photonics

Review

Machine Learning in Short-Reach Optical Systems: A Comprehensive Survey

Chen Shao [1,*], Elias Giacoumidis [2], Syed Moktacim Billah [1], Shi Li [2], Jialei Li [2], Prashasti Sahu [3], André Richter [2], Michael Faerber [1] and Tobias Kaefer [1]

[1] Department of Economics and Management, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany; ge59wah@mytum.de (S.M.B.); michael.faerber@kit.edu (M.F.); tobias.kaefer@kit.edu (T.K.)

[2] VPIphotonics GmbH, Hallerstraße 6, 10587 Berlin, Germany; elias.giacoumidis@vpiphotonics.com (E.G.); shi.li@vpiphotonics.com (S.L.); jia.li9@kit.edu (J.L.); andre.richter@vpiphotonics.com (A.R.)

[3] Electronic and Information Engineering, Technical University of Chemnitz, Str. der Nationen 62, 09111 Chemnitz, Germany; prashastisahu13@gmail.com

* Correspondence: chen.shao2@kit.edu

Abstract: Recently, extensive research has been conducted to explore the utilization of machine learning (ML) algorithms in various direct-detected and (self)-coherent short-reach communication applications. These applications encompass a wide range of tasks, including bandwidth request prediction, signal quality monitoring, fault detection, traffic prediction, and digital signal processing (DSP)-based equalization. As a versatile approach, ML demonstrates the ability to address stochastic phenomena in optical systems networks where deterministic methods may fall short. However, when it comes to DSP equalization algorithms such as feed-forward/decision-feedback equalizers (FFEs/DFEs) and Volterra-based nonlinear equalizers, their performance improvements are often marginal, and their complexity is prohibitively high, especially in cost-sensitive short-reach communications scenarios such as passive optical networks (PONs). Time-series ML models offer distinct advantages over frequency-domain models in specific contexts. They excel in capturing temporal dependencies, handling irregular or nonlinear patterns effectively, and accommodating variable time intervals. Within this survey, we outline the application of ML techniques in short-reach communications, specifically emphasizing their utilization in high-bandwidth demanding PONs. We introduce a novel taxonomy for time-series methods employed in ML signal processing, providing a structured classification framework. Our taxonomy categorizes current time-series methods into four distinct groups: traditional methods, Fourier convolution-based methods, transformer-based models, and time-series convolutional networks. Finally, we highlight prospective research directions within this rapidly evolving field and outline specific solutions to mitigate the complexity associated with hardware implementations. We aim to pave the way for more practical and efficient deployment of ML approaches in short-reach optical communication systems by addressing complexity concerns.

Keywords: machine learning; optical communications; passive optical network; equalization; optical performance monitoring; modulation format identification; bit-error ratio; optical signal-to-noise ratio; nonlinearities

check for updates

Citation: Shao, C.; Giacoumidis, E.; Billah, S.M.; Li, S.; Li, J.; Sahu, P.; Richter, A.; Faerber, M.; Kaefer, T. Machine Learning in Short-Reach Optical Systems: A Comprehensive Survey. *Photonics* **2024**, *11*, 613. https://doi.org/10.3390/photonics11070613

Received: 12 March 2024
Revised: 28 May 2024
Accepted: 29 May 2024
Published: 28 June 2024

1. Introduction

Short-reach optical transmission systems have gained substantial attraction owing to their remarkable attributes of high bandwidth and low latency [1]. In the evolving landscape of communication technologies, short-reach optical communication has emerged as an essential domain, driven by the increasing demand for high-speed data transfer in applications such as inter-data centers [2], access/local area networks, and industrial automation [3]. This increasing demand requires efficient, low-latency communication systems tailored to short-reach scenarios, typically up to 100 km. While long-haul, optical communication has been immersive in data transmission, its applicability encounters challenges when adapting to the constraints of shorter distances. This is mainly due to physical and technical limitations that prevent its seamless integration into existing networking environments characterized by the need for energy-efficient and cost-effective data transmission over limited distances. Passive optical networks (PONs) utilize passive optical splitters and combiners, which are less expensive than the active components required in traditional point-to-point fiber networks. This makes PONs a cost-effective fiber-optic solution.

Since PONs rely on passive optical splitters, they inherently introduce power losses, limiting the overall power budget and the number of users that can be supported on a single PON. In addition, effects caused by the fiber, such as chromatic dispersion (CD) and nonlinearity can limit the PON-reach [4], especially when intensity-modulated and direct-detected (IMDD) high baud-rate signals are considered [5].

Ongoing research endeavors are dedicated to advancing optical detection schemes to overcome these limitations and increase the signal bit rate in both short-reach and long-haul optical communication networks [6]. For instance, the regeneration of coherent optical systems in the last decade has been a major breakthrough, as they have gone beyond just using intensity-only modulation [7]. Coherent systems employ external modulators to employ complex baseband signals to the optical field. The optical coherent receiver, equipped with phase diversity, linearly recovers signals and compensates for fiber impairments through digital signal processing (DSP) [8]. Coherent technology enables the transmission of advanced modulation formats and polarization multiplexing to increase the signal bit rate significantly. Additionally, coherent optical systems enable dense wavelength division multiplexing (DWDM) and super-channels, which push long-distance optical networks into the multi-terabit per second capacity range [9].

Except for traditional homodyne-coherent technology, coherent communication strategies include diverse techniques, such as phase detection through heterodyne detection. While this approach has its merits [10], a notably favored incoherent approach such as IMDD is practically preferred due to its inherent simplicity and cost-effectiveness in short-reach communications [11,12].

In contrast to coherent transmission, IMDD operates by encoding information into the intensity of the optical signal, with the modulation signal being real-valued and positive [12]. The implementation of IMDD eliminates the need for complex optical components and local oscillators, reducing hardware complexity. Additionally, IMDD systems are less susceptible to phase noise and polarization-related issues, making them robust and practical for scenarios where cost efficiency and simplicity are paramount [12]. Furthermore, practical considerations like operation and safety can limit the highest and average values of the modulated signal in IMDD systems. These restrictions give IMDD systems specific characteristics in how they function [13]. Various models, such as the Poisson channel, square-root Gaussian channel, and Gaussian channel with input-dependent noise, among others, exist to rapidly assess and characterize IMDD systems [14–16]. In contrast to conventional methodologies that depend on analog components and processing [17], IMDD can potentially integrate machine learning (ML) algorithms at the receiver DSP if required [18], providing a flexible and adaptable solution for enhancing the transmission performance. According to [19], the combination of ML and DSP techniques allows IMDD systems to dynamically adapt and optimize signal parameters. This addresses impairments and variations in real time without needing complex hardware adjustments. This approach represents a significant benefit, as it not only reduces the costs associated with complex hardware setups in short-reach systems, but also highlights the effectiveness of intelligent signal processing [18,20,21].

In this survey, we examine the significant progress made in short-distance optical communications research over the past decade. First, we summarize several key research areas (Section 2). Afterwards, we focus on the equalization problem, introducing benchmark DSP methods (Section 3) and ML algorithms (Section 4). Then, we categorize recent sequence models in the ML field (Section 5), dividing them into convolution-based, transformer-based, and Fourier-based neural networks. We explore the advantages, disadvantages, and complexities of each method in addressing the equalization problem. In the final section, we provide an overview of the model compression field, outlining two approaches to compress models. We see these approaches as potential solutions for addressing hardware complexity concerns.

The primary contribution of this survey is to summarize the existing research on ML implementations for short-reach optical communications across a range of applications. Specifically, our contributions are the following:

1. We review existing deep learning (DL) models, providing a comprehensive understanding of their principles, characteristics, and hypothesis classes. This facilitates an in-depth exploration for researchers seeking supervised neural-network-based ML models suitable for their specific applications.
2. We highlight the features and complexities of these models, elucidating recent developments in the field of DL. This information is valuable for researchers interested in delving deeper into research and staying abreast of current advancements.
3. We discuss the current limitations and research gaps in the ongoing development of DL, addressing the challenges posed by these factors in real-world applications. Furthermore, we provide constructive insights regarding the selection of models and potential future directions.
4. Given the challenge of high hardware complexity, we introduce model compression as a potential solution from the DL field. We present existing works that employ this approach within the optical communication field, aiming to inspire more researchers to pursue research in this domain.

2. Applications in Short-Reach Systems

After systematically organizing recent literature in the past few years, we have categorized ML-based research for short-reach optical systems into four classes based on application tasks: Bandwidth Request and Prediction, Subcarrier Allocation, Equalization, and Fault Detection. We clarify the physical and mathematical aspects of their respective tasks, enumerate several recent works, and provide a summary of current advancements.

Bandwidth Request and Prediction: It aims to leverage network information to predict future bandwidth availability and enable its utilization by related applications. In mathematical terms, the real-time bandwidth forecast at a specific time (t) involves estimating the available bandwidth that will be accessible in the immediate future ($t + \tau$) [22]. One proposed method, known as predictive-dynamic bandwidth allocation (P-DBA), utilizes this concept to predict high-priority traffic during waiting periods, resulting in reduced latency and packet loss rates within a Gigabit PON (GPON) [22]. Another approach demonstrated in [23] leverages the k-nearest neighbor algorithm to predict additional bandwidth requirements for each optical network unit (ONU) in a PON. This adaptive learning-based approach dynamically adjusts the k value based on real-time traffic conditions, showcasing the adaptability of ML in optimizing bandwidth allocation [23]. Artificial neural networks (ANNs) have also shown promise in achieving flexible bandwidth allocations across various application scenarios, particularly emphasizing low-latency objectives [24,25]. For example, feed-forward-based ANNs, explored in [26], are utilized to predict packet arrivals in time-division multiple access (TDMA) ONUs, effectively reducing additional DBA processing delays [26]. Furthermore, Xgboost [27] is employed to predict bandwidth requests for ONUs in Ethernet PON (EPON), optimizing bandwidth utilization across polling periods. This study introduced a dynamic wavelength and bandwidth assignment scheme for time and WDM (TWDM) PONs, incorporating regression techniques for efficient resource allocation [28]. Recent studies show that ML approaches are versatile in addressing challenges related to predicting and managing bandwidth needs. This paves the way for developing more adaptive and efficient short-reach optical communication systems in the near future [22–26,28].

