method from the ground up to be explainable.

B. Machine learning techniques

1. Neural networks

Neural networks (NNs) are universal function approximators, meaning that a sufficiently large NN structure can approximate any function.[37] The structure of NNs is analogous to that of animal brains, consisting of a network of units, called neurons, connected via edges with associated weights. The neurons can send signals to one another along these weighted edges and process these signals. The most commonly used type of NN in ML applications is a feedforward NN (FFNN). The mathematical structure of such networks is given by an input layer, followed by a series of layers of neurons, each representing a function that is applied to the previous layer in a chain rule-fashion.[38] The final layer yields the model output, and the layers in between the input and output layers are known as hidden layers. As an example, consider a supervised NN model with a single hidden layer $f^{(1)}$ and an output layer $f^{(2)}$,

$$f\left(\boldsymbol{x}\,\middle|\,\mathbb{W}^{(1)},\boldsymbol{b}^{(1)},\mathbb{W}^{(2)},\boldsymbol{b}^{(2)}\right)=f^{(2)}\left[f^{(1)}(\boldsymbol{x})\right], \tag{1}$$

$$f^{(i)}\triangleq g^{(i)}\left(\mathbb{W}^{\mathrm{T}(i)}\,\boldsymbol{x}^{(i)}+\boldsymbol{b}^{(i)}\right), \tag{2}$$

where $\boldsymbol{x}^{(i)}$ is the input vector for layer i such that $\boldsymbol{x}^{(1)}=\boldsymbol{x}$, $\mathbb{W}^{(i)}$ is the matrix of weights in layer i, $\boldsymbol{b}_i$ is a vector of additive constants known as biases in layer i, $g^{(i)}$ is the activation function applied element-wise to yield a vector output for layer i, and $(\cdot)^{\mathrm{T}}$ denotes the transpose of a given matrix. A pictorial representation of this NN, adapted from the work of Bishop,[26] is given in Fig. 2.

For this example network, nonlinear and linear activation functions may be applied to the hidden layer and output layer, respectively. If both $g^{(1)}$ and $g^{(2)}$ are linear, the entire NN model is itself simply a linear function of $\boldsymbol{x}$. Therefore, nonlinear activation functions are crucial for approximating interesting functions.

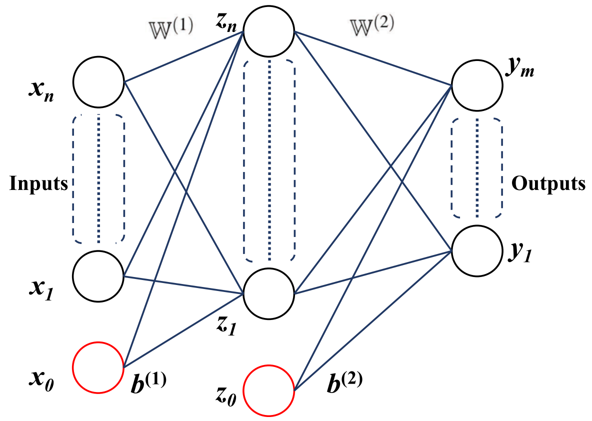


FIG. 2. Pictorial representation of the NN described in Eq. (1), with one hidden layer. The nodes depict the input variables $\boldsymbol{x}_i$ and hidden variables $\boldsymbol{z}_i$. The edges represent the matrices of weights $\mathbb{W}^{(1)}$ and $\mathbb{W}^{(2)}$, whereas the biases are represented by the weights from the additional variables $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$.

The term deep learning (DL) refers to NNs with at least one hidden layer—often, networks with multiple hidden layers are used. Choosing the structure of the NN, including the activation functions, is often done in an *ad hoc* trial and error fashion. As a result, NNs are often viewed as being black box opaque models, which are difficult for humans to interpret. In fact, the highly nonlinear layered structure of NNs is what makes them so flexible and powerful. Training NNs—the process of obtaining the optimal set of weights that solve a given problem and generalize well, for data not seen during training—can be achieved in multiple ways, the most commonly used of which is backpropagation and gradient descent.[39] To train NNs, we first have to define a loss function that, for supervised learning, measures how far the predictions of the network are from the measured data; a commonly used loss function is the mean squared error (MSE). In backpropagation, the gradient of the loss function can be computed efficiently for a given training example input–output pair, allowing for NNs to be trained using gradient descent—update the weights in the opposite direction to that of the gradient, in order to move toward the local minimum.[40]

There are many extensions to the simple NNs described above, designed to solve a range of specific problems. However, the basic structure and methodology for learning remain the same. One such example is the autoencoder, which can be either supervised or unsupervised. An unsupervised autoencoder learns an efficient encoding of unlabeled data, whereas a supervised autoencoder can be used to obtain the set of inputs that yields a desired output. An autoencoder consists of a FFNN with two parts: the encoder that learns to map the input data to an optimal representation and the decoder that learns to decode this representation and recover the initial data.[38] This structure is outlined in Fig. 3.

In optical fiber communication systems, there are a number of monitors that provide network operators with time series data, and hence, time series ML techniques are of particular interest. Recurrent NNs (RNNs) are a class of NNs that exhibit temporal dynamic behavior, meaning that they can be used to approximate functional

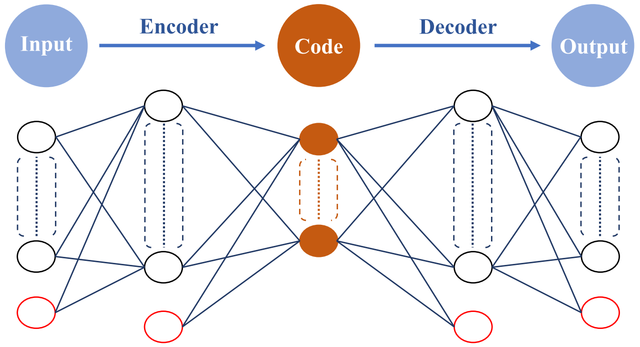


FIG. 3. Diagram outlining the structure of an autoencoder model, adapted from the work of Li *et al.*[96] and Goodfellow *et al.*[38] The input is fed into a NN, known as the encoder, that learns an internal representation or code. A second NN, the decoder, learns to map this code to the output.

relationships found in time series data.[41] This is achieved by considering the previous state of the network and the current input when determining the current state of the network. A schematic outlining the basic structure of a RNN is shown in Fig. 4. RNN models can maintain state information, allowing them to perform tasks such as traffic sequence prediction that are beyond the ability of a standard FFNN. However, RNNs are affected by gradient explode or gradient vanish problems[42] that prevent complete learning of the time series. Due to this issue, special cases of RNNs such as Gated Recurrent Units (GRUs)[43] and long-short term memory (LSTMs)[44] have been proposed that are capable of adaptively capturing dependencies on different time scales.

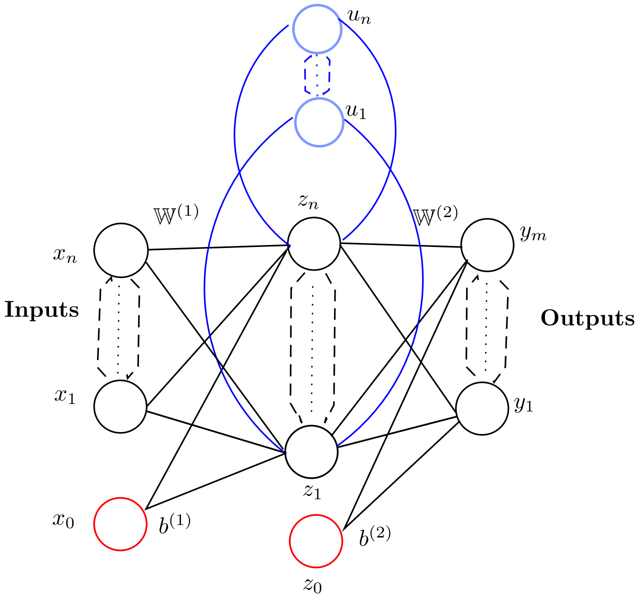


FIG. 4. An example RNN model architecture including context nodes $u_1, u_2, \ldots, u_n$ associated with each node in hidden layer vector z_t with fixed weights of one. Similarly to FFNNs [Eq. (2)], at each time step t, the input x_t is fed forward and a learning rule is applied. Additionally, the fixed back-connections save a copy of the previous values of the hidden nodes in the context nodes.[171]

Furthermore, as optical networks have a topological structure that can be represented by a graph, it is natural to utilize graph-based machine learning techniques, such as Graph NNs (GNNs) that leverage the network structure.[45] GNNs combine graph theory with NNs in a way that draws parallels with RNNs. There are two key sequential steps involved in updating a GNN for a given node: aggregation of the of the states of neighboring nodes, including the target node itself, followed by an update to the state of the node, depending on the specific analysis goal of the GNN.[46] Figure 5 describes an example GNN model for node-based prediction tasks. Based on the variations of the aggregation and update functions, several models of GNNs have been proposed in the recent literature, such as message-passing NNs,[47] graph convolutional networks (GCNs),[48] graph attention networks,[49] and gated graph NNs.[50] Examples of applications include classification and regression on nodes or edges, i.e., predicting classes or continuous values for these elements of a given graph. GNNs can also be supervised or unsupervised, providing some flexibility with regard to the application domain.

Another NN that has been used in network layer applications is the generative adversarial network (GAN).[51] GANs achieve their unique capabilities owing to their design based on zero-sum game theory. At a high level, they are composed of two NNs, the discriminator and the generator, which compete against each other. A schematic showing the structure of a GAN is shown in Fig. 6. GANs are designed for realistic data generation and have been successfully used for both image and video data generation in the recent literature. Thus, GANs show potential for traffic data generation in optical networks.

2. Gaussian processes

Gaussian processes (GPs) are a probabilistic ML approach in which the uncertainty associated with predictions is well-quantified.[52] This makes them attractive for optical fiber communication systems, in which the accepted failure rate is low and thus knowledge of the limitations of ML models is desirable. GPs can be used for regression or classification and are non-parametric methods,[53] meaning that no specific parametric form is assumed for the model but rather Bayes theorem is used to search the space of functions directly. In the context of GPs, the Bayes theorem can be stated as[52]

$$\text{posterior} \triangleq \frac{\text{prior} \times \text{likelihood}}{\text{marginal likelihood}}, \tag{3}$$

where the posterior is the predictive distribution we wish to obtain, the prior contains the information we know about the target function before we have seen the data, and the likelihood includes information from the measured data. In general, we wish to condition our prior on the measured data in order to obtain the predictive posterior distribution. Figure 7, adapted from the work of Rasmussen and Williams,[52] demonstrates a function drawn from an uninformative GP prior, which is then conditioned on data to produce an accurate model.

In general Bayesian inference, this involves numerical integration to calculate the required posterior. However, in GP regression, we assume that the likelihood function is a Gaussian, which means

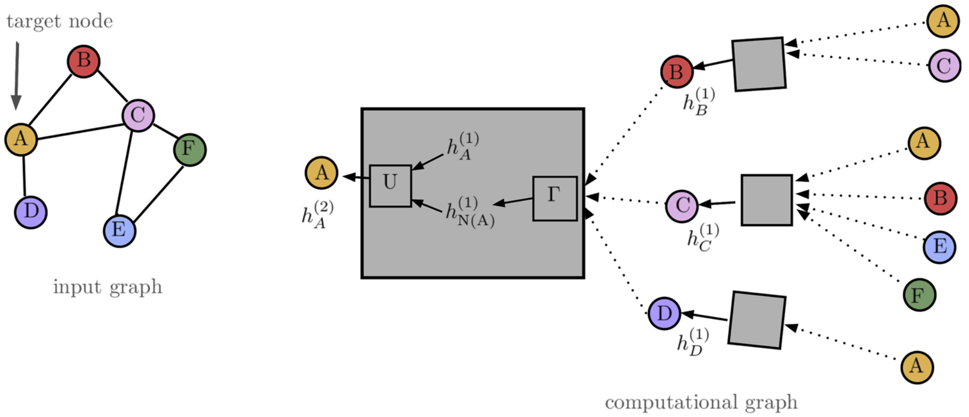


FIG. 5. An example architecture of a GNN for node-based predictions. The computational graph for target node A is shown on the right, where N(A) represents the neighborhood of node A, $h^{(1)}$ and $h^{(2)}$ represent hidden layers 1 and 2, respectively, and Γ and U represent the aggregation and update functions,[46] respectively. The complete GNN may comprise computational graphs for multiple nodes of interest.

that these integrals then become analytical and thus much less computationally expensive. This assumption is not valid for GP classification, however, making it more computationally demanding than GP regression models.

GPs are a kernel-based ML method, in which the kernel trick—the fact that it is more computationally efficient to work in the space of inner products than fixed coordinates—is leveraged.[26] As a result, the user must specify the kernel function at the design stage, which means making an assumption about the features we expect to see in the data. For instance, a commonly used kernel function is a squared exponential kernel plus a white Gaussian noise (GN) kernel, giving

$$k(x_i, x_j) = \nu \exp\left(-\frac{\|x_i - x_j\|^2}{2\mu}\right) + \Xi\left(\sigma^2\right), \tag{4}$$

where ν and μ are scalar hyperparameters controlling the absolute scale and the length scale of the target function, x_i and x_j are data points, $\|\cdot\|$ denotes the Euclidean distance operator, and $\Xi(\sigma^2) \sim \mathcal{N}\left(0, \sigma^2\right)$ if $i = j$ and 0 otherwise, where $\mathcal{N}\left(0, \sigma^2\right)$ denotes a zero-mean Gaussian distribution with variance σ^2. Choosing this kernel means assuming *a priori*, meaning before we have seen the data, that the function we are trying to learn has one length scale and white Gaussian noise. More complex kernels exist to describe features such as periodicity and decay, and one can design a kernel by noting that the sum of any two valid kernel functions is itself a valid kernel function.