Subcarrier Allocation: The optimization of bandwidth allocation for enhanced spectral efficiency has led to increased interest in subcarrier allocation for PONs. This approach involves mathematically formulating the allocation problem as an integer linear programming (ILP) task, which includes tasks such as optimizing wavelength configurations, assigning subcarriers to transmitters, and minimizing lost traffic and energy costs. To address this challenge, deep reinforcement learning has emerged as a promising technique that enables dynamic subcarrier sharing among ONUs, facilitating efficient DBA. At the medium access control (MAC) layer, the dynamic subcarrier allocation (DSA) algorithm schedules ONU upstream transmissions by considering instantaneous bandwidth requirements and existing traffic conditions [29]. This showcases the adaptability of ML in resource scheduling. Several studies focus on algorithm-level cost reduction and two-dimensional resource scheduling for orthogonal frequency-division multiplexing (OFDM)-PONs including [29–31]. These DSA algorithms address challenges related to latency, throughput, and energy efficiency, highlighting the versatility of ML in enhancing subcarrier allocation strategies [32]. Moreover, the integration of traffic prediction technology and fair-aware DSA algorithms, as proposed in [32,33], further enhances the performance of subcarrier allocation in short-reach optical communication systems. These advancements improve the efficiency and adaptability of subcarrier allocation by applying ML methodologies [34].

Power Budget Limitations: The electric power budgeting issue is about predicting future energy consumption using historical data on power usage and related environmental factors like weather, user behavior, and equipment efficiency. The goal is to forecast power consumption for upcoming time periods. However, the development of large-scale, systematic ML models for this task is limited by the lack of publicly available datasets. Recent research has provided a basic process for constructing the necessary data and has also presented baseline ML models as a starting point. Specifically, the data construction process involves compiling and organizing relevant datasets, including time-series power consumption data, weather information, occupancy patterns, and equipment performance metrics. This standardized data can then be used to develop and test ML models for power consumption forecasting. For instance, the recent work in [35] has introduced baseline ML models that demonstrate the feasibility of using these techniques to predict future power consumption, despite the constraints posed by the scarcity of publicly accessible datasets.

Equalization: The objective of this task is to minimize fiber-induced distortions by employing post-processing techniques that compensate for linear effects, such as CD. Mathematically, the equalizer optimizes the function $f(x)$ to ensure that the equalized output sequence y closely approximates the input signal. Performance evaluation primarily relies on the bit-error ratio (BER). In PON systems, using shallow-based DL models for post-equalizers has shown potential in addressing nonlinear distortions for both IMDD and coherent signals. This is especially useful in scenarios with modulator nonlinearities or high-launched optical power to meet tight power budgets [7]. As the fiber-induced nonlinear effects are increasing in the latter case, in single-channel coherent PONs, this results in self-phase modulation (SPM). In multi-channel PONs, the increased nonlinear effects result in cross-phase modulation (XPM) and four-wave mixing (FWM). In IMDD PONs, low-complexity artificial neural network (ANN)-based equalizers have demonstrated performance comparable to Volterra-based equalizers in pulse amplitude modulation with four levels (PAM4) systems [36]. While post-equalization techniques have proven effective, the computational complexity at the ONU receiver is a challenge. To address this, strategies for centralized pre-equalization at the transmitter side have been proposed. Examples include memory polynomial-based pre-equalizers [36] and trained neural-network-based pre-equalizers [37]. These methods enhance equalization effectiveness while keeping the ONU receiver simple.

Fault Detection: Short-reach optical communication systems, including PONs, are susceptible to failures such as fiber cuts, equipment failures, power outages, natural disasters, and ONU transceiver malfunctions [38]. Service disruptions can result in significant financial losses for service providers. Identifying faulty ONUs presents challenges, especially when nearly equidistant branch terminations lead to overlapping reflections, making it difficult to pinpoint the exact defective branch [38]. Conventional monitoring approaches become less reliable as PON systems grow in complexity. Recent advancements in ML-enabled proactive fault monitoring offer promising solutions to ensure stable network operation. ML-based fault prediction algorithms utilize past network fault data to discover underlying patterns and similarities. By doing so, these algorithms enhance the detection of optical network problems and facilitate proactive repairs, thereby preventing potential issues from occurring. Several research papers propose using ML algorithms for monitoring management in optical networks. Notably, technologies like random forest and ANN algorithms have been employed to continuously monitor the BER, predict network component failures, and assess fault severity [39]. Wang et al. [40] introduced a hybrid approach combining double exponential smoothing and support vector machines for equipment failure prediction in software-defined metropolitan area networks. Bayesian-network-based models have also been developed for diagnosing PON faults [40].

3. DSP for Signal Equalization in Communication Systems

In this section, we provide an overview of conventional signal equalization techniques, ranging from basic zero-forcing equalization to more advanced approaches such as feed-forward equalizers (FFEs), decision-feedback equalizers (DFEs), Viterbi and Volterra equalizers, and adaptive equalizers. We discuss the advantages and limitations of these techniques, comparing the performance of ML models. Table 1 provides the complexity analysis for each method.

Zero Forcing: It is a linear equalizer (LE) derived by minimizing inter-symbol interference (ISI). A study in [41] has established the analytical foundation for optimal zero-forcing and minimum mean-squared error (MSE) equalization in channels with additive white noise and specified frequency response. The study demonstrates that an optimal LE can be implemented as a cascade of filters, with taps spaced at symbol intervals. However, when the channel effect exhibits deep frequency response "valleys", equalization will yield poor performance due to noise enhancement.

Feed-Forward Equalizer: The FFE [42] mitigates ISI in communication channels by processing the received signal forwardly without feedback. Its simplicity makes it suitable for systems where feedback is unstable or challenging for implementation.

Decision-Feedback Equalizer: Due to the noise enhancement, the DFE is designed to reduce ISI by subtracting already-known symbols. In this way, ISI from already detected symbols is eliminated. Adaptation of the forward and feedback filters of DFEs follow the same pattern as for LEs [43]. The disadvantage is that it could potentially lead to accumulated errors from feeding back incorrect detection decisions

Viterbi Equalizer: The Viterbi equalizer seeks to estimate the most likely sequence of transmitted symbols, given the received sequence. By constructing a trellis diagram where nodes represent possible transmitted symbols and transitions denote potential channel transitions, the Viterbi algorithm dynamically optimizes path metrics to identify the most probable sequence. This process involves state transition probabilities and precise calculations to mitigate the impact of channel impairments. Mathematically, the Viterbi equalizer applies the Viterbi algorithm, which belongs to the dynamic programming algorithm for finding the most likely sequence of hidden states in a hidden Markov model. The time complexity of the Viterbi equalizer is determined by the Viterbi algorithm, which depends on the length of the input sequence and the number of states, making it $O(T \cdot N^2)$, where T denotes the length of the sequence and N refers to the number of hidden states [44].

Volterra Equalizer: This is a nonlinear equalizer used in optical communication systems to compensate for nonlinear distortions introduced by the fiber channel [45]. In PAM4 systems, severe ISI can be introduced due to the imperfect bandwidth of optical and electrical components. The main bandwidth bottleneck in IMDD systems comes from the transmitter side, as the achievable bandwidth of receiver-side devices is typically twice as high as the bandwidth of transmitter-side devices. In such scenarios, the Volterra equalizer can be effectively employed to address both a potential nonlinearity from the transmitter and the bandwidth limitations of the optical components. The higher-order Volterra kernels can model the frequency-dependent distortion and nonlinear effects caused by the limited transmitter bandwidth and nonlinear devices, such as Mach-Zehnder modulators. The Volterra equalizer is based on the Volterra series expansion, which allows for the modeling of nonlinear systems. The key idea is to use a set of nonlinear filters, known as Volterra kernels, to capture the nonlinear characteristics of the channel. The structure of a Volterra equalizer consists of multiple stages, each representing a different order of nonlinearity. The first stage corresponds to the linear equalizer, which performs initial equalization to address linear distortions. Subsequent stages of the Volterra equalizer capture and compensate for higher-order nonlinear distortions. These stages involve nonlinear filters that take multiple past symbols as inputs and produce outputs based on their interaction. The number of stages and the complexity of the Volterra equalizer depend on the specific system requirements and the level of nonlinear distortions present. The coefficients of the Volterra kernels are typically adapted or optimized using algorithms such as the least mean squares (LMS) or recursive least squares (RLS) algorithms. These algorithms iteratively adjust the coefficients based on the error between the equalized signal and the desired signal, aiming to minimize the distortion and improve the overall system performance.

Adaptive Filtering: Adaptive filtering [46] is used in communication systems where channel characteristics vary over time. The mathematical interpretation involves using an adaptive algorithm that iteratively modifies the filter parameters to minimize the error signal between the desired output and the actual output, enabling the filter to adapt to changing input conditions. The actual convergence time and the total time complexity over multiple iterations depend on the convergence behavior of the specific algorithm and its sensitivity to the input data. Assuming t taps, the total time complexity for updating all coefficients is $O(t)$. FFEs and DFEs are regarded as adaptive filtering versions designed explicitly for short-reach communications.

Table 1. Complexity analysis for DFE, FFE, LE, Adaptive Filtering, and Viterbi algorithms. t refers to the number of the taps. N in Viterbi denotes the number of the hidden states.

Models	DFE	FFE	LE	Adaptive Filtering	Viterbi
Train	$O(t)$	$O(t)$	$O(t)$	$O(t)$	$O(t \cdot N^2)$
Inference	$O(t)$	$O(t)$	$O(t)$	$O(t)$	$O(t \cdot N^2)$

4. Traditional Sequential ML Methods

With the increasing demand for higher data transmission rates and the limitations of traditional prediction methods reaching their practical limits in terms of accuracy, the need for algorithms with high precision, reliability, and low complexity has become urgent. In this section, we introduce new DL-based models to address this challenge. We overview relevant research studies, providing a chronological exploration of key sequential models, namely, recurrent neural networks (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), and convolutional neural networks (CNN). The key architectural parts of DL models are explained, with clear examples showing how they work, how they are used, and how complex they are.

In 2018, Karanov et al. [47] introduced an end-to-end deep neural network system for optical communications, encompassing the entire chain of a transmitter, receiver, and channel model. This research showed that transceiver optimization can be achieved in a complete, end-to-end way. Owing to the sequential structure of communication systems, sequential models, including LSTM networks [48], RNNs [49], and GRUs [50] have been extensively employed. They are considered as baseline algorithms in order to generate more advanced and efficient algorithms.

RNN: Originally designed for machine translation in natural language processing, this model is based on the Markov assumption about the hidden state and output sequence:

the output sequence depends only on the current potential state h_t. The potential state depends on the previous moment's latent h_{t-1} and input variables x_{t-1} rather than on the historical data $x_{(t-1,\ldots,0)}, h_{(t-1,\ldots,0)}$. Renowned for their adaptability in handling variable-length sequences and preserving state information across elements, these models find valuable applications in diverse communication fields [3]. In recent work, they have shown promising results in equalization compared to benchmark methods based on Volterra and Viterbi equalizers in two-dimensional eight-level PAM (2D-PAM8) links [51].

Despite its great equalization performance, this model suffers from exploding gradient issues caused by the direct gradient flow of multiple layers [52]. In such networks, the backpropagation of the gradient is performed by accumulating the gradient matrix. This can cause the gradient to grow exponentially if the eigenvalues of the gradient matrix are greater than 1, making the training process very difficult to converge. Conversely, when the eigenvalues of the gradient matrix are less than 1, the gradient will decrease over time until it vanishes completely, causing the parameters to stop updating [53].

LSTM: The LSTM architecture can assist in overcoming this issue by extending the hidden state to a cell state, which is built using a gating mechanism. This mechanism has input, forget, and output gates that help control the flow of information [54]. LSTM models have additional internal states beyond just the hidden state. This allows them to learn a weight matrix that can better preserve useful information in the hidden state. The input gate decides what new information from the current input to be stored in the cell state. The forget gate decides what memories from the previous cell state to keep or discard. The output gate controls what information gets passed to the next cell state. This gating mechanism provides the ability to effectively hold onto relevant details from long sequences while filtering out irrelevant information. This makes it easier to learn dependencies between distant parts of the input. As a result, LSTMs have been widely used in short-range communication tasks that require capturing complex long-term relationships in the data [55].

GRU: This architecture simplifies the gating mechanism used in LSTM models. It has an update gate and a reset gate, instead of the three gates in LSTM [56]. The update gate determines what relevant information to retain from the previous state and the current input. The reset gate controls what data to discard. It is useful in scenarios where the temporal dependencies and relationships between adjacent symbols in a sequence are important. For example, in short-range communication systems, GRUs can help mitigate signal distortions caused by CD and nonlinearities [57]. Recent research in 120 Gb/s coherent 64-quadrature amplitude-modulated optical systems for transmission at 375 km has shown that using a bi-directional GRU as a nonlinear equalizer can help improve the quality factor (Q-factor) beyond the 8.52 dB limit (8.52 dB estimated from $Q = 20\log_{10}(\sqrt{2}\mathrm{erfc}^{-1}(2\mathrm{BER}))$ [58], typically required for hard-decision forward error correction (HD-FEC).

CNN: CNNs are not technically considered sequential models. However, they are widely used across many different domains. This is because of its important advantages, such as high parameter efficiency, weight-sharing mechanism, and plug-and-play characteristics [59]. CNNs use a convolutional kernel to scan the input signal in a specific dimension, capturing temporal features that are important for the task at hand. This convolutional layer is typically followed by a pooling layer and a nonlinear activation function. The pooling layer reduces redundancy, while the activation function introduces nonlinearity. The convolutional kernel is designed to extract features that closely match the input signal. Afterwards, backpropagation is used to optimize the weights of the network. This allows the CNN to learn and enhance the features that are most relevant for the target task. The weights in the network's weight matrix are updated through backpropagation to amplify the important features needed for effective performance on the given ML problem. Furthermore, it has been observed that using multiple layers of small convolutional kernels is often more efficient than using large kernels. This approach, known as the inception architecture, was first introduced in the GoogleNet model [60]. Two commonly used blocks in CNN are the inception module and the inception reduction module, which extract temporal dependencies of different scales by employing a concatenation of 1 × 1, 3 × 3, and 5 × 5 convolutional kernels. In addition, It also uses a special type of convolutional kernel with a size of 1 × 1. This 1 × 1 convolution serves a unique purpose: it helps to reduce the number of feature map channels or dimensions. It is commonly used between two regular convolution layers or at the output layer of the network.

Summary: In this section, we have introduced the most common building blocks used in ML models for short-reach optical communication systems. These fundamental components are still widely used in current approaches. To summarize the complexity of the models discussed earlier, we have provided a table (Table 2) that outlines the complexity analysis for each of the models. In this complexity analysis, we have focused solely on the computation required per batched sample, without considering the choice of hyperparameters like the number of epochs or batch size. This provides a compact overview of the computational demands of each model on a per-sample basis.

Table 2. Complexity of DNN, GRU, LSTM, RNN, and CNN. t refers to the number of taps. n_s, n_o, n_h, and d denote input, output, hidden neuron, and depth of the DNN, respectively.

Models	DNN	GRU	LSTM	RNN	CNN
Train	$O(d-1)n^2)$	$O((3n_h^2+6n_h)n_s)$	$O(4n_h^2+7n_h)n_s$	$O(n_h^2 n_s)$	$O(n_o)$
Inference	$O(d-1)n^2)$	$O((3n_h^2+6n_h)n_s)$	$O(4n_h^2+7n_h)n_s$	$O(n_h^2 n_s)$	$O(n_o)$

5. Advanced Sequential ML Methods

In Section 4, we introduced traditional sequential models, such as RNN, LSTM, GRU, and CNN. The key question we aim to answer in this section is how to effectively incorporate the unique characteristics of time-series data into the modeling process and leverage the temporal convolution model to mitigate channel distortion. Compared to other DL models like transformers and Fourier-based neural networks, convolutional models exhibit better generalization performance. Convolutional models are also more robust to changes in their parameter values when applied to new datasets, unlike the other models that require careful parameter initialization and hyperparameter tuning when used on new data [61].

This section starts with channel modeling, encompassing four distinct noise models. We derive the characteristics and capabilities required for the algorithm based on these models. Subsequently, we provide a detailed exposition of the architectures and fundamental assumptions underlying three models: Frequency-Calibrated Sampled-Interaction Neural Network (FC-SCINet) [62], Light Time-Series (LightTS) [63], and DLinear [64].

5.1. Distortion Model

The main limiting factor for the equalization task in a short-reach/PON system is ISI as a result of CD, sampling error (jitter), frequency shift (chirp), and Kerr-induced nonlinearity [47]. In this section, we will focus on the effects of CD, jitter, and chirp, as these are the dominant distortion mechanisms in short-reach PAM-based systems. The impact of Kerr-nonlinearities is limited in single-channel PONs due to the relatively short fiber lengths and low optical powers involved.

CD in an optical communication system is caused by different phase velocities with respect to frequency. It fundamentally constitutes a linear transformation, and its mathematical representation involves a differential equation that considers spatial position and time, which can be presented as

$$\frac{\partial A}{\partial z} = -j\frac{\beta_2}{2}\frac{\partial^2 A}{\partial t^2} \tag{1}$$

where A denotes the amplitude of the complex signal; t denotes time; z is the spatial position along the fiber, where the pulse pattern propagates [47]; and β_2 is the dispersion coefficient. Following the Fourier transformation, we have

$$D(z,w) = \exp(j\frac{\beta_2}{2}\omega^2 z) \tag{2}$$

w is the angular frequency. In the time domain, it is primarily manifested by significant attenuation in the high-frequency components and rapidly changing components.

Jitter is caused by fluctuations in sampling time. It presents itself as signal distortion, exhibiting a comparable impact to superimposed interference signals that adhere to the Gaussian distribution. Timing jitter can be described as

$$\begin{aligned} y_*(t) &= y(t)\sum_{n=-\infty}^{+\infty}\delta(t - n * t_A - \tau) \\ &= y(n * t_A + \tau) \end{aligned} \tag{3}$$

where τ is the timing sampling error, where the correctly sampled value is $y(n * t_A)$. This sampling error can be quantitively estimated as follows:

$$|y(n * t_A + \tau) - y(n * t_A)| \leq M_1|\tau| \tag{4}$$

where M_1 is the first moment of the band-limited spectrum of Fourier transformation of original signal $Y(f)$, can be simply written as:

$$|\frac{\partial y(t)}{\partial t}| \leq \sum_{-f_g}^{f_g}|2\pi f||Y(f)| = M_1 \tag{5}$$

Jitter refers to high-frequency fluctuations in the amplitude of a signal. This high-frequency perturbation can have a significant impact on neural networks that rely on low-frequency signals.