GPs are trained by finding the optimal kernel hyperparameters via maximizing the log marginal likelihood in order to find the most likely interpretation of the data.[52] Once optimal hyperparameters are found, the predictive distribution of the GP can be calculated using Algorithm 2.1 of Rasmussen and Williams.[52] The predictive mean function and predictive variance of the GP can then be used to make probabilistic inferences about the data.

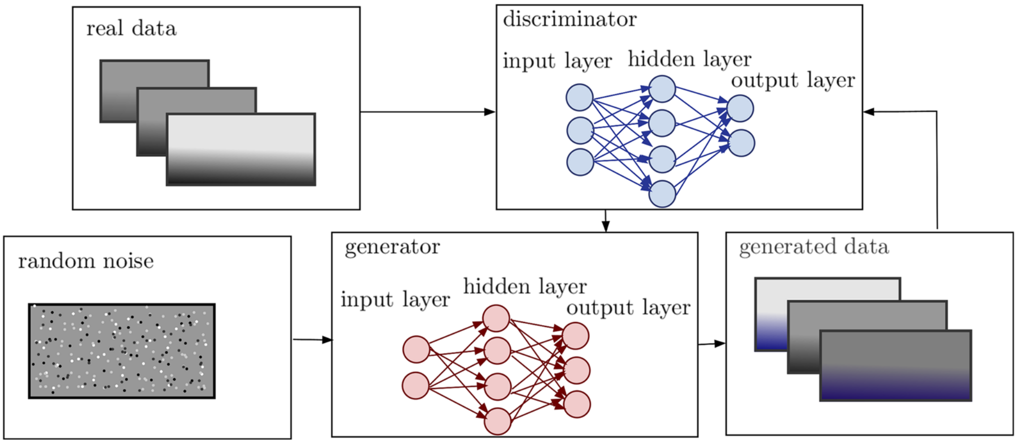


FIG. 6. An example architecture of a GAN model.

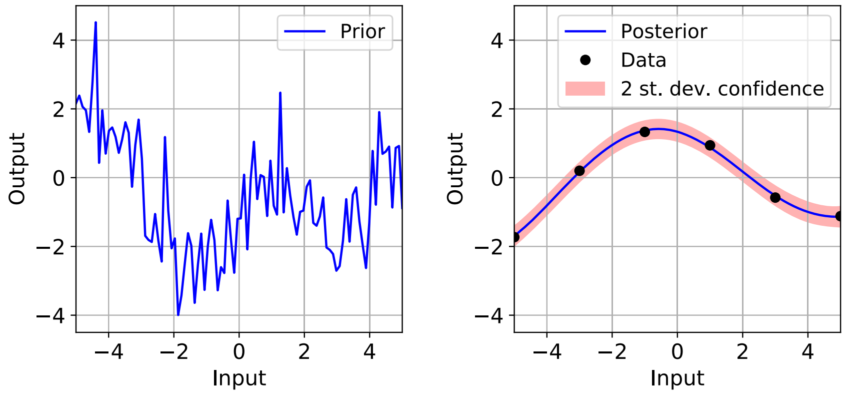


FIG. 7. Example adapted from Fig. 2.2 of the work of Rasmussen and Williams,[52] demonstrating how the GP prior, in this case chosen to be weak uninformative prior, is conditioned on data to produce a predictive posterior. (a) Function drawn at random from the GP prior. (b) The predictive posterior distribution after conditioning on the data. A confidence region is also shown, corresponding to two standard deviations or 95% confidence.

One of the major issues associated with using GPs is the computational complexity, which is $O(n^3)$, where n is the number of training examples. It is possible to use sparse approximations that reduce this computational burden,[54] at the cost of some accuracy.

3. Support vector machines

Another kernel-based ML method is the support vector machine (SVM), a method which can be used for supervised regression and classification[26] and for unsupervised learning.[55] However, the vast majority of SVM use within optical networking is for classification, and therefore, we focus on SVM classifiers here. Unlike standard GPs, SVMs are sparse kernel methods, meaning that the model predictions do not require evaluation of the kernel function for all training examples, but rather we only need to evaluate the kernel for a subset of the training data.

SVM classifiers work by constructing a decision boundary that separates the labeled data into distinct classes such that the margin, defined as the perpendicular distance between the closest data points in each of the classes and the decision boundary, is maximized. These points that are closest to the boundary are known as the support vectors, so-called because they directly specify the position of the boundary. Being the closest to the optimal boundary, these points are also the most difficult to classify. Figure 8 shows the example of a binary SVM classifier, with the decision boundary and support vectors highlighted.

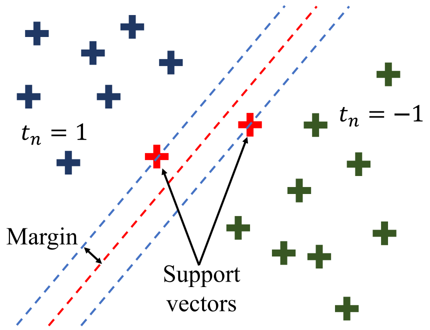


FIG. 8. Diagram showing the support vectors for a binary SVM classifier, where the data are labeled 1 or −1, adapted from Bishop Chapter 7[26]. The margin is also shown, which we maximize in order to find the most general decision boundary.

As a demonstrative example to provide intuition for SVMs, we follow Bishop[26] and consider the simple case of a binary classifier, with data labeled as one of the two classes, $t_n \in (-1, 1)$, modeled by a linear decision boundary model of the form

$$y(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b, \tag{5}$$

where $\boldsymbol{w}$ is a vector of weights, $\boldsymbol{x}$ is the vector of inputs, ϕ represents a fixed transformation in the input space, and b is a constant. A data point is classified depending on the sign of $y(\boldsymbol{x})$. It can be shown that, as the distance of the points $\boldsymbol{x}_n$ to the decision boundary is invariant under linear transformation, all data points satisfy the constraints

$$t_n\left(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b\right) \geq 1, \tag{6}$$

and the distance from point x_n to the decision boundary is given by

$$\frac{t_n\, y(x_n)}{\|\boldsymbol{w}\|}. \tag{7}$$

Thus, we find the decision boundary by solving the constrained optimization

$$\arg\min \|\boldsymbol{w}\|^2. \tag{8}$$

It can be shown that this is a quadratic programming problem, which can be tackled using Lagrange multipliers. Once the optimal decision boundary is found, new examples can be classified by their position in the input space relative to the boundary. This is an oversimplification of the SVMs used, in practice, but should give the reader some intuition for how an optimal decision boundary can be found. In practice, SVMs are formulated in terms of kernel space, as this allows us to keep the computational load reasonable by working in terms of inner products between the input variables. The kernel is defined in terms of the fixed transformation in Eq. (5) as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})\,\phi(\boldsymbol{x}'). \tag{9}$$

Moreover, the method described above finds a hard decision boundary, which only exists for linearly separable data. In general, SVMs are formulated to find a soft boundary, allowing for some degree of misclassification. Finally, SVMs are not limited to binary classification and can be constructed to facilitate multiple-output classes.

4. Reinforcement learning

RL is a discipline of ML that involves a learner known as the agent that learns interactively by taking actions in its environment, where the environment consists of everything outside of the agent.[56] The environment can be simulated or experimental; a simple example for the case of optical fiber communication networks could be the currently established light paths, current requests, and the SNR of these light paths.

Here, we outline the key concepts of RL, following Chap. 3 of Sutton and Barto.[27] The agent interacts with its environment at a series of time steps $t, t+1, t+2, \ldots$. At each time step t, the agent takes as input some representation of the state of the environment $S_t \in \mathcal{S}$ and chooses an action $A_t \in \mathcal{A}$, where $\mathcal{S}$ is the set of all possible states and $\mathcal{A}$ is the set of all actions that are possible given a state S_t, respectively. In the proceeding time step, the agent reaches a new state S_{t+1} and receives a reward $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$. The method by which the agent selects the action A_t given a state S_t is called the policy, denoted as Π_t, a mapping from states to probabilities of selecting each possible action. Informally, the goal of the agent is to maximize the cumulative reward received over time. A schematic showing how the agent interacts with its environment, adapted from the work of Sutton and Barto,[27] is shown in Fig. 9.

We also make a distinction between two types of agent–environment interaction: continuous tasks in which the number of time steps is infinite and episodic tasks, for which the interaction consists of a series of episodes each with a terminal time step. We denote this terminal step as T, and as RL has been applied to both continuous and episodic tasks in optical networks, we introduce a general notation that is valid for both types, in which a continuous task is represented by $T = \infty$. Thus, the agent aims to maximize the expected discounted return,

$$G_t = \sum_{\tau=0}^{T-t-1} \kappa^{\tau} R_{t+\tau+1}, \tag{10}$$

where $\kappa \in [0,1)$ is a parameter called the discount factor, which controls the value of future rewards at the present time step. If $\kappa = 0$, the agent will learn to maximize the immediate reward, whereas as κ approaches 1, the agent will strongly weight future rewards when choosing a policy. An important element of the RL framework is that we desire to have a state representation that conveys to the agent all relevant information about the environment such that the probability of entering a specific new state at $t+1$ can be defined only in terms of the state and action representations at t. In other words, we do not need the entire set of previous states and actions to find an optimal policy, but only the state and action at the previous time step. State representations that satisfy this are said to have the Markov property, and tasks that involve learning with a Markov state are called Markov decision processes (MDPs). For a finite MDP, meaning an MDP for which the state and action spaces are finite, we can completely determine the dynamics by the probability distribution

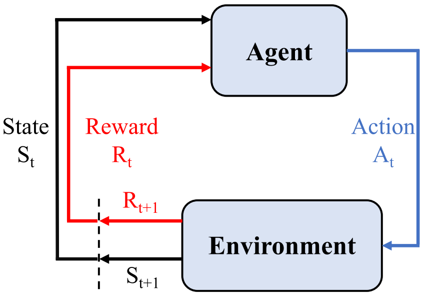


FIG. 9. Diagram showing the interaction between the RL agent and the environment. At time step t, the agent receives a state S_t from the environment and chooses to take an action A_t. In the proceeding time step, this yields a state S_{t+1} and returns a reward R_{t+1}. By iterating through this process, the agent learns to maximize the long-term reward.

$$p(s', r \mid s, a) = \mathbb{P}\left[S_{t+1} = s',\ R_{t+1} = r \mid S_t = s,\ A_t = a \right], \tag{11}$$

where s and a are a given state and action, s' is the new state, and r is the reward received. Here, it is assumed that $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $r \in \mathcal{R}$. Using Eq. (11), we can compute all other quantities needed by the RL agent.

In order to learn an optimal policy, RL algorithms attempt to estimate the value function, defined as the expected value of the cumulative reward obtained by starting in a state s and following policy Π,

$$v_{\Pi}(s) = \mathbb{E}_{\Pi}[G_t \mid S_t = s]. \tag{12}$$

Crucially, it can be shown that Eq. (12) follows a recursive relationship that has the form of a Bellman equation,[27,57]

$$v_{\Pi}(s) = \sum_{a} \Pi(a \mid s) \sum_{s',\,r} p(s',\, r \mid s,\, a)\left[r + \kappa\, v_{\Pi}(s') \right]. \tag{13}$$

This relationship allows the agent to compute an approximation to v_{Π}. The agent's goal of maximizing the long-term cumulative reward can be stated as finding the policy that has an optimal value function, and we can write a Bellman equation

$$\begin{aligned} v^{*}(s) &\triangleq \max_{\Pi} v_{\Pi}(s) \\ &= \max_{a \in A(s)} \sum_{s',\,r} p(s', r \mid s,\, a)\left[r + \kappa\, v^{*}(s') \right]. \end{aligned} \tag{14}$$

Here, v^* denotes the optimal value function, which may be achieved by more than one policy but will always exist for a finite MDP. In practice, the computational cost of computing v^* exactly is too high, and thus, we learn a suitably good approximation.

There are a number of different algorithms for finding Π^*, and these algorithms can be either model-based or model-free.[58] Model-based RL algorithms are concerned with computing an optimal policy for a MDP, assuming that a perfect model of the environment is available. Contrastingly, model-free algorithms do not rely on the assumption that such a model exists, but rather sample the MDP to obtain statistical knowledge about the unknown model. Such algorithms do not attempt to construct a model of the environment. Moreover, RL algorithms can be further categorized: for on-policy approaches, the agent will update its action-value function using the action determined by the current policy, whereas for off-policy approaches, a different policy is used to select the action.[27] Commonly, off-policy algorithms will utilize the ε-greedy policy, in which a threshold $\varepsilon \in [0,1] \subset \mathbb{R}$ is selected, and at each time step, a random real number is generated between 0 and 1. If the value of this number is greater than ε, the agent will perform the action that maximizes the expected cumulative reward; otherwise, it will perform a random action. This demonstrates the trade-off between exploration and exploitation that is crucial within RL—only exploiting current knowledge leads to short-sighted policies, but we need to refine successful policies to achieve high performance. Therefore, it is important to allow some degree of continuous exploration of the environment to achieve a policy that is optimal in the long-term.[59] One final distinction that will be encountered in the RL literature is that of value-based algorithms, in which the value function is parameterized in order to find an approximation to the optimal policy[60] and policy-based algorithms, where the policy is parameterized instead.[60] Finally, it is possible to combine these approaches by utilizing two learners, known as the actor and the critic. The actor learns the optimal action to take for a given state, and the critic learns to compute the value function of a given action.[27] Below, we summarize the specific RL algorithms used by works referenced in this Tutorial, highlighting useful references for the reader. The algorithms included in this section are within the scope of deep reinforcement learning (DRL), a sub-field of RL that has become of great interest in the recent literature owing to its successful adaptations in several application domains.[61] DRL relies on the intersection of reinforcement learning (RL) and deep learning (DL). In general, DRL algorithms incorporate DL to solve MDPs, often representing the policy or other learned functions as a NN.