In conclusion, the error can be estimated as $|y(t_n) - y(n\tau_a)| \leq M_1|t_n - n\tau_a| = M_1|\tau_n|$. The error $|e_n|$ is bounded by $M_1 \cdot |\tau_n|$ for a given n. The value of e_n depends solely on τ_n. Assuming that the timing error τ_n follows a statistical nature with $E\{\tau_n\} = 0$ and $E\{\tau_n^2\} = \sigma_\tau^2$, it follows that the amplitude errors e_n are statistically independent. Consequently, the error variance is then given by $E\{e_n^2\} \leq M_1^2\sigma_\tau^2$. For more details, please refer to [65].

Chirp is a signal whose frequency varies with time. Mathematically, it can be described as follows,

$$s(t) = a(t) \cdot \exp[j(\omega_0 \cdot t + \theta(t))] \tag{6}$$

The frequency spectrum of this waveform is obtained as

$$S(\omega) = \int_{-\infty}^{\infty} a(t) \cdot \exp[j(\omega_0 t + \theta(t))] \cdot \exp(-j\omega t)\,dt \tag{7}$$

Simplifying further:

$$S(\omega) = \int_{-\infty}^{\infty} a(t) \cdot \exp[j\{(\omega_0 - \omega)t + \theta(t)\}]\,dt \tag{8}$$

In summary, all types of effects encountered in equalization issues, except for jitter, involve concurrent alterations in both the time and frequency domains. It is noteworthy that such changes are not statistically independent. Consequently, no single effect can be eliminated through straightforward nonlinear operations in a single domain.

5.2. Temporal Convolution Neural Network

DL-based equalizers fundamentally capture domain-specific nonlinear disturbances by employing linear transformations and nonlinear activation functions. Within the temporal CNN, the core modules comprise interval, continuous, and interaction sampling modules, alongside convolutional neurons and linear layers, as depicted in Figure 1. Each of these modules offers practical flexibility for hardware implementation, due to their computational efficiency. Subsequently, we delve into three of the most efficient convolution-based sampling networks.

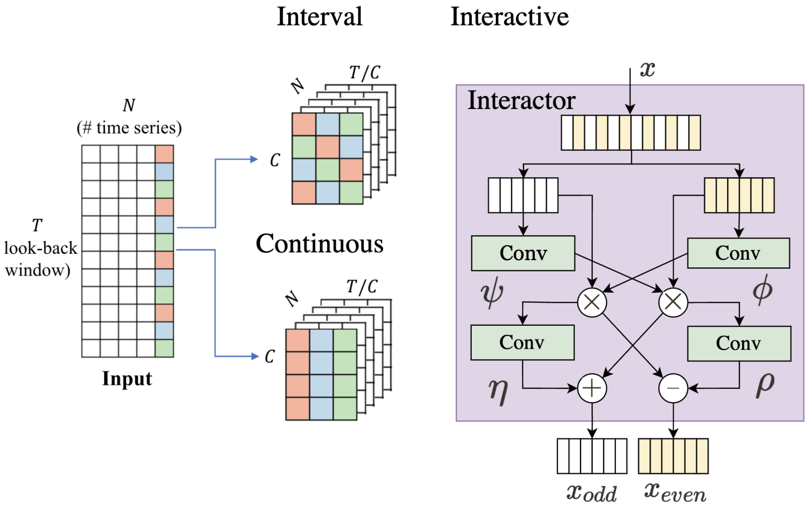

Figure 1. The overview of all sampling modules in temporal convolution networks is modified from [62,63], namely, interval sampling and continuous sampling in LightTS [63], and interactive sampling in [61,62].

FC-SCINet: This novel approach introduces an improved series decomposition technique as a spectrum correction module. In conjunction with the interaction sampling module, it has proven to be a robust tool for mitigating CD and addressing various real-world channel effects [62].

Decomp: In the case of FC-SCINet, it utilizes a moving averaging filter with kernel size w_1 to extract low-frequency signals from the input. Additionally, high-frequency signals are obtained by calculating the residuals between the original and low-frequency signals. The final output signal is generated through a weighted linear combination of these two components, which is $\hat{x}$ defined as Equation (9).

$$\hat{x} = W_s^T x_s + W_f^T x_f \tag{9}$$

The complexity of this module is $O(k)$, where k is the size of the kernel and is independent of the input sequence length.

However, as demonstrated in the empirical study in [62], the performance of FC-SCINet in mitigating CD remains strong. Moreover, the plug-and-play nature, low complexity, and interpretability of FC-SCINet make it highly flexible for seamless integration with various other algorithms. The DLinear architecture is another impressively low-complexity yet high-performance design.

SCIBlock: The SCIBlock, is a key component of FC-SCINet, because it can iteratively decompose a signal into sub-sequences at various scales while incorporating nonlinear transformations between adjacent layers. In contrast, the decomp block is restricted to enhancing fixed signal components and is limited to a single scale. From a mathematical perspective, the SCIBlock applies a hierarchical structure by systematically downsampling the input sequence into even-positioned and odd-positioned samples, denoted as x_{even} and x_{odd}. Following the convolutional layer, the sub-sequences in adjacent layers are iteratively multiplied together utilizing exponential and multiplication operations, as shown in Equations (10) and (11).

$$x^{s}_{even} = x_{even} \odot \exp(\psi(x_{odd})), \quad x^{s}_{odd} = x_{odd} \odot \exp(\phi(x_{even})) \tag{10}$$

$$x'_{odd} = x^{s}_{even} + \exp(\eta(x^{s}_{odd})) \quad x'_{even} = x^{s}_{odd} - \exp(\rho(x^{s}_{even})) \tag{11}$$

Here, $\odot$ represents an element-wise product, and ψ, ϕ, η, and ρ are independent 1D convolutional layers. The intermediate outputs can be presented as x^{s}_{even}, x^{s}_{odd}, x'_{even}, and x'_{odd}. Upon completion of the processing, the resulting sub-sequences are then reassembled and aligned back to their original positions within the original signal. Ultimately, all the sub-series are concatenated based on their original index in the raw sequence, as illustrated in Figure 1. To sum up, FC-SCINet is a framework capable of efficiently learning local-dependent patterns. Its distinctive feature lies in performing interactive learning on sub-sequences with odd-even positions after odd-even sampling, allowing for a larger receptive field under the premise of using the same convolutional kernel.

DLinear: As previously mentioned in FC-SCINet, while the concatenation of the decomp module may not offer optimal equalization, it has exhibited great performance in real-world datasets. Therefore, we will provide a brief introduction to this module: It first decomposes a raw data input into a low $\mathbf{x_s}$ and high-frequency $\mathbf{x_f}$ signal. $\mathbf{x_s}$ is extracted by a moving average kernel. It is equivalent to filtering the signal using a *sinc* function in the frequency domain. These two components are added in a linear combination form, expressed by W_s, W_f. The operation above is presented in Equation (12).

$$\mathbf{x_s} = \text{AvgPool}(\mathbf{x}) \quad \mathbf{x_f} = \mathbf{x} - \mathbf{x_s} \quad \mathbf{x}' = W_s\mathbf{x} + W_f\mathbf{x_f} \tag{12}$$

By iterative decomposition with different kernel sizes, DLinear can be extended to a deeper network. The complexity is $O(kn_s)$, where k is the number of the layer, and n_s is the length of the model input. To simplify the complexity, the weight matrix W could be replaced by the convolutional kernel.

LightTS: Both FC-SCINet and DLinear utilize only convolution and different sampling modules to capture the local and global dependencies. The LightTS architecture, detailed in [63], employs a multi-layer perceptron (MLP) structure to enhance predictive abilities.

Sampling: In contrast to SCIBlock, which samples the raw sequence using odd and even indices, LightTS introduces two generic sampling strategies: Interval Sampling and Continuous Sampling, as shown in Figure 1. Interval sampling partitions time-series data into non-overlapping sub-sequences based on fixed time intervals, as shown in Figure 1. This approach helps identify periodic patterns or regularities within the data while minimizing information loss. On the other hand, continuous sampling divides sequences into corresponding sub-sequences, extracting data points continuously throughout the time series and preserving temporal continuity. This sampling method enables the capture of patterns within the period, ensuring a more comprehensive representation of the underlying dynamics. The subsequent section presents an MLP-based architecture to extract useful features from both the downsampled sub-sequences and continuously sampled sub-sequences.

Information Exchange Block (IEBlock): The IEBlock serves as the central module in LightTS, designed to effectively process the 2D matrix resulting from continuous sampling and interval sampling. This block comprises three essential components: (1) temporal projection, which identifies temporal features following continuous sampling; (2) channel projection, which captures inter-channel information following interval sampling; and (3) the exchange block, which integrates the information from the outputs mentioned above, facilitating information fusion. All of them utilize MLP as the nonlinear behavior learning module. The design of LightTS is notably concise, employing only two sampling modules and an MLP. On certain datasets, it surpasses the performance of FC-SCINet [64].

Compared to the models mentioned earlier, the FC-SCINet model requires less structural adaptation and pre-processing when applied in practical PON applications. The FC-SCINet has been successful in recent PON-related work [62]. Different from LSTM, which offers the advantage of ensuring information flow strictly from past to future, temporal CNN goes beyond this by modeling the global dependency between input and output, while also leveraging stacked causal convolution layers. Additionally, the FC-SCINet introduces interaction modeling, enabling the explicit capture of interactions between elements within a sequence, making it a more advanced alternative to LSTM. In addition to these benefits, CNNs and SCINet offer several advantages over LSTM:

- CNNs can identify patterns regardless of their position within the input sequence. This property makes them well-suited for tasks where the position or timing of relevant features is not fixed, providing greater flexibility compared to LSTM.
- CNNs excel at capturing local patterns and extracting relevant features from the input sequence. This ability is particularly useful for tasks that require identifying and recognizing specific patterns or motifs within the data.
- Both CNNs and SCINet architectures typically have fewer parameters compared to LSTM models. This reduced parameter count can make training and inference more efficient, especially when working with limited computational resources or when dealing with large datasets.