Deep Q learning is a model-free value-based DRL algorithm that involves trying to find an optimal action-value function for a policy Π.[62] The key idea is to use a deep NN (DNN) to estimate the optimal action-value function,

$$q_{\Pi}(s, a; \mathbb{W}) \approx q^{*}_{\Pi}(s, a). \tag{15}$$

This method is best suited to solving RL problems with discrete low-dimensional action spaces.[63]

Asynchronous advantage actor-critic, or A3C, is another model-free DRL algorithm. In contrast to valued-based deep Q–learning, A3C is policy-based and the policy is parameterized by a NN in order to learn an approximation to the optimal policy,[60]

$$\Pi(a \,|\, s) \approx \Pi(s \,|\, a; \mathbb{W}). \tag{16}$$

The asynchronous aspect of A3C comes from the fact that multiple agents are trained in parallel on copies of the environment, providing asynchronous updates to the model weights. Crucially, this results in greater exploration of the space and hence improved performance over other algorithms such as deep Q learning for a number of tasks.[60]

Another commonly used RL algorithm is deep deterministic policy gradient (DDPG),[63] an extension of the deterministic policy gradient (DPG) algorithm[64] inspired by deep Q learning. The key idea behind DPG is to assume a deterministic policy, the gradient of which can be shown to follow the gradient of the action-value function $q(s, a)$. In DDPG, this is extended by using DNNs to parameterize the actor function and by employing some innovative techniques from deep Q learning and DL.[64] The resulting algorithm is effective for exploring continuous action spaces, addressing a shortcoming of deep Q learning.

III. PHYSICAL LAYER APPLICATIONS

In this section, we outline several key research problems within the physical layer and highlight selected applications of ML to these problems from the literature. Specifically, we discuss quality of transmission (QoT) estimation, digital twins, equalization in short reach applications, and fiber nonlinear noise mitigation in long-haul transmission systems. A summary of the works discussed detailing the physical layer applications tackled and different ML techniques proposed is given in Table I.

A. Quality of transmission estimation

One of the most widely researched applications of ML in optical fiber communications is QoT estimation, evidenced by a recent survey focusing on this application alone.[6] QoT is an umbrella term for a number of metrics of the quality of a transmitted optical communication signal, including SNR, bit error rate (BER), and Q-factor.[65] ML techniques are a logical approach to QoT estimation because of the numerous sources of uncertainty that make the estimation and prediction of QoT challenging[6] and the necessity of QoT estimation for performing network level control, such as for the routing and spectrum assignment of new light paths. A number of models of QoT exist that are based on the physics of transmission within the fiber, which have varying degrees of accuracy. Two commonly used examples are the Gaussian noise (GN) model[66] and split-step Fourier transform method (SSFTM).[67] However, these are plagued by limited applicability due to limited accuracy and high computational requirements, respectively. Moreover, both are limited by uncertainty in the physical layer inputs, with the magnitude of these uncertainties varying between deployed networks. For example, installed fibers can be accidentally damaged, before being spliced back together, resulting in variations in the fiber attenuation. Moreover, other components such as amplifiers and filters can suffer degradation in performance as they age,[68] which can change physical layer parameters such as EDFA noise figure. Additionally, parameters such as the fiber type and fiber chromatic dispersion (CD) may not be known to the operator in deployed networks.[69] ML can be used either as a replacement for physics-based models or alongside them in order to combat the input uncertainty and to reduce the computational burden.

The QoT estimation sub-domain can be further divided into three main problems. First, ML can be used to predict the QoT from physical layer inputs, such as the number of channels, operating wavelength, modulation format, and number of spans. This can be formulated as a regression problem, where the QoT itself is the target, or as a classification problem where the goal is to predict whether or not a given light path will have sufficient QoT. Second, ML can be deployed to aid with QoT monitoring, commonly to learn the mapping between the variables that are measured by using monitors and the QoT, often for the purpose of prediction of failures.[3,6] Finally, the modeling of the optical amplifiers used in optical fiber communications presents a challenge due to the nonlinear dependence of amplifier gain on wavelength, channel launch power, and the number of channels. As amplifiers can have a significant effect on the QoT, there have been a number of works in which ML has been applied to modeling amplifiers.[3,5,6]

Here, we outline selected examples of ML applied to QoT estimation from the literature that demonstrate what is typical in

TABLE I. ML approaches to physical layer applications.

Application	ML technique(s)	Advantages	References
QoT estimation	Simple learning process, LMA	Interpretable	72, 73, 75, and 87
	GP	Well-quantified uncertainty	76
	CBR, NN	Experimental demonstration	77–79
	GP, NN	Physics-informed ML, less data required, and explainable	86 and 85
	NN, SVM	Self-adaptive and reduced computational complexity	81 and 83
Digital twins for optical networks	RNN, DRL, XGBoost	Experimental data and general framework	41, 92, 94, 91, and 95
Short reach equalization	DNN	Outperform conventional equalizers	103 and 104
	CNN	Outperforms DNN	106 and 107
	RNN, LSTM	Improved performance compared to FFNN via feedback	108–112
	SVM	Unsupervised and enable decoding of PAM-N signals	113 and 114
	DNN, RNN	Low complexity FPGA implementation	105 and 111
Fiber nonlinear noise mitigation	NN, ELM	Reduced computational complexity	119 and 128
	LSTM	Better performance than six-step DBP	120
	SVM, KNN, PW	Increased optimal launch power	121, 122, and 124
	K-means clustering	Found required overhead for transmission	123
	NN	Physics-informed ML and explainable	129, 130, and 131
	NN, transfer learning	Increased flexibility and reduced computational load	137
	K-means clustering	Low complexity FPGA implementation	127

the field. Interesting works using regression include using a simple learning process based on gradient descent[40] to reduce the uncertainty in the inputs to a physics-based QoT model.[70] This represents a hybrid approach where ML is used in concert with physical models of the QoT, rather than relying solely on the data. A similar approach was also demonstrated experimentally—a learning process was used to update the parameters of a physical model based on measurements of the Q-factor of an experimental system.[71] Additionally, ML based on the Levenberg–Marquardt algorithm[72] (LMA) was recently utilized for online optimization of the inputs to the GN model, specifically the launch powers, for a simulated network.[73] Interestingly, the number of iterations used for the optimization is adaptive, which reduces the time and measurement resources required to perform the optimization. Again, the role of ML here is to configure the inputs to the physical model, rather than replacing it. There are also approaches in which the goal is to replace the physical model. For example, a GP regression model has been used to learn the functional relationship between the BER and system transmission parameters, specifically the launch power, length of fiber over which the signal is transmitted, symbol rate, and channel spacing.[74] This model was trained on both simulated and experimental data, and it was shown that the model could make accurate predictions on a system with a different configuration to that upon which it was trained. As many of the QoT estimation works utilize NNs, this Tutorial highlights that more principled approaches such as GPs with well-quantified predictive uncertainty can also be used successfully for QoT estimation. Moreover, an experimental network has been operated at a reduced margin via a case-based reasoning (CBR) approach,[75] where margin means the difference between the minimum acceptable QoT and the current signal QoT. In CBR, the QoT for established light paths is stored and used as a lookup table to estimate the QoT of new light paths that take a similar route through the network. This work is particularly interesting as it demonstrates that ML, albeit a simple version of it, can be useful for controlling an experimental optical fiber network—in this case, it allows us to reduce the required margin. Another recent experimental demonstration of the efficacy of ML-based QoT estimation utilized NNs trained on synthetic QoT data to estimate the SNR on a live network operated by Tele2 Estonia.[76] Crucially, these models demonstrated a maximum SNR error of 0.5 dB and were able to compute the SNR estimate on microsecond scale, indicating that such models could feasibly be deployed in real networks. DNNs have also been used recently to estimate the SNR based on historical telemetry of the optical amplifiers in an experimental system, focusing on the effect of the amplifiers, rather than the nonlinear noise generated by transmission in a fiber, which they assume can be estimated using a physical model.[77] Moreover, a NN-driven nonlinear SNR estimator was presented, for which the optimal combination of input features was found.[78] In this work, knowledge of the physics of fiber transmission is used to aid with feature engineering, in order to obtain the set of input features with the highest efficacy.

Classifiers have also been leveraged for QoT estimation, such as a binary NN classifier trained on historical network data, which was used to determine whether or not a given request will have sufficient QoT to be established.[79] The performance of this classifier was compared to that of an analytical QoT model[80] and was found to efficiently replace this model, while providing a key benefit of self-adaptivity to changes in the network conditions. Another work[81] utilized an SVM classifier model, again as a binary classifier designed to label light paths as having sufficiently high QoT to be established or not. Simulated data are used for training, as is common in network-scale research due to the lack of availability of detailed datasets from deployed networks.

Furthermore, an interesting research avenue within QoT estimation is the use of physics-based models in concert with ML. This can be done in a number of ways, for instance, our physical models can be embedded into the ML directly. For example, a methodology for the training of NNs that obey physical laws defined by partial differential equations was recently presented.[82] The first steps toward using this in optical fiber communications have been taken, where a physics-informed NN was used to solve the nonlinear Schrödinger equation (NLSE) in an optical fiber and model pulse evolution.[83] An alternative approach is the physics-informed GP regression method, in which a physical model, in this case the SSFTM, is embedded within the GP.[84] This allows one to train GPs with fewer measurements of the system and represents an explainable ML approach with a well-quantified prediction uncertainty. Additionally, there are works such as those described above,[70,71] which focus on learning more accurate inputs to a physics-based QoT model. A similar approach has been applied to nonlinearity estimation.[85] Specifically, ML is utilized to reduce physical model errors and to combine modeling and monitoring schemes for nonlinearity estimation. Moreover, it is possible to use our knowledge of system physics to improve ML in other ways, such as to engineer higher performing input features.[78]

Finally, a recent paper[86] highlights the remaining roadblocks that stand in the way of effective deployment of ML in QoT estimation. Specifically, due to competition-related concerns, telecommunication companies are not willing to give external researchers access to real network datasets, resulting in a reliance on simulated data, or data that are produced using a lab setup. Due to the limitations of physics-based models outlined above and the fact that a lab-based network is always going to be more idealized than a deployed network, such data may not be fully representative of deployed network data. As a result, the true efficacy of ML approaches for deployed networks is unknown. Moreover, many of the applications of ML in optical networks utilize error metrics that are standard in ML but may not be suited to optical networks. For example, it has been found that for optical network applications of ML, using only the mean squared error may result in an inflated measure of model efficacy and novel error metrics have been recently proposed to address this.[87] Thus, although the first problem is tricky to address and is largely up to network operators, the second problem provides an interesting avenue for further research.

B. Digital twins

Digital twins are models that act as a virtual copy or "twin" of a real system. They are inherently data-driven,[88] taking as input measurements from the real system to build up a model of its governing physical laws, states, and behavior. Information drawn from the digital twins can then be passed to the real system in the form of changes to its operational configuration. This framework is outlined in Fig. 10. As we move toward higher levels of automation in optical communication network design and operation, digital twins are gaining increasing popularity within the research community.[89] It is hoped that digital twins can help bridge the gap between the ideal physical layer that is commonly assumed in optical communications and physical layer behavior in deployed networks, which is far from ideal. Although ML is not a required component of digital twins, due to their data-driven nature it is natural that ML approaches can be useful for creating digital twin models. ML can be used as the basis for the digital twin itself—we can take measurements from the real network and train a sufficiently complex ML algorithm to emulate the behavior of the network. Alternatively, we can build the digital twin from physics-based models and utilize ML to reduce the gap between these models and reality. For example, we can use ML to reduce the uncertainty in the model inputs, as discussed in Sec. III A. Additionally, ML can also be used in order to extract more information from network monitors, which may allow for the development of more detailed digital twins.

A framework for applying digital twins in optical networks has recently been proposed,[89] focusing on three crucial applications: fault prediction, hardware configuration, and simulation of transmission. Different ML approaches from the literature are proposed for each of these applications. For fault prediction and diagnosis, two models are proposed, a RNN to extract the operating state from time series data taken from monitors[90] and an XGBoost[91] model to map information from network monitors to new features to aid with fault diagnosis.[92] Moreover, DRL is proposed to learn an optimal strategy for hardware optimization.[93] Specifically, the agent learns to control the configuration of the programmable optical transceiver in order to maximize the QoT for varying operating conditions. Finally, a RNN-based approach is proposed to learn a model of the physical layer transmission in the network as a function of time series monitoring data.[94] Thus, the digital twin is created by combining these models, continually updating them with new data and using them to control the network.[89] Another recent work demonstrated a digital twin model based on an autoencoder, which is trained on an open-source dataset of power spectral density (PSD) profiles before and after transmission through an experimental optical network.[95,96] Specifically, this model is used to find the input PSD that produces

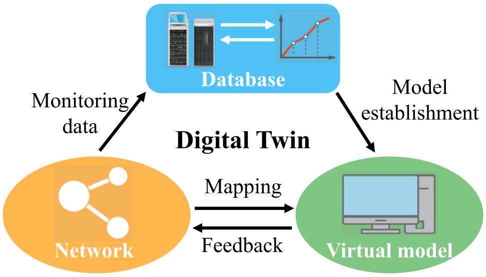


FIG. 10. Schematic showing the digital twin framework adapted from the work of Wang *et al.*[89] Monitoring data from the physical network is stored in a database, and useful information is extracted from these data from which a virtual model is built. This model is used to provide feedback to the physical network, while any changes to the network state are mapped back to the virtual model.

a desired output PSD. Thus, this model can be used as part of a digital twin to achieve optimal control of the network. It should also be noted that PSD may be a less widely understandable QoT metric and that methods to obtain the optical signal-to-noise ratio (OSNR) from PSD data have been proposed that could be used to convert these data, either before or after training the autoencoder. Other works have successfully utilized autoencoders for end-to-end learning of an intensity modulation direct-detection (IM/DD) optical communication system, outperforming conventional signal processing techniques.[97] This has been recently extended to include optimization of the symbol distribution for coherent systems.[98] Such techniques may be of use in the development of digital twins as they constitute an end-to-end virtual model of the system with inherent mapping and feedback between the virtual model and the physical system.

C. Equalization in short reach applications

Optical short reach systems, defined as having a length less than 100 km, are applied in server-to-server, intra-data center, inter-data center, access, and metro links. Due to stringent requirements of low complexity and cost, minimal power consumption, and small carbon footprint, IM combined with DD with simple on–off keying (OOK) or pulse amplitude modulation (PAM)-4 modulation format is still a preferable transceiver technology compared to coherent systems.[99]

Increasing demand for high data rate short reach applications such as IM/DD based systems causes several performance limiting factors that need to be addressed. A schematic of a short reach link with possible sources of linear and nonlinear impairments is shown in Fig. 11. First, chromatic dispersion (CD) severely limits the link power budget margin. With a high symbol rate and several kilometers of transmission, the interaction of CD and DD causes a power-fading effect and the detected signal may contain frequency notches. DD is based on square law detection, which complicates the CD equalization, as we cannot simply multiply the received signal spectrum with the inverse of the CD transfer function as in coherent systems. Another common impairment in short reach systems is considerable low-pass filtering effects due to the insufficient bandwidth of various components, which can cause severe inter-symbol interference. Furthermore, as short reach systems often have constrained financial budgets, low-cost components produce non-idealities, resulting in performance degradation. Similarly, low-cost devices such as lasers, modulators, photodiodes, and trans-impedance amplifiers also produce nonlinear distortions, such as level-dependent skew and level-dependent noise.[100]

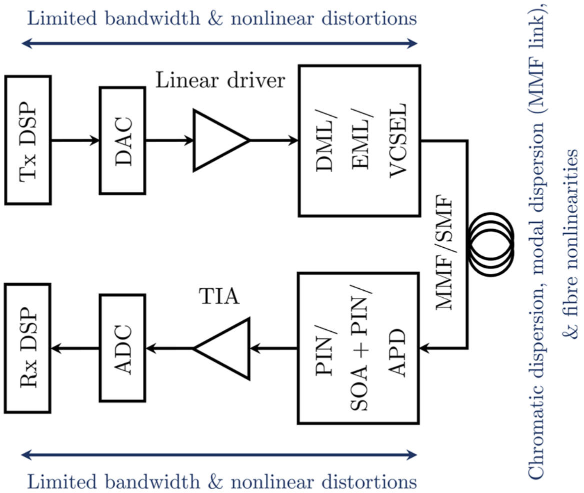

Limited bandwidth & nonlinear distortions
Tx DSP
DAC
Linear driver
DML/ EML/ VCSEL
MMF/SMF
Rx DSP
ADC
TIA
PIN/ SOA + PIN/ APD
Limited bandwidth & nonlinear distortions
Chromatic dispersion, modal dispersion (MMF link), & fibre nonlinearities

FIG. 11. Schematic showing a IM/DD-based short reach link and possible sources of impairments (DSP: digital signal processing, DAC: digital to analog converter, DML: directly modulated laser, EML: electro-absorption modulator, VCSEL: vertical cavity surface emitting laser, MMF: multi-mode fiber, SMF: single-mode fiber, SOA: semiconductor optical amplifier, APD: avalanche photodiode, TIA: transimpedence amplifier, and ADC: analog-to-digital converter).

For equalization of linear impairments, a feed-forward equalizer (FFE), usually based on a finite impulse response filter, is commonly used. The effect of frequency notches cannot be mitigated by using a FFE, although a decision feedback equalizer (DFE) can be added after a FFE to combat such an effect. However, DFEs may suffer from error propagation and instability due to the decision feedback scheme. Moreover, FFEs/DFEs cannot mitigate the nonlinear effects. Volterra nonlinear equalizers are an effective way to mitigate both fiber nonlinearity and component nonlinearities.[101] However, the major drawback of this equalizer is the large implementation and computational complexity.

Recently, ML techniques attracted significant attention for equalization of short reach systems. Among different ML-based techniques, NN-based equalization is in the center of this interest. A sufficiently large NN having at least one hidden layer can approximate any function and thus can be used as an equalizer of both linear and nonlinear impairments. Usually, the input vector of the equalizer corresponds to a set of consecutive sampled symbols. The length of vector should be long enough to consider the channel memory. The NN can be structured with a single hidden layer and large number of nodes or multiple hidden layers (i.e., a DNN) with relatively fewer nodes. The choice of nonlinear activation function in each hidden layer is important as it enables approximation of nonlinear functions to deal with the distortion of short reach systems. The commonly used hidden layer activation functions are the sigmoid function, the rectified linear unit (ReLU), and the hyperbolic tangent (tanh) function. On the other hand, the Softmax activation function is usually chosen for the output layer, as this function facilitates making symbol decisions for any PAM-N signal in addition to the equalization.[102] In several experimental demonstrations, it has been shown that NN-based equalizers outperform conventional equalizers, such as FFE and Volterra nonlinear equalizers.[102,103] In addition, a field programmable gate array (FPGA) implementation of a fixed point DNN-based equalizer was demonstrated for high-speed passive optical networks.[104]

The CNN-based equalizer was also investigated by Li *et al.*[105] As the convolution layer acts as a multi-channel nonlinear learned local pattern detector, it allows the equalizer to overcome the inter-symbol interference and device nonlinearity. In CNN-based nonlinear compensation, the time series input signal is converted to a 1D input array with N elements comprising $(N-1)/2$ past and post-symbols, followed by the multiple convolutional layers and fully connected layers with a nonlinear activation function. Experimental demonstrations showed that the CNN-based approach yields a considerable performance improvement as compared to a DNN-based approach.[105,106]

NNs also exhibit powerful equalization capabilities compared to feed-forward Multilayer perceptron (MLP) or CNNs, as they can use the feedback of past output values as an additional input while calculating the present output value.[107,108] With such additional feedback information, RNNs perform better than FFNNs, which is analogous to the performance improvement given by the combination of FFE and DFE compared to FFE only. Auto-regressive RNN and layer RNN are two commonly used types of RNN, and the former has better equalization performance.[109] An RNN-based equalizer with parallel outputs was investigated using an FPGA implementation for 100 Gb/s passive optical network application.[110] As a variant of RNNs, LSTMs were also demonstrated for the equalization of a 50 Gb/s PAM-4 transmission system.[111]

In addition to various NN-based equalizers, SVM-based approaches have been demonstrated as an effective tool for mitigation of nonlinear impairments in a short reach application scenario.[112,113]

The computational complexity of the nonlinear equalizer is a critical issue for short reach in optical communications because the equalizer needs to be implemented in real-time operating at an extremely high symbol rate. It has been shown that a NN-based equalizer with a single hidden layer can provide better performance with lower computational complexity compared to a Volterra equalizer.[114,115] However, a comprehensive analysis of computational complexity and performance for various advanced ML-based equalization approaches is required. In addition, the techniques for reduction in complexity need to be explored. Given that there is significant potential for practical NN-based equalizers to be implemented on digital signal processing (DSP) ASICs, ML-based equalization may become the mainstream technology for next generation short reach IM/DD-based systems.

D. Fiber nonlinear noise mitigation in long-haul transmission systems

In long-haul fiber transmission systems, the optical signal suffers from fiber nonlinear noise-like distortions due to the optical Kerr effect. Generally, the following system of coupled NLSEs is used to describe the evolution of complex-valued envelopes of the electrical field in the optical fiber:[116]

$$\begin{cases} i\,\dfrac{\partial E_x(z,t)}{\partial z} = -i\,\dfrac{\alpha}{2}\,E_x(z,t) + \dfrac{\beta_2}{2}\,\dfrac{\partial^2 E_x(z,t)}{\partial t^2} \\ \qquad\qquad - \dfrac{8}{9}\gamma\left[|E_x(z,t)|^2 + |E_y(z,t)|^2\right]E_x(z,t), \\ i\,\dfrac{\partial E_y(z,t)}{\partial z} = -i\,\dfrac{\alpha}{2}\,E_y(z,t) + \dfrac{\beta_2}{2}\,\dfrac{\partial^2 E_y(z,t)}{\partial t^2} \\ \qquad\qquad - \dfrac{8}{9}\gamma\left[|E_x(z,t)|^2 + |E_y(z,t)|^2\right]E_y(z,t), \end{cases} \tag{17}$$

where $\boldsymbol{E}(z,t) = [E_x(z,t),\ E_y(z,t)]^{\mathrm{T}}$ is the Jones vector, α is the fiber loss coefficient, β_2 is the fiber group velocity dispersion coefficient, and γ denotes the fiber nonlinear coefficient.

Although the SSFTM can be used to numerically solve the NLSE, the accuracy is low when the interplay among signal, noise, nonlinearity, and dispersion effects is considered. Therefore, the performance improvement of the conventional digital backpropagation (DBP) method based on the NLSE is limited.[117] Since the performance improvement is related to the modeling accuracy, ML techniques can be applied to describe the evolution of the optical signal after long-haul transmission. Specifically, ML techniques are applied to find a nonlinear function $\boldsymbol{f}$ that can map the received symbol to the transmitted symbol under certain criteria.

Unlike in short reach transmission scenarios, the nonlinear function $\boldsymbol{f}$ has to be obtained by separating the I and Q branches of the complex-valued signal. In the early works,[118] an artificial neural network (ANN) has been used in a coherent receiver after CD compensation with extreme learning machine (ELM)-based training techniques. The simulation results for 27.59 GBd/s return-to-zero (RZ) quadrature phase shift keying (QPSK) show that the ELM-based technique can provide similar performance to conventional DBP with much lower computational complexity after 2000 km standard single-mode fiber (SSMF) transmission. Recently, LSTMs have been proposed to mitigate the fiber nonlinear impairments in dual polarization WDM transmission systems. It was shown in simulation that LSTMs can provide better performance than conventional DBP techniques with six steps per span.[119]

It is known that the nonlinear noise is non-Gaussian distributed. Therefore, conventional linear boundaries are not effective in the nonlinear fiber channels. One general idea of ML-based coherent receivers is to design nonlinear decision boundaries. These are assumed to be more suitable for the nonlinear fiber channel because the nonlinear noise generated in the fiber channel need not be a Gaussian distribution. A few techniques have been applied to design such nonlinear classifiers. An M-ary SVM has been introduced to mitigate the nonlinear phase noise in the single-channel single-polarization (SCSP) 16-QAM coherent optical systems. Compared with the linear channel equalization case, the simulation results show that M-ary SVMs can increase the optimal launch power by around 4 dB and extend the transmission distance by around 1200 km.[120] The K-nearest neighbor (KNN) algorithm has also been utilized to mitigate the channel impairments, including the laser phase noise and nonlinear fiber noise. The simulation results show that the optimal launch power can be enhanced by ~0.4 dB in the SCSP 16-QAM coherent transmission system.[121] Another work using K-means clustering[122] experimentally investigated the requirements of the length of the training symbols for the fiber nonlinear mitigation in the SCSP 64-QAM 80-km transmission scheme. It was observed that a 10% training overhead is sufficient to obtain the optimal performance. Another recent publication utilizing nonlinear classification is based on the Parzen window (PW) classifier technique, which is inherently a multi-class technique and can be implemented in online learning mode.[123] Considering the DBP technique as a benchmark, simulation results prove that a PW classifier can further improve the performance by ~0.35 and ~0.2 dB for 16-QAM after 1600 km and 64-QAM after 480-km fiber transmission. A density-based spatial clustering of applications with noise algorithm was employed for blind fiber nonlinearity compensation.[124] The experimental result showed that this algorithm can provide up to 0.83 and 8.84 dB enhancement in the Q-factor when compared to conventional k-means clustering and linear equalization, respectively, in a 40 Gb/s 16-QAM system after 50-km SSMF transmission. A histogram-based clustering algorithm was also demonstrated in a coherent optical long reach passive optical network, which achieves a Q-factor 0.57 dB higher than that achieved using maximum likelihood and 0.21 dB higher than that obtained using k-means clustering.[125] In another recent work, an FPGA-based real-time fiber nonlinearity compensator using the sparse K-means++ clustering algorithm was experimentally demonstrated in a 40 Gb/s 16-QAM self-coherent optical system. This resulted in a 3 dB improvement in the Q-factor compared to linear equalization at 50-km transmission.[126] More recently, a DNN-based nonlinear classifier with a cross-entropy cost function was used as a soft-demapper for soft-decision forward error correction (FEC).[127] In optical coherent 92 GBd dual polarization 64-QAM 950 Gb/s back-to-back measurements, the DNN-based nonlinear classifier is shown to have better performance than pruned Volterra nonlinear equalizers by 0.35 dB in OSNR with equal complexity or achieve the similar performance with 65% less computational complexity.