Recent Progress: CNNs play a crucial role in current time-series prediction research and applications. This is due to their high parameter efficiency, model stability, and strong theoretical foundation (Multiscale Decomposition). The complexity of these convolutional networks mostly depends on the number of layers and the size of the convolutional kernels. Nowadays, more advanced designs like dilated convolution and inception are often combined with other modules to create complex DL models, but they are rarely used on their own. Even so, temporal CNNs still have a distinct advantage in terms of the performance-to-complexity ratio. They are also straightforward to implement in hardware.

5.3. Transformer-Based Network

Attention: The scaled dot-product attention mechanism is the key component aiming to aggregate information across different parts of the input sequence. Each input vector is transformed into three distinct vectors: Queries (Q), Keys (K), and Values (V). The process involves calculating the dot products of the queries with all keys, scaling them by the square root of the dimension, and applying a softmax function to obtain weights on the values [66].

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{13}$$

The resulting matrix of outputs is obtained through a weighted sum of the values, where the weights are determined by the softmax-processed dot products of queries and keys. This attention module allows the model to focus on relevant parts of the input sequence, capturing local dependencies during the training process. A residual connection is applied around the two sub-layers, followed by layer normalization to maintain the information flow.

Transformers use multiple attention heads to look at the input sequence from different perspectives. This allows the model to simultaneously learn and consider various views of the input data. Equations (14) and (15) represent the functionality of the multi-head attention step. $head_i$ represents the single attention head. The final result of the multi-head attention is concatenating all the attention heads.

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^o \tag{14}$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{15}$$

While the standard ("vanilla") transformer model has shown great performance on time-series data, the computational complexity of its attention mechanism makes it struggle to handle long sequences effectively. To overcome this limitation, researchers have developed various attention mechanism variants. An example is the locality-sensitive gashing (LSH) attention mechanism, which was introduced as part of the Reformer architecture [67]. The LSH attention mechanism utilizes specialized hash functions to transform queries and keys, thereby facilitating the categorization of similar items into shared hash buckets. Through sorting tokens based on their hash codes, items with similarities are grouped together, enabling the aggregation of relevant information. To enable parallel processing, the sorted sequence is divided into chunks. Subsequently, attention mechanisms are selectively applied to these chunks and their neighboring segments, allowing for focused examination of localized portions. The LSH attention mechanism uses hash coding to greatly improve the computational efficiency of the transformer compared to the original version. This helps address the challenge of processing long sequences by reducing complexity without sacrificing performance.

Decoder: The decoder consists of several stacked sub-decoders. The ground truth follows a process similar to that of the encoder, being transformed into Query Q, Key K, and Value V representations. The attention weights are calculated by comparing the Queries Q from the decoder with every value V in the encoder. This process is repeated in parallel across N sub-decoders, resulting in a final attention matrix. The attention matrix then undergoes a softmax operation, yielding probabilities for each value. In addition to the two sub-layers in each encoder layer, the decoder introduces a third sub-layer, performing multi-head attention over the encoder stack's output.

During the decoding process, the model is auto-regressive, using the previously generated outputs as additional input to generate the next output. Residual connections and layer normalization are used around each sub-layer. The self-attention sub-layer is modified to prevent positions from attending to future positions. This, along with the offset output embedding, ensures that the predictions for a position only depend on the known outputs at earlier positions in the sequence.

Attention Variants: The purpose of the attention layer is to identify connections and dependencies among the various input embeddings. This allows the model to evaluate the importance of each element in relation to the others. The attention mechanism explicitly computes the relationships between different elements in the sequence, providing insights into how information flows through the model. However, except for the computational complexity issue, the mechanism in the vanilla transformer [66] needs to be improved in terms of processing inter-dependencies and periodicity of signal data. The Autoformer model [68] introduces a new type of encoder that replaces the original encoder. This new encoder applies series decomposition and autocorrelation to detect dependencies between different parts of the input sequence, and then combines the representations of the sub-series. The series decomposition component divides the original signal into two distinct parts: the seasonal component, which captures short-term patterns, and the trend component, which captures long-term behavior. This partitioning allows for identifying and representing both the short-term and long-term characteristics present in the time-series data. Additionally, the auto-correlation mechanism utilizes the fast Fourier transform (FFT) to compute correlations between the time series and its delayed version, providing insights into how the series relates to its past values at different time lags. The combination of series decomposition and autocorrelation effectively captures and represents the underlying trend and seasonality in the time-series data.

Recent Progress: In this section, we provide a comprehensive overview of the vanilla transformer and its architecture, particularly within the domain of time series and traffic prediction. Over the years, substantial improvements have been made to enhance the transformer for accurate time-series prediction. Notably, advancements have been achieved in reducing computational complexity while improving the effectiveness of the attention mechanism [69,70]. However, recent research has introduced compact models based on multi-scale transformation [71], which surprisingly outperforms benchmark-designed models. This new development has sparked an important debate on the fundamental structure of sequence models. In the following sections, we summarize and explore this particular model in depth, providing insights into its implications. For the latest work, please refer to Table 3.

Table 3. Benchmark models.

Models		Efficient Techniques	Literature
Transformer	Attention	Sparsity inductive bias	Ref. [69] LogTrans leverages convolutional self-attention for improved accuracy with $O(L(\log L)^2)$ lower memory costs.
		Low-rank property	Ref. [70] Informer selects dominant queries based on queries and key similarities.
		Learned rotate attention (LRA)	Ref. [72] Quatformer introduces learnable period and phase information to depict intricate periodical patterns.
		Hierarchical pyramidal attention	Ref. [73] Pyraformer proposed one hierarchial attention mechanism with a binary tree following the path with linear time and memory complexity
		Frequency attention	Ref. [74] FEDformer: proposed the attention operation with Fourier transform and wavelet transform.
		Correlation attention	Ref. [68] Autoformer: the Auto-Correlation mechanism to capture sub-series similarity based on auto-correlation and seires decomposition
		Cross-dimension dependency	Ref. [75] Crossformer utilizes multiple attention matrices to capture cross-dimension dependency
	Architecture	triangular patch attention	Ref. [76] Triformer features a triangular, variable-specific patch attention with a lightweight and linear complexity
		Multi-scale framework	Ref. [71] Scaleformer iteratively refines a forecasted time series at multiple scales with shared weights
	Positional rncoding	Vallina Position	Ref. [66] cos and sin functions with a sampling rate-relevant period.
		Relative positional encoding	Ref. [77] Introduces an embedding layer that learns embedding vectors for each position index.
		Model-based learned	Ref. [69] LogSparse utilize one LSTM to learn relative position between series tokens
Fourier-NN	Time Domain	Series Decomposition	Ref. [64] DLinear performs one linear series decomposition with multiple layers
		Frequency Attention	Ref. [78] TimesNet proposes the attention mechanism related to the amplitude of the signal
	Frequency Domain	Frequency MLP	Ref. [79] FreqMLP performs MLP in frequency domain by leveraging the global view and energy compaction characteristic
TConv-NN	Sampling	Continous	Refs. [63,78] both utilize continous sampling to split original signal into windowed subseries similar to short time transformation
		Interval	Ref. [63] Interval sampling with fixed step to extract periodic feature
		Even-Odd/Multiscale	Ref. [62] proposes one iterative multiscale framework where even and odd series are interacted between layers
		Frequency Continous	Ref. [64] leverages series decomposition module in a iterative manner to decompose signal in frequency domain with *sinc* function.
		Negative sampling	Ref. [80] custom loss function is employed in an unsupervised manner, wherein distant or non-stationary subseries maximize the loss, while similar subseries minimize the loss.
	Feature module	MLP	Ref. [63] applies an MLP-based structure to both interval sampling and continuous sampling for extracting trend and detail information.
		Dilated convolutions	Ref. [81] leverages stacked dilated casual convolutions to handle spatial-temporal graph data with long-range temporal sequences

5.4. Fourier Convolution Neural Network

In the previous section, we introduced models based on convolutional kernels and subsampling as fundamental modules. Their core principle involves decomposing signals into different scales in the time domain and subsequently applying nonlinear transformations to learn salient features. However, for the majority of real-world signals, transforming them into the Fourier domain is often more efficient. This efficiency is attributed to the following factors: (1) The majority of real-world signals are bandpass or lowpass, and in the Fourier domain, their dynamic range decreases from n to $exp(-n)$; (2) The Fourier transformation is a bijective (one-to-one) transformation, which ensures energy conservation and controllable error in both the forward and inverse transformations; (3) The computational complexity of existing (fast) Fourier transform algorithms, after improvements, is $O(nlog(n))$, making it convenient for hardware implementation.

In this section, we introduce TimesNet [78], which utilizes a frequency-attention mechanism, and FreTS [79], which explicitly performs non-linear transformations in the frequency domain. For extensive models please refer to Table 3. The fundamental concept of TimesNet involves transforming the initial signal into k distinct 2D tensors instead of directly processing the original sequence. This approach empowers the model to effectively capture both intra-periodic and inter-periodic variations within these fixed windows. A variant of this model has been recently reported in [82].

TimesNet: An attention mechanism based on spectral amplitude is employed to determine the significance of signal segments at various frequencies. Simultaneously, across different temporal resolution scales, a shared convolution module is utilized to reconstruct nonlinear distortions introduced by the channel. It does not explicitly perform nonlinear transformations in the frequency domain; instead, it combines reconstructed signals at different window lengths through a linear combination. The FConvNet primarily comprises four key blocks: Component Detection, Alignment, ConvNet, and Reconstruction.

Component Detection: The identification of the k most crucial frequencies is based on the amplitude of the Fourier coefficients. Then, using only the selected components within the k frequency range, the signal is sampled using a continuous sampling method, and these sub-series are arranged into a two-dimensional tensor.

Alignment: The aligned sub-series are then fed into a convolution-based module, specifically an Inception network, to mitigate distortion caused by channel effects. The Fourier coefficients pass through a softmax function to generate attention weights, which are then multiplied by the output of each convolution module to produce the final output.

Fourier Attention: The Fourier transformation is a global operation, meaning that any changes in the signal's amplitude will cause periodic oscillations throughout the entire signal. Significant variations in the amplitude of the primary components lead to substantial fluctuations. TimesNet leverages this characteristic by using the Fourier spectrum of the nonlinearly transformed signal to determine the attention value for each component.

Reconstruction: Finally, employing a residual form, we obtain the reconstructed individual sub-components multiplied by their respective attention values, denoted as y', and add them to the input signal x to yield y.

FreTS: Time-domain-based processing is limited by information bottlenecks, as the local characteristics vary. FreTS explicitly uses frequency-domain features in its model architecture to directly mitigate distortion without manipulating the time-domain. FreTS is essentially an MLP-based network that is able to effectively learn patterns of time-series data in the frequency domain. As presented in [79], FreTS consists of two learners: a Frequency Channel Learner and a Frequency Temporal Learner. In the equalization problem, there is no actual channel dimension, but rather a stack of independent experiments. Therefore, FreTS only introduces a frequency-domain MLP.

Frequency MLP: The frequency temporal learner aims to capture temporal patterns in the frequency domain. Specifically, for a complex number input $\mathcal{H} \in \mathbb{C}^{m \times d}$, the MLP aims to optimize the weight matrix $\mathcal{W} \in \mathbb{C}^{d\times d}$ and bias $\mathcal{B} \in \mathbb{C}^{d}$ so that the final output $\mathcal{Y} \in \mathbb{C}^{m\times d}$ could approximately reconstruct the ground truth.

$$\mathcal{Y}_\ell = \sigma(\mathcal{Y}_{\ell-1}\mathcal{W}_\ell + \mathcal{B}_\ell) \tag{16}$$

$$\mathcal{Y}_0 = \mathcal{H} \tag{17}$$

The MLP in the frequency domain is equivalent to global convolutions in the time domain as detailed in [79]. An increasing number of studies have demonstrated the feasibility of DL models operating in the frequency domain. Simultaneously, the corresponding computational complexity of frequency-domain processing has decreased from $O(n)$ to $O(nlogn)$ due to the reduction in the signal's dynamic range. However, the advantages and disadvantages of networks in both the frequency and time domains remain inadequately explored. Due to space limitations, we offer a detailed categorization, along with corresponding references and keywords, in Table 3 for researchers with specific interests.

6. Model Compression

In recent years, the proliferation of large-scale ML models has significantly advanced state-of-the-art technology across various domains, ranging from natural language processing to computer vision. The surge in model complexity, often characterized by sophisticated architectures and many parameters, has driven the need for efficient hardware implementations to harness their full potential. The advent of single graphics processing units (GPUs) as a critical computational resource has been pivotal, offering a parallelized architecture suitable for accelerating the training and inference processes [83]. The significance of deploying large ML models on a single GPU lies in optimizing computational efficiency and reducing latency. Single GPU implementations facilitate parallel processing, enabling the simultaneous execution of multiple tasks and handling extensive model parameters. This enhances the speed of model training and facilitates real-time inference, a critical requirement in applications such as autonomous systems and edge computing. However, the hardware implementation of large ML models on a single GPU is challenging. The complexity of these models often exceeds the computational capacity and memory constraints of a single GPU, necessitating innovative solutions for efficient utilization [84]. Techniques such as model pruning, quantization, vector quantization, and knowledge distillation have emerged as strategies to mitigate these challenges, ensuring that even formidable models can be accommodated within the limitations of a single GPU without compromising performance. The authors in [84] examine how to use a single GPU effectively for implementing large ML models. They discuss methods that balance complexity and computational efficiency to maximize hardware utilization [84].

In addition, conducting a comprehensive performance-versus-complexity analysis is necessary to evaluate the suitability of various ANNs in short-reach optical communication systems. DL models, including CNNs, RNNs, and LSTMs, find applications in critical tasks such as equalization, fault detection, subcarrier allocation, nonlinearity compensation, and bandwidth request and allocation. The complexity of these models is a significant factor affecting their feasibility. For instance, CNNs may introduce convolutional and pooling layers, increasing model complexity. Similarly, RNNs and LSTMs, designed for sequential data, introduce recurrent connections that enhance their ability to capture temporal dependencies and contribute to increased complexity [85]. Analyzing the neural network architectures in detail, including their depth, the number of parameters, and computational demands, is crucial for understanding the trade-offs between performance and complexity. DL models often exhibit enhanced capabilities in capturing complex patterns and relationships in optical communication data. Still, their complexity may pose challenges regarding training time, computational resources, and practical implementation [86]. A thorough examination of these complexities is essential for identifying optimal models, such as choosing between a CNN for image-based tasks or an LSTM for sequential data, that balance high performance and manageable complexity, facilitating their efficient integration into short-reach optical communication systems [85]. Four prominent types, namely, the feed-forward neural network (FFNN), the radial basis function neural network (RBF-NNs), the auto-regressive RNN (AR-RNN), and the layer-RNN (L-RNN), offer distinct trade-offs in complexity and performance. Among nonlinear neural-network-based equalizers with equivalent numbers of inputs and hidden neurons, FFNN-based equalizers have the lowest computational complexity; however, AR-RNN demonstrates superior transmission performance in 50 Gb/s PAM4 systems [87].

Distillation model: Knowledge distillation, a model compression technique, transfers knowledge from complex, large-scale models or groups to more compact, feasible models suitable for real-world applications. Pioneered by Bucilua et al. in 2006 [88], knowledge distillation primarily operates on neural network architectures characterized by multifaceted structures comprising multiple layers and parameters. Knowledge distillation has been recently considered an important technique for practical DL applications such as speech recognition, image recognition, and natural language processing [89]. Deploying large deep neural network models can be especially challenging for edge devices, which are limited in memory and computational power. To address this challenge, an innovative model compression method was developed in [89], allowing transferring knowledge from larger, more complex models to train smaller, more efficient models without significant performance loss. This process, where a smaller model learns from a larger one, was formalized into the "Knowledge Distillation" framework by Hinton et al. [90]. This framework has become crucial for deploying the essential knowledge from sophisticated, large-scale models on computationally constrained edge devices.

Optimizing DL models through knowledge distillation shows great potential for advancing short-range optical communication systems. RNNs have been particularly effective at addressing nonlinear distortions [57,85]. However, the feedback loop inherent in RNN structures makes it difficult to parallelize them, preventing their use in low-complexity hardware designed for high-speed processing in optical networks [91]. Using knowledge distillation is a promising approach to enable parallelization of RNNs [85,92]. This application of knowledge distillation is set to revolutionize the implementation of RNNs, ensuring compatibility with low-complexity hardware and meeting the stringent processing requirements of high-speed optical networks [93].

Beyond just RNNs, knowledge distillation can be applied to many different ML models important for optical communications, such as CNNs, LSTMs, FFNNs, RBF-NNs, AR-RNNs, and L-RNNs [92]. These models each have their own challenges regarding complexity, adaptability, and real-time implementation. For example, using knowledge distillation in LSTMs for optical communication systems, can reduce model complexity without losing the ability to handle time-dependent patterns [92].

Another promising application of knowledge distillation is when facing challenges with limited time-series data. As "big data" impacts various fields, the scarcity of target events or high data acquisition costs can hinder ML in certain scenarios. A proposed method uses "privileged information" from partial time-series data during training to enhance long-term predictions for small datasets. Applied to optical communications, this distillation approach offers a solution to data constraints, demonstrating effectiveness on both synthetic and real-world data [94].

Vector quantization: Vector quantization (VQ) is a model compression technique targeting large-scale ML models. VQ represents complex data with a small set of prototype vectors, significantly cutting the computational load during inference. This makes VQ useful for applications that require balanced model efficiency and performance, such as when resources are limited. The VQ process involves partitioning the input space into regions, each with a representative prototype vector. During encoding, input vectors are assigned to the nearest prototype, quantizing the data. In the decoding or reconstruction phase, these prototype vectors are used to rebuild the original data.

The effectiveness of VQ relies on carefully selecting and updating the prototype vectors. The goal is to optimize the prototypes so they can effectively capture the essential information in the dataset [95]. By clustering and quantizing input vectors into a representative codebook, VQ enables encoding information in a more compact form. This is particularly beneficial in scenarios with limited data availability or high computational demands [96]. For example, VQ can be useful when applying knowledge distillation to RNNs. RNNs face challenges with parallelization due to their feedback loop structure. Using VQ in the distillation process for RNNs can help address the parallelization issue. VQ can represent the essential information from the RNN using a smaller set of prototype vectors [87,97,98]. This compression not only aids in overcoming hardware complexity but also contributes to faster processing in high-speed optical networks.

VQ uses an iterative process to improve the prototype vectors and enhance their ability to represent the data. Commonly, algorithms like k-means clustering are used for this. The prototypes are adjusted to minimize the difference between the original data and the quantized representation. This iterative refinement allows VQ to adapt to the patterns and structures in the data. This optimizes the compression capabilities of VQ while still preserving the critical information needed for training tasks [95].

Finally, VQ can be beneficial in optimizing other DL models, such as CNNs or LSTMs, by efficiently capturing essential features with a minimal set of representative vectors [99]. Exploring the use of VQ together with these models provides a promising way to improve the performance and scalability of ML applications in short-reach optical communication systems.