The above ML techniques in optical communications are operated as a black box to obtain the data-driven models with unparalleled performance. Therefore, some works have tried to contribute more insights into how the nonlinear fiber noise is mitigated by the ML techniques. Recently, the structure of a NN is designed to be similar to the DBP structure, which is called a learned DBP algorithm.[128] It is known that the conventional DBP algorithm is a cascade of linear filters D^{-1} for CD compensation and nonlinear operations N^{-1} for nonlinear phase derotation, as shown in Fig. 12(a). Each linear filter D^{-1} is given by the frequency-domain transfer function $H_k(\omega) = \exp\left[-\left(\alpha + i\,\omega^2\beta_2\right) L_k/2\right]$, where L_k is the length of the kth span. The nonlinear operation N^{-1} for the kth span is given by $\delta_k(x) = x\,\exp\left(-\,i\,\gamma\,\xi_k\,|x|^2\right)$, where ξ_k is a scaling factor. It should be noted that practical implementation of the linear filter D^{-1} is realized based on a time-domain finite impulse response filter and the filter coefficients are adjusted during training of the NN. Therefore, the interleaving linear and nonlinear processing in DBP can be regarded as the linear and nonlinear operations in the multi-layer NN, as shown in Fig. 12(b), where the input is the received samples and the output is the estimated symbol sequence. In this case, the parameters ξ_k and the filter coefficients of D^{-1} can all be optimized via ML techniques. An experimental demonstration is also conducted to evaluate the effectiveness of the learned DBP algorithm in a DP 5 channel WDM transmission system considering other channel impairments in a coherent transmission system, including frequency offset and laser phase noise.[129] The experimental results show that 1-steps per span and 2-steps per span learned DBP provide an additional gain of 0.25 and 0.45 dB over conventional 50-steps per span DBP and a total gain of 0.85 and 1 dB over linear equalization, respectively. It is also shown that learned DBP can give an insight into how and what the NN learns, which may guide people to analyze the interplay between CD, nonlinearity, and noise more closely. As of the complexity, it is shown that the performance of learned DBP based on 1 step per span is better than conventional DBP with 50 steps per span.[130] Note that the performance improvements of learned DBP originate from optimizing the parameters in DBP, and it incurs no additional computational complexity.

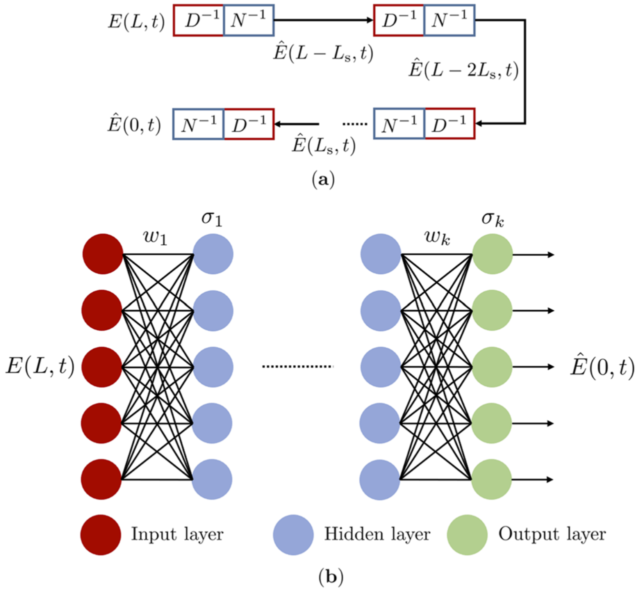


FIG. 12. (a) Classical DBP structure with interleaving operations of CD compensation and nonlinear-phase derotation. (b) DBP structure as an ANN with interleaving linear and nonlinear operations.

In another method, perturbation terms are used to analyze the fiber nonlinear terms, which can be expressed as[130]

$$\begin{bmatrix} E_x(z,0) \\ E_y(z,0) \end{bmatrix} = \sum_{m,n} P_0^{3/2}\left(H_n H_{m+n}^* H_m + V_n V_{m+n}^* V_m\right) C_{m,n}, \qquad (18)$$

where $P_0, H_m, V_m, C_{m,n}$ are the optical power, sample sequences for x and y polarization, and the perturbation coefficients, respectively. In the conventional method,[131] the perturbation coefficients $C_{m,n}$ can be analytically computed, given the link parameters and signal pulse duration/shaping factors. Alternatively, the perturbation coefficients $C_{m,n}$ can be obtained via a two-layer NN, which can describe the model with higher accuracy by taking into account higher-order nonlinearities. In a single-channel 32 GBd DP-16QAM transmission system, ~0.6 dB Q-factor improvement is observed after 2800-km SSMF transmission when the transmitted symbols are pre-distorted based on the estimated perturbation coefficients via NN.

ML-based compensation for multicarrier modulation formats has also been investigated. For the orthogonal frequency-division multiplexing (OFDM) format, an ANN was proposed, which provides 2 dB Q-factor improvement for the 40 Gb/s 16-QAM signal after 2000 km fiber link.[132] This improvement increased to 4 dB at the data rate of 70 Gb/s. A multiple-input and multiple-output-DNN-based nonlinear dispersion compensator was also demonstrated for the 40 Gb/s coherent OFDM system that achieved significant power margin improvement over both a conventional linear equalizer and a single-input single output DNN.[133] Considering the same experimental setup, support vector regression shows 1 dB Q-factor improvement over the full-field DBP method for 40 Gb/s 16-QAM OFDM over 2000 km SSMF transmission.[134] In a further work, a Newton-based SVM method that requires significantly less computational load than a conventional SVM was proposed to extend the optimum launched optical power by 2 dB compared to the Volterra-based nonlinear equalizer.[135] Finally, we consider the issue of flexibility in NN-based nonlinear channel equalizers. A general question concerning flexibility is whether we need to repeat the training process when the channel conditions (modulation format, launch power, transmission distance, etc.) are changed. In order to solve this issue, transfer learning has been proposed recently to reuse some parameters from the NN model trained for the previous system to build a new NN model that fits the modified system with a smaller amount of training resources.[136] The simulation results indicate that the number of epochs or size of the training dataset can be reduced by up to 99% when transfer learning is used. Therefore, a fast re-configurable nonlinear equalizer is possible for the practical implementation of optical networks.

IV. NETWORK LAYER APPLICATIONS

In this section, we describe crucial research domains within the network layer and highlight selected ML approaches to tackling the problems in these domains from the literature. Namely, these domains are network traffic prediction and generation and core network parameter optimization. As detailed below, we find that supervised learning approaches have been successfully deployed in the former domain, whereas RL approaches have shown great potential in the latter. Table II summarizes these applications, highlighting the advantages of the particular ML methods employed.

A. Network traffic prediction and generation

In the state-of-the-art optical networks, traffic is typically represented by demands.[137–139] The optical network operates based on a time scale and can be divided into time steps or iterations. In particular, in each time step/iteration, a number of demands arrive to the network, some of which are established. Every demand can be described by the time step in which it appears, a source node that represents the demand initial node and a destination node that represents the demand final node, demand volume, and holding time.[137] In a real-time flexible networking scenario such as elastic optical networks (EONs), where the network can adapt to accommodate the incoming traffic,[140] ML techniques coupled with dynamic routing algorithms can improve the overall network performance significantly.[141] One of the key challenges in increasing the efficiency of network operation is to predict the bandwidth requirement in the next time step based on the measurement of traffic characteristics in real time. When using ML methods, the goal is to forecast future traffic rate variations as precisely as possible based on the measured history.

NN-based approaches are the most commonly used ML technique in the literature of traffic prediction,[141–145] with early research utilizing standard ANNs.[141] Following this, later research used different variations of NNs.[142–145] Moreover, others employed NNs with an improved optimizer such as Zhan *et al.*,[146] who utilized a NN model optimized by the adaptive artificial fish swarm algorithm to predict tidal traffic.

Variations of NN approaches appearing in the state-of-the-art of traffic prediction include RNNs, such as Gated Recurrent Units (GRU) and LSTM owing to their capability of adaptively capturing dependencies on different time scales (see Sec. II). GRU is studied to make predictions of traffic matrices for a fixed-grid WDM network[142] and for a backbone EON.[143] LSTM is studied for traffic prediction in passive optical networks[144] and for core networks.[137] Figure 13 describes an example of a traffic prediction model based on GRU.

Another recent type of NN studied in the traffic prediction literature is GNNs. In the context of network topology based traffic data, the ability of GNNs to leverage a graphical representation to learn inter-node dependencies of the network graph shows strong potential for applicability in this domain. Gui *et al.*[143] studied the

TABLE II. ML approaches to network layer applications.

Application	ML technique(s)	Advantages	Reference(s)
Traffic prediction and generation	FFNN	Adaptive method and improved resource utilization	147 and 142
	RNN (GRU, LSTM)	Captures temporal aspects and more capacity available	143, 145, and 146
	GP	Improved efficiency and reduced traffic disruption	140 and 148
	SVM, DT, RF, LDA	Classification	138 and 139
	GNN	Captures graph structure	144
	GAN	Ability to generate realistic data	149
Core network parameter optimization	RL	Handles dynamic traffic request	161 and 162
	GNN	Leverages network structure and topology invariance	163 and 164

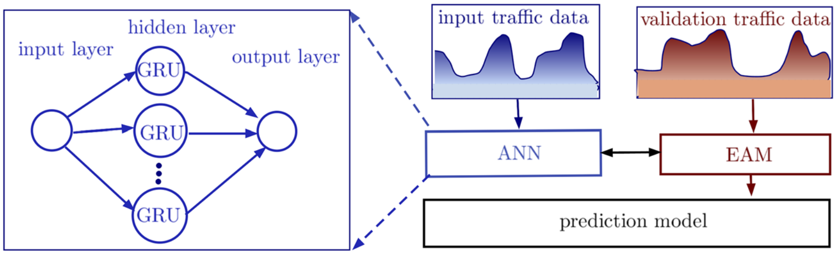


FIG. 13. An example traffic prediction model based on GRUs.[142] The Evaluation Automation Module (EAM) consists of the prediction error for both training data and validation data at each epoch and stores the best prediction model.

pair-wise spatial correlations between optical network nodes using a directed graph. The nodes of this graph represent switch traffic, and the weights of edges denote connections among optical network nodes. A GCN was then employed to leverage these spatial correlations. Vinchoff *et al.*[145] employed GCNs and GANs for prediction of traffic bursts in the optical network. Three types of burst events were modeled, namely, plateau, single-burst, and double-burst, representing steady traffic, a rapid traffic spike followed by a steady decrease, and a rapid traffic spike followed by an unexpected greater traffic spike, respectively.

Another ML approach that has been successfully applied to traffic prediction is GPs. The ability of GPs to capture temporal aspects of traffic flows allows both the short term and long-term prediction of input traffic. Studies have shown agile management of resources in a core optical network using GP-based traffic prediction.[139,147]

Recent comparative studies[137,138,143] on traffic prediction highlight the relative strengths of different ML methods used in the state of the art. Szostak and Walkowiak[137] compared the efficacy of different ML methods, including FFNN, SVM, DT, random forest (RF), and linear discriminant analysis (LDA), for the problem of predicting source and destination for demands in a dynamic optical network setting. Furthermore, this was extended by including the prediction of traffic volume and holding time.[138] They observed that the best classifier for such tasks was LDA.[137] Additionally, Gui *et al.*[143] benchmarked their GCN-GRU based traffic prediction over several approaches, including LSTM, CNN, and GRU, and the results suggested that GCN-GRU has a greater prediction quality as compared to these other approaches.

As introduced in Sec. II, GANs are designed for realistic data generation and thus show potential in simulated traffic generation for optical networks. In a GAN-based traffic data generation scenario,[148] the objective of the generator is to transfer the random noise into the generated traffic data and attempt to make the characteristics of generated traffic data close to those of the real world traffic data. In contrast, the discriminator attempts to correctly determine whether the data are from the actual traffic dataset or the generated traffic dataset. Via intense competition, the discriminator and the generator are improved by each other and the generated traffic data become increasingly similar to the actual real world traffic data.

B. Core network parameter optimization

In this section, we intend to discuss the core optical network parameter optimization given in the frameworks of RL. Core optical networks play the most substantial role in the national and international communication infrastructure. They typically consist of flexible devices, such as the re-configurable optical add/drop multiplexers (ROADMs) and bandwidth variable transponders (BVTs). ROADMs are commonly used to transmit optical signals between different nodes, whereas BVTs are used to adapt a large set of core optical network parameters, such as signal modulation format, coding scheme, forward error correction overhead, and symbol rate, based on the current optical link requirements. Adopting the core optical network parameters is especially vital when attempting to maximize the ultimate network information throughput. However, this procedure requires the optimization of a large parameter space. In addition, finding much more efficient use of core optical network spectral resources is essential to cope with ever-growing bandwidth demand.