7. Conclusions

In this survey, we have undertaken a comprehensive examination of powerful machine learning models that exhibit the potential to achieve robust equalization in cost-sensitive short-reach optical systems, with a particular focus on PONs. Our objective has been to explore these models' capacity to operate efficiently and deliver effective computational performance. For the first time, we have classified the current models into three distinct types and conducted an extensive analysis of their core concepts, highlighting their differences, similarities, and the underlying insights they provide. Additionally, we have presented a simplified complexity analysis considering various input sizes. In the final stages of our survey, we have also investigated the potential of machine learning solutions in addressing the challenges associated with hardware implementation and complexity. We firmly believe that this survey will serve as a valuable resource, inspiring future research endeavors to develop efficient models explicitly tailored for short-reach and PON systems.

Author Contributions: Conceptualization, C.S.; data curation, J.L., C.S. and E.G.; formal analysis, M.F.; funding acquisition, T.K. and A.R.; investigation, M.F.; methodology, C.S. and E.G.; project administration, T.K.; resources, M.F. and T.K.; software, C.S. and J.L.; supervision, M.F. and T.K.; validation, C.S. and J.L.; visualization, J.L. and C.S.; writing—original draft preparation, C.S., J.L., P.S. and S.M.B.; writing—review and editing, S.L., M.F. and E.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Federal Ministry of Education and Research grant number 16KIS1228, and 16KIS1230.

Conflicts of Interest: Elias Giacoumidis, Shi Li, and Andre Richter were employed by the VPIphotonics (Germany) company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Durisi, G.; Koch, T.; Popovski, P. Toward massive, ultrareliable, and low-latency wireless communication with short packets. *Proc. IEEE* **2016**, *104*, 1711–1726. [CrossRef]
2. Kapoor, R.; Porter, G.; Tewari, M.; Voelker, G.M.; Vahdat, A. Chronos: Predictable low latency for data center applications. In Proceedings of the Third ACM Symposium on Cloud Computing 2012, San Jose, CA, USA, 14–17 October 2012; pp. 1–14. [CrossRef]

3. Xie, Y.; Wang, Y.; Kandeepan, S.; Wang, K. Machine learning applications for short reach optical communication. *Photonics* **2022**, *9*, 30. [CrossRef]
4. Wu, Q.; Xu, Z.; Zhu, Y.; Zhang, Y.; Ji, H.; Yang, Y.; Qiao, G.; Liu, L.; Wang, S.; Liang, J.; et al. Machine Learning for Self-Coherent Detection Short-Reach Optical Communications. *Photonics* **2023**, *10*, 1001. [CrossRef]
5. Ranzini, S.M.; Da, R.F.; Bülow, H.; Zibar, D. Tunable Optoelectronic Chromatic Dispersion Compensation Based on Machine Learning for Short-Reach Transmission. *Appl. Sci.* **2019**, *9*, 4332. [CrossRef]
6. Che, D.; Hu, Q.; Shieh, W. Linearization of Direct Detection Optical Channels Using Self-Coherent Subsystems. *J. Light. Technol.* **2015**, *34*, 516–524. [CrossRef]
7. Li, G.; Li, Z.; Ha, Y.; Hu, F.; Zhang, J.; Chi, N. Performance assessments of joint linear and nonlinear pre-equalization schemes in next generation IM/DD PON. *J. Light. Technol.* **2022**, *40*, 5478–5489. [CrossRef]
8. Seb, S.J.; Gavioli, G.; Killey, R.I.; Bayvel, P. Electronic compensation of chromatic dispersion using a digital coherent receiver. *Opt. Express* **2007**, *15*, 2120–2126. [CrossRef]
9. Fludger, C.R.; Duthel, T.; Van den Borne, D.; Schulien, C.; Schmidt, E.; Wuth, T.; Geyer, J.; De Man, E.; Khoe, G.; de Waardt, H. Coherent Equalization and POLMUX-RZ-DQPSK for Robust 100-GE Transmission. *J. Light. Technol.* **2008**, *26*, 64–72. [CrossRef]
10. DeLange, O.E. Optical heterodyne detection. *IEEE Spectrum* **1968**, *5*, 77–85. [CrossRef]
11. Kahn, J.M.; Barry, J.R. Wireless infrared communications. *Proc. IEEE* **1997**, *85*, 265–298. [CrossRef]
12. Huang, L.; Xu, Y.; Jiang, W.; Xue, L.; Hu, W.; Yi, L. Performance and complexity analysis of conventional and deep learning equalizers for the high-speed IMDD PON. *J. Light. Technol.* **2022**, *40*, 4528–4538. [CrossRef]
13. Kartalopoulos, S.V. *Free Space Optical Networks for Ultra-Broad Band Services*; IEEE: New York, NY, USA, 2011; Volume 256, ISBN 978-111-810-423-1.
14. Tsiatmas, A.; Willems, F.M.J.; Baggen, C.P.M.J. Square root approximation to the Poisson channel. In Proceedings of the 2013 IEEE International Symposium on Information Theory (ISIT), Istanbul, Turkey, 7–13 July 2013; pp. 1695–1699. [CrossRef]
15. Moser, S.M. Capacity results of an optical intensity channel with input-dependent Gaussian noise. *IEEE Trans. Inf. Theory* **2012**, *58*, 207–223. [CrossRef]
16. Safari, M. Efficient optical wireless communication in the presence of signal-dependent noise. In Proceedings of the ICCW, London, UK, 8–12 June 2015; pp. 1387–1391. [CrossRef]
17. Fadlullah, Z.M.; Fouda, M.M.; Kato, N.; Takeuchi, A.; Iwasaki, N.; Nozaki, Y. Toward intelligent machine-to-machine communications in smart grid. *IEEE Commun. Mag.* **2011**, *49*, 60–65. [CrossRef]
18. Yi, L.; Li, P.; Liao, T.; Hu, W. 100 Gb/s/λ IM-DD PON Using 20G-Class Optical Devices by Machine Learning Based Equalization. In Proceedings of the 44th European Conference on Optical Communication, Roma, Italy, 23–27 September 2018; pp. 1–3. [CrossRef]
19. Kaur, J.; Khan, M.A.; Iftikhar, M.; Imran, M.; Haq, Q.E.U. Machine learning techniques for 5G and beyond. *IEEE Access* **2021**, *9*, 23472–23488. [CrossRef]
20. Simeone, O. A very brief introduction to machine learning with applications to communication systems. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 648–664. [CrossRef]
21. Rodrigues, T.K.; Suto, K.; Nishiyama, H.; Liu, J.; Kato, N. Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 38–67. [CrossRef]
22. Zhang, Q.; Li, B.; Wu, R. A dynamic bandwidth allocation scheme for GPON based on traffic prediction. In Proceedings of the FSKD 2012: 9th International Conference on Fuzzy Systems and Knowledge Discovery, Chongqing, China, 29–31 May 2012; pp. 2043–2046. [CrossRef]
23. Sarigiannidis, P.; Pliatsios, D.; Zygiridis, T.; Kantartzis, N. DAMA: A data mining forecasting DBA scheme for XG-PONs. In Proceedings of the 2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 12–14 May 2016; pp. 1–4. [CrossRef]
24. Ruan, L.; Wong, E. Machine intelligence in allocating bandwidth to achieve low-latency performance. In Proceedings of the 2018 International Conference on Optical Network Design and Modeling (ONDM), Dublin, Ireland, 14–17 May 2018; pp. 226–229. [CrossRef]
25. Yi, L.; Liao, T.; Huang, L.; Xue, L.; Li, P.; Hu, W. Machine Learning for 100 Gb/s/λ Passive Optical Network. *J. Light. Technol.* **2019**, *37*, 1621–1630. [CrossRef]
26. Mikaeil, A.M.; Hu, W.; Hussain, S.B.; Sultan, A. Traffic-Estimation-Based Low-Latency XGS-PON Mobile Front-Haul for Small-Cell C-RAN Based on an Adaptive Learning Neural Network. *Appl. Sci.* **2018**, *8*, 1097. [CrossRef]
27. Chen, T.; Guestrin, C. XGBoost, A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD (2016), International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1–3. [CrossRef]
28. Ye, C.; Zhang, D.; Hu, X.; Huang, X.; Feng, H.; Zhang, K. Recurrent Neural Network (RNN) Based End-to-End Nonlinear Management for Symmetrical 50Gbps NRZ PON with 29dB+ Loss Budget. In Proceedings of the 44th European Conference on Optical Communication, Roma, Italy, 23–27 September 2018; pp. 1–3. [CrossRef]
29. Kanonakis, K.; Giacoumidis, E.; Tomkos, I. Physical-Layer-Aware MAC Schemes for Dynamic Subcarrier Assignment in OFDMA-PON Networks. *J. Light. Technol.* **2012**, *30*, 1915–1923. [CrossRef]
30. Lim, W.; Kourtessis, P.; Senior, J.M.; Na, Y.; Allawi, Y.; Jeon, S.B.; Chung, H. Dynamic Bandwidth Allocation for OFDMA-PONs Using Hidden Markov Model. *IEEE Access* **2017**, *5*, 21016–21019. [CrossRef]