Conventionally, in the case of fixed-grid WDM optical networks with a static traffic request assumption, the network parameters adjustment can be realized via adapting launch power per channel and signal modulation format with regard to stochastic system impairments in the physical layer.[149] The typical core optical network physical layer impairments occurred between its nodes are the amplified spontaneous emission noise arising from the optical amplifiers and the nonlinear interference noise-like distortions induced by the four-wave mixing process in Kerr-type nonlinear media, i.e., in the optical fiber. In essence, the exact behavior of optical data signals between two nodes can be obtained numerically by solving the NLSE/Manakov equation via the SSFTM, when the step-size tends toward zero. However, the numerical solution is a comparatively time-consuming process, especially for wideband transmission systems. Currently, the most widely used physical layer impairments models are the family of the so-called Gaussian noise (GN) models, which commonly rely on the first-order perturbation theory.[66] Moreover, under fairly reasonable assumptions, these models admit analytical closed-form approximations that significantly speed up the evaluation of the physical layer impairments. The resource allocation problem in the case of a single flexible-grid fiber link via the GN model closed-form approximation was considered in Ref. 150. Here, it is also worth mentioning that the possibility of quickly performing physical layer impairments estimations is essential regardless of the type of optimization frameworks.

RL has recently appeared as an alternative to conventional approaches, such as integer linear programming (ILP),[151,152] and heuristics, such as simulated annealing, *k*-shortest path routing and first-fit[153] and the genetic algorithm (GA).[154] Generally speaking, RL is capable of efficiently overcoming a wide class of complex optimization problems.[155] However, in the context of core optical networks, RL cannot be applicable straight away, as it must be generalized to learn over arbitrary network topologies with dynamically changing scenarios, such as network topology, traffic, routing, and link failures. Over the last few years, some initial works have suggested deep RL for solving various resource allocation and dynamic routing problems in core optical networks,[156–159] in which the advantages of using RL-enabled methods over traditional heuristic optimization algorithms were emphasized.

Yet, more interesting examples of using an RL framework for maximizing the point-to-point link capacity by means of adjusting controllable parameters in core optical networks have been recently reported in Refs. 160 and 161, where the heuristic GA based results were used as a performance benchmark. The predicted performance of these two approaches remains very similar. However, after an initial training phase, the computation time of BVT parameters optimization to maximize the overall network throughput based on the RL approach is up to 1 second on average, while traditional heuristic algorithms may take in the order of minutes to hours. Additionally, preliminary investigations into network routing and parameter optimization show promising potential in leveraging the ability of GNNs to learn and model graph-structured information.[162,163] Such models are able to generalize over arbitrary network topologies, routing schemes, and traffic intensity.

V. FUTURE DIRECTIONS

A number of ML-driven future research directions are emerging within optical networks across both the physical layer and the network layer. In this section, we outline selected future directions within the physical layer, the network layer, and those spanning both layers.

A. Physical layer

An emerging theme within applied ML that is interesting in the context of optical networks is explainable ML,[32] a subset of explainable artificial intelligence[164] that aims to make the processes by which ML algorithms make decisions more understandable to humans. Optical networks are operated with high availability, meaning that light paths must stay within the accepted QoT ranges often at least 99.999% of the time, which translates to just over 5 minutes of downtime per year.[165] This is enforced by service level agreements, which mean that operators must deliver the quality of service that customers have paid for. As a result, ML approaches deployed on optical networks must meet the stringent reliability requirements that are already satisfied by conventional techniques. Thus, understanding how ML algorithms work is crucial for adoption. Both post-hoc explainability methods and inherently explainable ML approaches have potential to yield substantial benefits for ML applied within optical communication networks. There are now open-source libraries that provide implementations of post-hoc techniques,[166] making their application convenient. It may be better to have a more easily interpretable model with slightly worse performance in some situations if operators can understand how it makes decisions and therefore can be more confident in its reliability. Additionally, probabilistic ML methods such as GPs provide well-quantified predictive uncertainties that can aid with the interpretation of ML model predictions, which would be greatly beneficial for many applications of ML within optical networks.

Another interesting avenue for future research is the combination of physical models with ML, so as to embed our knowledge of system physics into models such as NNs and GPs, as discussed in Sec. III A. For example, physics-informed ML approaches to QoT estimation can allow us to train models with fewer measurements of the system and enhance model explainability. Additionally, we can use our knowledge of the physics to design more effective model architectures. For example, NNs can be designed using the DBP structure for nonlinear noise mitigation. Certainly, the concept of utilizing the information available before we have seen the data and the data itself, rather than discarding this and relying solely on the data, presents an interesting research direction.

A further promising future research direction is digital twins—having been shown to be effective in other research areas, such as healthcare technology, manufacturing, and smart cities,[88] there are many open research questions for the development of digital twins for optical networks. The realization of true digital twins for optical networks, meaning a high-fidelity virtual copy of a deployed network, will require the amalgamation of models of all aspects of optical networks discussed in this Tutorial. It will also require access to high-quality datasets that are representative of deployed networks, as described above. Additionally, there is an important question regarding how fast digital twins will operate and whether a truly real-time digital twin is realizable. This depends on two factors—how dynamic installed networks become in the future and how operator confidence in ML approaches evolves over time. As networks become increasingly more dynamic, meaning that light paths are established and torn down with greater frequency, the time required to accurately measure the network may begin to form a bottleneck for how fast a digital twin can respond to a change in the network. Moreover, the time taken to retrain models may also limit this responsivity, meaning that online and transfer learning will likely be needed to ensure that ML models remain accurate as the network changes and to support rapid modeling of new light paths. Operator confidence in ML is also crucial as a true digital twin framework requires automatic control of the network based on data. As a result, explainability techniques are important for the development of digital twins as they will increase confidence in the ML models upon which the digital twins are built.

Furthermore, work is required to reduce the complexity of ML algorithms, in order for them to be successfully deployed with a reasonable use of computational resources. For example, in short reach equalizer applications, lower complexity ML is desirable due to the requirements for real-time equalization at high symbol rates. In general, ML techniques will need to have sufficiently low complexity in order to adapt to increasingly dynamic networks. One solution to this may be online learning, where ML models can be trained offline before deployment and adapt to monitoring data once deployed without completely re-training the model. An additional related challenge is the flexibility of ML algorithms—to what extent can the deployed models generalize to cover different network scenarios? One potential solution to this issue is transfer learning, which has been proposed as a method for increasing the flexibility of NNs for fiber nonlinear noise mitigation by re-using some of the initial trained network weights to adapt to a new situation.

An additional future direction is provided by hardware-driven ML approaches to equalization and nonlinearity compensation problems in optical networks. Due to the challenging requirements to operate at real-time data rates, the use of specialist hardware such as FPGAs is crucial for these applications. Low complexity implementations of ML architectures, such as DNN and RNN equalizers[104,110] and real-time nonlinearity compensation[126] discussed in Sec. III, present an interesting future direction for performing such signal processing tasks in next generation optical networks.

B. Network layer

As in the physical layer, explainable ML is a promising field of research within network layer applications. Similarly, reduction in ML algorithm complexity is also an interesting future direction for network layer applications, particularly for any methods which are required to work in an online scenario.

Obtaining sufficiently detailed datasets from deployed networks remains a significant challenge for ML research in optical networks. Such data may often be difficult to find, as network operators may not be able to grant researchers access to detailed network data without a non-disclosure agreement, due to competition-related concerns. In the cases where such data are provided,[167,168] it could still be insufficiently detailed to be of use. As discussed in Sec. IV, GANs show potential to address this issue to some extent with their ability to generate larger datasets from a small amount of input data. To this end, GANs have been successful in generating data that are indistinguishable from real world input data in optical network traffic generation applications[148] and numerous other applications in the computer vision domain.[169]

An additional promising research direction in network routing and parameter optimization is leveraging the ability of GNNs to learn and model graph-structured information to create models that are able to generalize over arbitrary network topologies, routing schemes, and traffic intensity.[162,163] Furthermore, the preliminary works applying RL techniques in dynamic parameter optimization have shown strong potential, with faster response time and similar quality of solutions compared to conventional optimization approaches.[156] To this end, it would be interesting to investigate the means of bringing the strengths of RL and GNNs together in a data-driven network routing and parameter optimization scenario.

Moreover, within traffic prediction and generation, future work includes extending the proposed methodologies to networks of different scales, such as core and access networks. Another potential direction is the introduction of novel methods that have been used successfully in time series forecasting problems in other domains, such as echo state networks,[170] and combining existing ML approaches to develop more effective hybrid methods. For example, hybrid models of GNNs and LSTMs could be investigated as these harness both the knowledge of the network structure and the temporal aspects of the traffic, respectively. Finally, integrating traffic prediction and simulation modules with other modules in a SDN setting will aid in achieving high performance in increasingly dynamic and flexible networks.

VI. CONCLUSIONS

In this Tutorial, we have outlined the key research challenges in optical networks that exist today, the ML techniques that have been proposed to solve these problems, and interesting works from the literature that have applied ML. We have introduced the crucial concepts required to navigate ML literature and highlighted techniques that are commonly used in optical networks: various forms of NNs, Bayesian approaches such as GPs, classifiers such as SVMs, and RL techniques such as deep Q-learning. In the physical layer, we have surveyed the literature applying ML to QoT estimation, digital twins, equalization for short reach networks, and nonlinear noise mitigation for long-haul systems. In the network layer, we have presented exemplary work tackling network traffic prediction and generation and the optimization of core network parameters. Thus, there has been a significant progress in ML applied to optical networks, with a vast range of methods utilized, each yielding benefits over previous approaches. There remain a number of interesting avenues for future research as discussed above, which will be crucial in delivering the next generation of optical networks and meeting the service requirements of the future.

ACKNOWLEDGMENTS

The authors acknowledge BT, Huawei, and the EPSRC [IPES CDT (Grant No. EP/L015455/1) and TRANSNET (Grant No. EP/R035342/1)] for funding.

AUTHOR DECLARATIONS

Conflict of Interest

We confirm that we do not have any conflicts of interest.

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

NOMENCLATURE

ADC	analog-to-digital converter
ANN	artificial neural network
APD	avalanche photodiode
BER	bit error rate
BVT	bandwidth variable transponder
CBR	case-based reasoning
CD	chromatic dispersion
CNN	convolutional neural network
DBP	digital back-propagation
DD	direct detection
DDPG	deep deterministic policy gradient
DFE	decision feedback equalizer
DL	deep learning
DML	directly modulated laser
DNN	deep neural network
DP	dual polarization
DQN	deep Q-network
DRL	deep reinforcement learning
DSP	digital signal processing
DT	decision tree
ELM	extreme learning machine
EML	electro-absorption modulator
EON	elastic optical network
FFE	feed-forward equalizer
FFNN	feed-forward neural network
FPGA	field programmable gate array
GA	genetic algorithm
GAN	generative adversarial network
GCN	graph convolutional network
GN	Gaussian noise
GNN	graph neural network
GRU	gated recurrent unit
ILP	integer linear programming
IM	intensity modulation
KNN	K-nearest neighbor
LDA	linear discriminant analysis
LMA	Levenberg–Marquardt algorithm
LSTM	long-short term memory
MDP	Markov decision process
ML	machine learning
MMF	multi-mode fiber
NLSE	nonlinear Schrödinger equation
NN	neural network
OFDM	orthogonal frequency-division multiplexing
OOK	on–off keying
OSNR	optical signal-to-noise ratio

PAM	pulse amplitude modulation
PSD	power spectral density
PW	Parzen window
QAM	quadrature amplitude modulation
QoT	quality of transmission
QPSK	quadrature phase shift keying
ReLU	rectified linear unit
RF	random forest
RL	reinforcement learning
RNN	recurrent neural network
ROADM	re-configurable optical add/drop multiplexer
RZ	return-to-zero
SCSP	single-channel single polarization
SDN	software defined network
SMF	single-mode fiber
SNR	signal-to-noise ratio
SOA	semiconductor optical amplifier
SSFTM	split-step Fourier transform method
VCSEL	vertical cavity surface emitting laser
WDM	wavelength division multiplexing

REFERENCES

[1] T. M. Mitchell, *Machine Learning* (McGraw-Hill, 1997).

[2] R. Stuart and P. Norvig, *Artificial Intelligence: A Modern Approach* (Pearson Education, 2003).

[3] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," IEEE Commun. Surv. Tutorials **21**, 1383–1408 (2018).

[4] F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An optical communication's perspective on machine learning and its applications," J. Lightwave Technol. **37**, 493–516 (2019).

[5] J. Mata, I. de Miguel, R. J. Durán, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," Opt. Switching Networking **28**, 43–57 (2018).

[6] Y. Pointurier, "Machine learning techniques for quality of transmission estimation in optical networks," J. Opt. Commun. Networking **13**, B60–B71 (2021).

[7] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," IEEE Trans. Cognit. Commun. Networking **3**, 563–575 (2017).

[8] D. Rafique and L. Velasco, "Machine learning for network automation: Overview, architecture, and applications [invited tutorial]," J. Opt. Commun. Networking **10**, D126–D143 (2018).

[9] G. P. Agrawal, *Fiber-Optic Communication Systems* (Wiley, 2010).

[10] H. Zimmermann, "OSI reference model-the ISO model of architecture for open systems interconnection," IEEE Trans. Commun. **28**, 425–432 (1980).

[11] CISCO systems, annual internet report, white paper, San Jose, CA, 2020.

[12] A. D. Ellis, M. E. McCarthy, M. A. Z. Al Khateeb, M. Sorokina, and N. J. Doran, "Performance limits in optical communications due to fiber nonlinearity," Adv. Opt. Photonics **9**, 429–503 (2017).

[13] G. P. Agrawal, *Nonlinear Fiber Optics* (Elsevier, 2019).

[14] P. P. Mitra and J. B. Stark, "Nonlinear limits to the information capacity of optical fibre communications," Nature **411**, 1027–1030 (2001).

[15] T. Zhu, X. Bao, L. Chen, H. Liang, and Y. Dong, "Experimental study on stimulated Rayleigh scattering in optical fibers," Opt. Express **18**, 22958–22963 (2010).

[16] N. A. Shevchenko, S. Nallaperuma, and S. J. Savory, "Ultra-wideband information throughput attained via launch power allocation," in *2021 International Conference on Optical Network Design and Modeling (ONDM)* (IEEE, 2021), pp. 1–3.

[17] A. R. Chraplyvy, "Limitations on lightwave communications imposed by optical-fiber nonlinearities," J. Lightwave Technol. **8**, 1548–1557 (1990).

[18] I. Roberts, J. M. Kahn, J. Harley, and D. W. Boertjes, "Channel power optimization of WDM systems following Gaussian noise nonlinearity model in presence of stimulated Raman scattering," J. Lightwave Technol. **35**, 5237 (2017).

[19] M. Cantono, D. Pilori, A. Ferrari, C. Catanese, J. Thouras, J.-L. Augé, and V. Curri, "On the interplay of nonlinear interference generation with stimulated Raman scattering for QoT estimation," J. Lightwave Technol. **36**, 3131–3141 (2018).

[20] R. M. Shelby, M. D. Levenson, and P. W. Bayer, "Guided acoustic-wave Brillouin scattering," Phys. Rev. B **31**, 5244 (1985).

[21] P. Serena, F. Poli, A. Bononi, and J.-C. Antona, "Scattering efficiency of thermally excited GAWBS in fibres for optical communications," in *Proceedings of the 45th European Conference on Optical Communication (ECOC)* (IET, 2019), pp. 1–4.

[22] G. S. Zervas and D. Simeonidou, "Cognitive optical networks: Need, requirements and architecture," in *Proceedings of the 12th International Conference on Transparent Optical Networks (ICTON), Munich, Germany* (IEEE, 2010), pp. 1–4.

[23] E. Ip, A. P. T. Lau, D. J. F. Barros, and J. M. Kahn, "Coherent detection in optical fiber systems," Opt. Express **16**, 753–791 (2008).

[24] K. Kirkpatrick, "Software-defined networking," Commun. ACM **56**, 16–19 (2013).

[25] I. de Miguel *et al.*, "Cognitive dynamic optical networks," J. Opt. Commun. Networking **5**, A107–A118 (2013).

[26] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).

[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).

[28] Ó. Fontenla-Romero, B. Guijarro-Berdiñas, D. Martinez-Rego, B. Pérez-Sánchez, and D. Peteiro-Barral, *Online Machine Learning* (IGI Global, PA, 2013), pp. 27–54.

[29] C. Pehlevan and D. B. Chklovskii, "Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks," IEEE Signal Process. Mag. **36**, 88–96 (2019).

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng. **22**, 1345–1359 (2009).

[31] W. Fuhl *et al.*, "Explainable online validation of machine learning models for practical applications," in *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)* (IEEE, 2021), pp. 3304–3311.

[32] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," IEEE Access **8**, 42200–42216 (2020).

[33] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," Queue **16**, 31–57 (2018).

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust You?' Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2016), pp. 1135–1144.

[35] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the International Conference on Neural Information Processing Systems* (NeurIPS, 2017), pp. 4768–4777.

[36] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 6541–6549.

[37] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks **2**, 359–366 (1989).

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016), http://www.deeplearningbook.org.

[39] R. Rojas, "The backpropagation algorithm," in *Neural Networks: A Systematic Introduction* (Springer, Berlin, Heidelberg, 1996), pp. 149–182.

[40] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747 (2016).

[41] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (Wiley, 2001).

[42] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE Trans. Neural Networks **5**, 157–166 (1994).

[43] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).

[45] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Trans. Neural Networks **20**, 61–80 (2008).

[46] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," AI Open **1**, 57–81 (2020).

[47] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the International Conference on Machine Learning (ICML)* (PMLR, 2017), pp. 1263–1272.

[48] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[49] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International conference on Learning Representations*, (2018); available at https://openreview.net/forum?id=rJXMpikCZ.

[50] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated graph sequence neural networks," in *Proceedings of ICLR* (ICLR, 2016).

[51] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)* (MIT Press, Cambridge, MA, 2014), pp. 2672–2680.

[52] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2006).

[53] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*, Wiley Series in Probability and Statistics (Wiley, 2013).

[54] H. Lui, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," IEEE Trans. Neural Networks Learn. Syst. **31**, 4405–4423 (2020).

[55] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," J. Mach. Learn. Res. **2**, 125–137 (2001).

[56] M. Lapan, *Deep Reinforcement Learning Hands-On: Apply Modern RL Methods, with Deep Q-Networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and More* (Packt Publishing Ltd., 2018).

[57] D. E. Kirk, *Optimal Control Theory: An Introduction* (Prentice-Hall, 1970).

[58] M. van Otterlo and M. Wiering, "Reinforcement learning and Markov decision processes," in *Reinforcement Learning* (Springer, 2012), pp. 3–42.

[59] H. Wang, T. Zariphopoulou, and X. Zhou, "Exploration versus exploitation in reinforcement learning: A stochastic control approach," arXiv:1812.01552 (2019); available at https://ssrn.com/abstract=3316387.

[60] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the International Conference on Machine Learning (ICML)* (PMLR, 2016), pp. 1928–1937.

[61] H. Dong *et al.*, *Deep Reinforcement Learning: Fundamentals, Research and Applications* (Springer Nature, 2020).

[62] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," Nature **518**, 529–533 (2015).

[63] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," arXiv:1509.02971 (2015).

[64] D. Silver *et al.*, "Deterministic policy gradient algorithms," in *Proceedings of the International Conference on Machine Learning (ICML)* (PMLR, 2014), pp. 387–395.

[65] W. Freude *et al.*, "Quality metrics for optical signals: Eye diagram, Q-factor, OSNR, EVM and BER," in *Proceedings of the 14th International Conference on Transparent Optical Networks (ICTON)* (IEEE, 2012), pp. 1–4.

[66] P. Poggiolini, "The GN model of non-linear propagation in uncompensated coherent optical systems," J. Lightwave Technol. **30**, 3857–3879 (2012).

[67] E. Ip and J. M. Kahn, "Compensation of dispersion and nonlinear impairments using digital backpropagation," J. Lightwave Technol. **26**, 3416–3425 (2008).

[68] J. Pesic, T. Zami, P. Ramantanis, and S. Bigo, "Faster return of investment in WDM networks when elastic transponders dynamically fit ageing of link margins," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2016), pp. 1–3.

[69] E. Seve *et al.*, "Automated fiber type identification in SDN-enabled optical networks," J. Lightwave Technol. **37**, 1724–1731 (2019).

[70] E. Seve, J. Pesic, C. Delezoide, S. Bigo, and Y. Pointurier, "Learning process for reducing uncertainties on network parameters and design margins," J. Opt. Commun. Networking **10**, A298–A306 (2018).

[71] M. Bouda *et al.*, "Accurate prediction of quality of transmission based on a dynamically configurable optical impairment model," J. Opt. Commun. Networking **10**, A102–A109 (2018).

[72] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis* (Springer, 1978).

[73] A. Mahajan, K. Christodoulopoulos, R. Martínez, R. Muñoz, and S. Spadaro, "Adaptive and iterative QoT estimator retraining for launch power optimization," in Proceedings of the Optical Fiber Communication Conference (OFC), 2021.

[74] J. Wass, J. Thrane, M. Piels, R. Jones, and D. Zibar, "Gaussian process regression for WDM system performance prediction," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2017), pp. 1–3.

[75] S. Oda *et al.*, "A learning living network with open ROADMs," J. Lightwave Technol. **35**, 1350–1356 (2017).

[76] J. Müller *et al.*, "Estimating quality of transmission in a live production network using machine learning," in Proceedings of the Optical Fiber Communication Conference (OFC), 2021.

[77] A. D'Amico *et al.*, "Using machine learning in an open optical line system controller," J. Opt. Commun. Networking **12**, C1–C11 (2020).

[78] A. S. Kashi, J. C. Cartledge, and W.-Y. Chan, "Neural network training framework for nonlinear signal-to-noise ratio estimation in heterogeneous optical networks," in Proceedings of the Optical Fiber Communication Conference (OFC), 2021.

[79] T. Panayiotou, S. P. Chatzis, and G. Ellinas, "Performance analysis of a data-driven quality-of-transmission decision approach on a dynamic multicast-capable metro optical network," J. Opt. Commun. Networking **9**, 98–108 (2017).

[80] G. Ellinas, N. Antoniades, T. Panayiotou, A. Hadjiantonis, and A. M. Levine, "Multicast routing algorithms based on *Q*-factor physical-layer constraints in metro networks," IEEE Photonics Technol. Lett. **21**, 365–367 (2009).

[81] J. Mata *et al.*, "A SVM approach for lightpath QoT estimation in optical transport networks," in *Proceedings of the IEEE International Conference on Big Data (Big Data)* (IEEE, 2017), pp. 4795–4797.

[82] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

[83] X. Jiang *et al.*, "Solving the nonlinear Schrödinger equation in optical fibers using physics-informed neural network," in Proceedings of the Optical Fiber Communication Conference (OFC), 2021.

[84] J. W. Nevin, F.-J. Vaquero-Callabero, D. J. Ives, and S. J. Savory, "Physics-informed Gaussian process regression for optical fiber communication systems," J. Lightwave Technol. **39**, 6833 (2021).

[85] Q. Zhuge *et al.*, "Application of machine learning in fiber nonlinearity modeling and monitoring for elastic optical networks," J. Lightwave Technol. **37**, 3055–3063 (2019).

[86] J. Pesic, "Missing pieces currently preventing effective application of machine learning to QoT estimation in the field," in Proceedings of the Optical Fiber Communication Conference (OFC), 2021.

[87] M. Lonardi, J. Pesic, T. Zami, and N. Rossi, "The perks of using machine learning for QoT estimation with uncertain network parameters," in *Photonic Networks and Devices* (Optical Society of America, 2020), p. NeM3B.2.

[88] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," IEEE Access **8**, 21980–22012 (2020).

[89] D. Wang *et al.*, "The role of digital twin in optical communication: Fault management, hardware configuration, and transmission simulation," IEEE Commun. Mag. **59**, 133–139 (2021).

[90] C. Zhang *et al.*, "Temporal data-driven failure prognostics using BiGRU for optical networks," J. Opt. Commun. Networking **12**, 277–287 (2020).

[91] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2016), pp. 785–794.

[92] C. Zhang *et al.*, "Interpretable learning algorithm based on XGBoost for fault prediction in optical network," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2020), pp. 1–3.

[93] J. Li, D. Wang, M. Zhang, and S. Cui, "Digital twin-enabled self-evolved optical transceiver using deep reinforcement learning," Opt. Lett. **45**, 4654–4657 (2020).

[94] D. Wang *et al.*, "Data-driven optical fiber channel modeling: A deep learning approach," J. Lightwave Technol. **38**, 4730–4743 (2020).

[95] M. P. Yankov and F. Da Ros (2020). "Input-output power spectral densities for three C-band EDFAs and four multi-span inline EDFAd fiber optic systems of different lengths," Dataset. https://doi.org/10.11583/DTU.13135754.v1; accessed July 29, 2021.

[96] S. Li *et al.*, "Digital twin-enabled power optimizer for multi-span transmission system using autoencoder," in Proceedings of the Optical Fiber Communication Conference (OFC), 2021.

[97] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," J. Lightwave Technol. **36**, 4843–4855 (2018).

[98] B. Karanov *et al.*, "End-to-end learning in optical fiber communications: Experimental demonstration and future trends," in *Proceedings of the European Conference on Optical Communications (ECOC)* (IEEE, 2020), pp. 1–4.

[99] K. Zhong *et al.*, "Digital signal processing for short-reach optical communications: A review of current technologies and future trends," J. Lightwave Technol. **36**, 377–400 (2018).

[100] H. Zhou *et al.*, "Recent advances in equalization technologies for short reach optical links based on PAM4 modulation: A review," Appl. Sci. **9**, 2342 (2019).

[101] N. Stojanovic, F. Karinou, Z. Qiang, and C. Prodaniuc, "Volterra and Wiener equalizers for short-reach 100G PAM-4 applications," J. Lightwave Technol. **35**, 4583–4594 (2017).

[102] L. Yi *et al.*, "Machine learning for 100Gb/s/λ passive optical network," J. Lightwave Technol. **37**, 1621–1630 (2019).

[103] J. Estaran *et al.*, "Artificial neural networks for linear and non-linear impairment mitigation in high-baudrate IM/DD systems," in *Proceedings of the 42nd European Conference on Optical Communication* (IEEE, 2016), pp. 1–3.

[104] N. Kaneda *et al.*, "FPGA implementation of deep neural network based equalizers for high-speed PON," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2020), p. T4D.2.

[105] P. Li, L. Yi, L. Xue, and W. Hu, "56 Gbps IM/DD PON based on 10G-class optical devices with 29 dB loss budget enabled by machine learning," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2018), p. M2B.2.

[106] C.-Y. Chuang *et al.*, "Convolutional neural network based nonlinear classifier for 112-Gbps high speed optical link," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2018), p. W2A.43.

[107] C. Ye *et al.*, "Recurrent neural network (RNN) based end-to-end nonlinear management for symmetrical 50Gbps NRZ PON with 29dB+ loss budget," in *Proceedings of the European Conference on Optical Communication (ECOC)* (IEEE, 2018), pp. 1–3.

[108] Z. Xu, C. Sun, T. Ji, J. H. Manton, and W. Shieh, "Feedforward and recurrent neural network-based transfer learning for nonlinear equalization in short-reach optical links," J. Lightwave Technol. **39**, 475–480 (2021).

[109] Z. Xu, C. Sun, T. Ji, J. H. Manton, and W. Shieh, "Computational complexity comparison of feedforward/radial basis function/recurrent neural network-based equalizer for a 50-Gb/s PAM4 direct-detection optical link," Opt. Express **27**, 36953–36964 (2019).

[110] X. Huang, D. Zhang, X. Hu, C. Ye, and K. Zhang, "Recurrent neural network based equalizer with embedded parallelization for 100Gbps/λ PON," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2021), p. M3G.2.

[111] X. Dai, X. Li, M. Luo, Q. You, and S. Yu, "LSTM networks enabled nonlinear equalization in 50-Gb/s PAM-4 transmission links," Appl. Opt. **58**, 6079–6084 (2019).

[112] G. Chen *et al.*, "Nonlinear distortion mitigation by machine learning of SVM classification for PAM-4 and PAM-8 modulated optical interconnection," J. Lightwave Technol. **36**, 650–657 (2018).

[113] X. Miao, M. Bi, J. Yu, L. Li, and W. Hu, "SVM-modified-FFE enabled chirp management for 10G DML-based 50Gb/s/λ PAM4 IM-DD PON," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2019), p. M2B.5.

[114] E. Giacoumidis *et al.*, "Experimental comparison of artificial neural network and Volterra based nonlinear equalization for CO-OFDM," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2016), p. W3A.4.

[115] T. Kyono, Y. Otsuka, Y. Fukumoto, S. Owaki, and M. Nakamura, "Computational-complexity comparison of artificial neural network and Volterra series transfer function for optical nonlinearity compensation with time- and frequency-domain dispersion equalization," in *Proceedings of the European Conference on Optical Communication (ECOC)* (IEEE, 2018), p. Th2.28.

[116] D. Marcuse, C. R. Manyuk, and P. K. A. Wai, "Application of the Manakov-PMD equation to studies of signal propagation in optical fibers with randomly varying birefringence," J. Lightwave Technol. **15**, 1735–1746 (1997).

[117] G. Gao, X. Chen, and W. Shieh, "Influence of PMD on fiber nonlinearity compensation using digital back propagation," Opt. Express **20**, 14406–14418 (2012).

[118] T. S. R. Shen and A. P. T. Lau, "Fiber nonlinearity compensation using extreme learning machine for DSP-based coherent communication systems," in *16th Opto-Electronics and Communications Conference* (IEEE, 2011), pp. 816–817.

[119] S. Deligiannidis, A. Bogris, C. Mesaritakis, and Y. Kopsinis, "Compensation of fiber nonlinearities in digital coherent systems leveraging long short-term memory neural networks," J. Lightwave Technol. **38**, 5991–5999 (2020).

[120] M. Li *et al.*, "Nonparameter nonlinear phase noise mitigation by using M-ary support vector machine for coherent optical systems," IEEE Photonics J. **5**, 7800312 (2013).

[121] D. Wang *et al.*, "Nonlinearity mitigation using a machine learning detector based on *k*-nearest neighbors," IEEE Photonics Technol. Lett. **28**, 2102–2105 (2016).

[122] J. Zhang, W. Chen, M. Gao, and G. Shen, "K-means-clustering-based fiber nonlinearity equalization techniques for 64-QAM coherent optical communication system," Opt. Express **25**, 27570–27580 (2017).

[123] A. Amari, X. Lin, O. A. Dobre, R. Venkatesan, and A. Alvarado, "A machine learning-based detection technique for optical fiber nonlinearity mitigation," IEEE Photonics Technol. Lett. **31**, 627–630 (2019).

[124] E. Giacoumidis *et al.*, "A blind nonlinearity compensator using DBSCAN clustering for coherent optical transmission systems," Appl. Sci. **9**, 4398 (2019).

[125] I. Aldaya *et al.*, "Histogram based clustering for nonlinear compensation in long reach coherent passive optical networks," Appl. Sci. **10**, 152 (2020).

[126] E. Giacoumidis, Y. Lin, M. Blott, and L. P. Barry, "Real-time machine learning based fiber-induced nonlinearity compensation in energy-efficient coherent optical networks," APL Photonics **5**, 041301 (2020).

[127] M. Schädler, G. Böcherer, and S. Pachnicke, "Soft-demapping for short reach optical communication: A comparison of deep neural networks and Volterra series," J. Lightwave Technol. **39**, 3095–3105 (2021).

[128] C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (IEEE, 2018), pp. 1–3.

[129] Q. Fan, G. Zhou, T. Gui, C. Lu, and A. P. T. Lau, "Advancing theoretical understanding and practical performance of signal processing for nonlinear optical communications through machine learning," Nat. Commun. **11**, 3694 (2020).

[130] S. Zhang *et al.*, "Field and lab experimental demonstration of nonlinear impairment compensation using neural networks," Nat. Commun. **10**, 3033 (2019).

[131] Z. Tao *et al.*, "Multiplier-free intrachannel nonlinearity compensating algorithm operating at symbol rate," J. Lightwave Technol. **29**, 2570–2576 (2011).

[132] E. Giacoumidis *et al.*, "Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based nonlinear equalization," Opt. Lett. **40**, 5113–5116 (2019).

[133] I. Aldaya *et al.*, "Compensation of nonlinear distortion in coherent optical OFDM systems using a MIMO deep neural network-based equalizer," Opt. Lett. **45**, 5820–5823 (2020).

[134] E. Giacoumidis *et al.*, "Comparison of DSP-based nonlinear equalizers for intra-channel nonlinearity compensation in coherent optical OFDM," Opt. Lett. **41**, 2509–2512 (2016).

[135] E. Giacoumidis *et al.*, "Reduction of nonlinear inter subcarrier intermixing in coherent optical OFDM by a fast Newton-based support vector machine nonlinear equalizer," J. Lightwave Technol. **35**, 2391–2397 (2017).

[136] P. J. Freire *et al.*, "Transfer learning for neural networks-based equalizers in coherent optical systems," J. Lightwave Technol. **39**, 6733–6745 (2021).

[137] D. Szostak and K. Walkowiak, "Machine learning methods for traffic prediction in dynamic optical networks with service chains," in Proceedings of the 21st International Conference on Transparent Optical Networks (ICTON), 2019.

[138] D. Szostak, K. Walkowiak, and A. Włodarczyk, "Short-term traffic forecasting in optical network using linear discriminant analysis machine learning classifier," in *22nd International Conference on Transparent Optical Networks (ICTON)* (IEEE, 2020), pp. 1–4.

[139] G. Choudhury, D. Lynch, G. Thakur, and S. Tse, "Two use cases of machine learning for SDN-enabled IP/optical networks: Traffic matrix prediction and optical path performance prediction [invited]," J. Opt. Commun. Networking **10**, D52–D62 (2018).

[140] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *2011 Proceedings IEEE INFOCOM* (IEEE, 2011), pp. 1503–1511.

[141] M. Aibin, "Traffic prediction based on machine learning for elastic optical networks," Opt. Switching Networking **30**, 33–39 (2018).

[142] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, "Deep learning-based traffic prediction for network optimization," in *20th International Conference on Transparent Optical Networks (ICTON)* (IEEE, 2018), pp. 1–4.

[143] Y. Gui, D. Wang, L. Guan, and M. Zhang, "Optical network traffic prediction based on graph convolutional neural networks," in *Proceedings of the Opto-Electronics and Communications Conference (OECC)* (IEEE, 2020), pp. 1–3.

[144] X. Zhu, O. Xu, and G. Li, "Prediction accuracy improvement of passive optical network traffic by a LSTM model with a new activation function," in *Proceedings of the 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)* (IEEE, 2020), pp. 662–666.

[145] C. Vinchoff, N. Chung, T. Gordon, L. Lyford, and M. Aibin, "Traffic prediction in optical networks using graph convolutional generative adversarial networks," in *Proceedings of the 22nd International Conference on Transparent Optical Networks (ICTON)* (IEEE, 2020), pp. 1–4.

[146] K. Zhan, H. Yang, A. Yu, Q. Yao, and J. Zhang, "Multi-path pre-reserved resource allocation based on tidal traffic prediction in metropolitan optical network," in Proceedings of the 17th International Conference on Optical Communications and Networks, 2018.

[147] G. Choudhury, G. Thakur, and S. Tse, "Joint optimization of packet and optical layers of a core network using SDN controller, CD ROADMs and machine-learning-based traffic prediction," in *Proceedings of the Optical Fiber Communication Conference (OFC)* (Optical Society of America, 2019), p. M2A.1.

[148] J. Li *et al.*, "Deep learning based adaptive sequential data augmentation technique for the optical network traffic synthesis," Opt. Express **27**, 18831–18847 (2019).

[149] D. J. Ives and S. J. Savory, "Transmitter optimized optical networks," in *National Fiber Optic Engineers Conference* (Optical Society of America, 2013), p. JW2A.64.

[150] L. Yan, E. Agrell, H. Wymeersch, P. Johannisson, R. Di Taranto, and M. Brandt-Pearce, "Link-level resource allocation for flexible-grid nonlinear fiber-optic communication systems," IEEE Photonics Technol. Lett. **27**, 1250–1253 (2015).

[151] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *2011 Proceedings IEEE INFOCOM* (IEEE, 2011), pp. 1503–1511.

[152] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," IEEE J. Sel. Areas Commun. **31**, 26–38 (2012).

[153] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," J. Lightwave Technol. **29**, 1354–1366 (2011).

[154] X. Zhou, W. Lu, L. Gong, and Z. Zhu, "Dynamic RMSA in elastic optical networks with an adaptive genetic algorithm," in *2012 IEEE Global Communications Conference (GLOBECOM)* (IEEE, 2012), pp. 2912–2917.

[155] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering Atari, Go, chess and shogi by planning with a learned model," Nature **588**, 604–609 (2020).

[156] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," J. Lightwave Technol. **37**, 4155–4163 (2019).

[157] X. Luo, C. Shi, L. Wang, X. Chen, Y. Li, and T. Yang, "Leveraging double-agent-based deep reinforcement learning to global optimization of elastic optical networks with enhanced survivability," Opt. Express **27**, 7896–7911 (2019).

[158] J. Suárez-Varela, A. Mestres, J. Yu, L. Kuang, H. Feng, A. Cabellos-Aparicio, and P. Barlet-Ros, "Routing in optical transport networks with deep reinforcement learning," J. Opt. Commun. Networking **11**, 547–558 (2019).

[159] C. Natalino and P. Monti, "The optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)* (IEEE, 2020), pp. 1–5.

[160] R. Weixer, S. Kühl, R. M. Morais, B. Spinnler, W. Schairer, B. Sommerkorn-Krombholz, and S. Pachnicke, "A reinforcement learning framework for parameter optimization in elastic optical networks," in *2020 European Conference on Optical Communications (ECOC)* (IEEE, 2020), pp. 1–4.

[161] S. Kuehl, R. Koch, W. Schairer, B. Spinnler, and S. Pachnicke, "Optimized bandwidth variable transponder configuration in elastic optical networks using reinforcement learning," in *Photonic Networks; 22th ITG Symposium* (VDE, 2021), pp. 1–4.

[162] Z. Xie, Y.-H. Huang, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen, and N. Corporation, "RouteNet: Routability prediction for mixed-size designs using convolutional neural network," in *Proceedings of the International Conference on Computer-Aided Design, ICCAD '18* (Association for Computing Machinery, 2018).

[163] K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos-Aparicio, "Unveiling the potential of graph neural networks for network modeling and optimization in SDN," in *Proceedings of the ACM Symposium on SDN Research (SOSR)* (ACM, 2019), pp. 140–151.

[164] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," IEEE Access **6**, 52138–52160 (2018).

[165] D. Staessens *et al.*, "Enabling high availability over multiple optical networks," IEEE Commun. Mag. **46**, 120–126 (2008).

[166] S. Sharma, J. Henderson, and J. Ghosh, "CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models," in *Proceedings of the 2020 AAAI/ACMConference on AI, Ethics, and Society* (ACM, 2020), pp. 166–172.

[167] Wide-area optical backbone performance, https://www.microsoft.com/en-us/research/project/microsofts-wide-area-optical-backbone/, 2021; accessed 08 01 2021.

[168] D. Ives *et al.* (2020). "Distributed abstraction and verification of an installed optical fiber network," Dataset. https://doi.org/10.17863/CAM.55075.

[169] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," ACM Comput. Surv. **54**, 37 (2021).

[170] J. Zhou, X. Yang, L. Sun, C. Han, and F. Xiao, "Network traffic prediction method based on improved echo state network," IEEE Access **6**, 70625–70632 (2018).

[171] J. L. Elman, "Finding structure in time," Cognit. Sci. **14**, 179–211 (1990).