31. Bi, M.; Xiao, S.; Wang, L. Joint subcarrier channel and time slots allocation algorithm in OFDMA passive optical networks. *Opt. Commun.* **2013**, *287*, 90–95. [CrossRef]
32. Zhu, M.; Gu, J.; Chen, B.; Gu, P. Dynamic Subcarrier Assignment in OFDMA-PONs Based on Deep Reinforcement Learning. *IEEE Photonics J.* **2022**, *14*, 1–11. [CrossRef]
33. Senoo, Y.; Asaka, K.; Kanai, T.; Sugawa, J.; Saito, H.; Tamai, H.; Minato, N.; Suzuki, K.I.; Otaka, A. Fairness-Aware Dynamic Sub-Carrier Allocation in Distance-Adaptive Modulation OFDMA-PON for Elastic Lambda Aggregation Networks. *J. Opt. Commun. Netw.* **2017**, *9*, 616–624. [CrossRef]
34. Nakayama, Y.; Onodera, Y.; Nguyen, A.H.N.; Hara-Azumi, Y. Real-Time Resource Allocation in Passive Optical Network for Energy-Efficient Inference at GPU-Based Network Edge. *IEEE Internet Things J.* **2022**, *9*, 17348–17358. [CrossRef]
35. Cabrera, A.; Almeida, F.; Castellanos-Nieves, D.; Oleksiak, A.; Blanco, V. Energy efficient power cap configurations through Pareto front analysis and machine learning categorization. *Clust. Comput.* **2023**, *27*, 3433–3449. [CrossRef]
36. Chen, Z.; Wang, W.; Zou, D.; Ni, W.; Luo, D.; Li, F. Real-Valued Neural Network Nonlinear Equalization for Long-Reach PONs Based on SSB Modulation. *IEEE Photonics Technol. Lett.* **2022**, *35*, 167–170. [CrossRef]
37. Xue, L.; Yi, L.; Lin, R.; Huang, L.; Chen, J. SOA pattern effect mitigation by neural network based pre-equalizer for 50G PON Opt. *Opt. Express* **2021**, *29*, 24714–24722. [CrossRef]
38. Abdelli, K.; Tropschug, C.; Griesser, H.; Pachnicke, S. Fault Monitoring in Passive Optical Networks using Machine Learning Techniques. In Proceedings of the 23rd International Conference on Transparent Optical Networks, Bucharest, Romania, 2–6 July 2023; pp. 1–5. [CrossRef]
39. Vela, A.P.; Shariati, B.; Ruiz, M.; Cugini, F.; Castro, A.; Lu, H.; Proietti, R.; Comellas, J.; Castoldi, P.; Yoo, S.J.B.; et al. Soft Failure Localization During Commissioning Testing and Lightpath Operation. *J. Opt. Commun. Netw.* **2018**, *10*, A27–A36. [CrossRef]
40. Wang, Z.; Zhang, M.; Wang, D.; Song, C.; Liu, M.; Li, J.; Lou, L.; Liu, Z. Failure prediction using machine learning and time series in optical network. *Opt. Express* **2017**, *25*, 18553–18565. [CrossRef]
41. Tufts, D.W. Nyquist's Problem—The Joint Optimization of Transmitter and Receiver in Pulse Amplitude Modulation. *Proc. IEEE* **1965**, *53*, 248–259. [CrossRef]
42. Munagala, R.L.; Vijay, U.K. A novel 3-tap adaptive feed forward equalizer for high speed wireline receivers. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4. [CrossRef]
43. Williamson, D.; Kennedy, R.A.; Pulford, G.W. Block decision feedback equalization. *IEEE Trans. Commun.* **1992**, *40*, 255–264. [CrossRef]
44. Forney, G.D. The Viterbi Algorithm: A Personal History. *arXiv* **2005**, arXiv:cs/0504020.
45. Du, L.B.; Lowery, A.J. Digital Signal Processing for Coherent Optical Communication Systems. *IEEE J. Light. Technol.* **2013**, *31*, 1547–1556.
46. Malik, G.; Sappal, A.S. Adaptive Equalization Algorithms: An Overview. *Int. J. Adv. Comput. Sci. Appl.* **2011**, 2. [CrossRef]
47. Karanov, B.; Chagnon, M.; Thouin, F.; Eriksson, T.A.; Bülow, H.; Lavery, D.; Bayvel, P.; Schmalen, L. End-to-End Deep Learning of Optical Fiber Communications. *J. Light. Technol.* **2018**, *36*, 4843–4855. [CrossRef]
48. Graves, A. Long Short-Term Memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45. [CrossRef]
49. Pascanu, R.; Gulcehre, C.; Cho, K.; Bengio, Y. How to Construct Deep Recurrent Neural Networks. *arXiv* **2013**, arXiv:1312.6026.
50. Jing, L.; Gulcehre, C.; Peurifoy, J.; Shen, Y.; Tegmark, M.; Soljacic, M.; Bengio, Y. Gated Orthogonal Recurrent Units: On Learning to Forget. *Neural Comput.* **2019**, *31*, 765–783. [CrossRef] [PubMed]
51. Qin, X.; Yang, C.; Guo, H.; Gao, Y.; Zhou, Q.; Wang, X.; Wang, Z. Recurrent Neural Network Based Joint Equalization and Decoding Method for Trellis Coded Modulated Optical Communication System. *J. Light. Technol.* **2023**, *41*, 1734–1741. [CrossRef]
52. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 2013 International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318. [CrossRef]
53. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
54. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
55. Ling, P.; Li, M.; Guan, W. Channel-Attention-Enhanced LSTM Neural Network Decoder and Equalizer for RSE-Based Optical Camera Communications. *Electronics* **2022**, *11*, 1272. [CrossRef]
56. Dey, R.; Salem, F.M. Gate-variants of Gated Recurrent Unit (GRU) neural networks. In Proceedings of the IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS 2017), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600. [CrossRef]
57. Deligiannidis, S.; Mesaritakis, C.; Bogris, A. Performance and Complexity Analysis of Bi-Directional Recurrent Neural Network Models versus Volterra Nonlinear Equalizers in Digital Coherent Systems. *J. Light. Technol.* **2021**, *39*, 5791–5798. [CrossRef]
58. Liu, X.; Wang, Y.; Wang, X.; Xu, H.; Li, C.; Xin, X. Bi-directional gated recurrent unit neural network based nonlinear equalizer for coherent optical communication system. *Opt. Express* **2021**, *29*, 5923–5933. [CrossRef] [PubMed]
59. Wu, N.; Wang, X.; Lin, B.; Zhang, K. A CNN-Based End-to-End Learning Framework Toward Intelligent Communication Systems. *IEEE Access* **2019**, *7*, 110197–110204. [CrossRef]
60. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1–9. [CrossRef]

61. Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; Xu, Q. Scinet: Time series modeling and forecasting with sample convolution and interaction. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022; pp. 5816–5828. [CrossRef]
62. Shao, C.; Giacoumidis, E.; Li, S.; Li, J.; Farber, M.; Kafer, T.; Richter, A. A Novel Machine Learning-based Equalizer for a Downstream 100G PAM-4 PON. In Proceedings of the 2024 Optical Fiber Communication Conference and Exhibition (OFC'24), San Diego, CA, USA, 24–28 March 2024. Available online: https://opg.optica.org/abstract.cfm?uri=ofc-2024-W1H.1&origin=search (accessed on 1 March 2024).
63. Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; Li, J. Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-oriented MLP Structures. In Proceedings of the Conference on Robot Learning (CoRR 2022), Auckland, New Zealand, 14 December 2022. [CrossRef]
64. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? In Proceedings of the Conference on Artificial Intelligence (AAAI 2022), Arlington, VA, USA, 17–19 November 2022. [CrossRef]
65. Kiencke, U.; Eger, R. *Messtechnik: Systemtheorie für Elektrotechniker*; Springer: Berlin/Heidelberg, Germany, 2008; XII 341, ISBN 978-3-540-78429-6.
66. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010. [CrossRef]
67. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. In Proceedings of the International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia, 26–30 April 2020. [CrossRef]
68. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Neural Information Processing Systems, (NIPS 2021), Virtual, 6–14 December 2021; Volume 34, pp. 22419–22430.
69. Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.; Yan, X. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, WA, USA, 8–14 December 2019.
70. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI conference on artificial intelligence, Vancouver, BC, Canada, 2–9 February 2021; pp. 11106–11115. [CrossRef]
71. Shabani, A.; Abdi, A.; Meng, L.; Sylvain, T. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In Proceedings of the International Conference on Learning Representations (2022), Virtual, 25–29 April 2022. [CrossRef]
72. Chen, W.; Wang, W.; Peng, B.; Wen, Q.; Zhou, T.; Sun, L. Learning to Rotate: Quaternion Transformer for Complicated Periodical Time Series Forecasting. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22), Association for Computing Machinery, Washington, DC, USA, 14–18 August 2022; pp. 146–156. [CrossRef]
73. Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A.X.; Dustdar, S. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In Proceedings of the International Conference on Learning Representations (ICLR 2022), Virtual, 25–29 April 2022. [CrossRef]
74. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In Proceedings of the 39th International Conference on Machine Learning (PMLR 2022), Baltimore, MD, USA, 17–23 July 2022; p. 162.
75. Zhang, Y.; Yan, J. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In Proceedings of the International Conference on Learning Representations (2023), Kigali, Rwanda, 1–5 May 2023.
76. Cirstea, R.; Guo, C.; Yang, B.; Kieu, T.; Dong, X.; Pan, S. Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting-Full Version. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2022), Vienna, Austria, 23–29 July 2022. [CrossRef]
77. Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; Eickhoff, C. A transformer-based framework for multivariate time series representation learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD 2021), Singapore, 8–14 December 2021; pp. 2114–2124. [CrossRef]
78. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In Proceedings of the International Conference on Learning Representations (ICLR 2022), Virtual, 25–29 April 2022. [CrossRef]
79. Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; Niu, Z. Frequency-domain MLPs are More Effective Learners in Time Series Forecasting. In Proceedings of the Thirty-Seventh Conference on Neural Information Processing Systems (NIPS 2023), New Orleans, LA, USA, 10–16 December 2023; arXiv:2311.06184. [CrossRef]
80. Franceschi, J.; Dieuleveut, A.; Jaggi, M. Unsupervised Scalable Representation Learning for Multivariate Time Series. In Proceedings of the Neural Information Processing Systems (NIPS 2019), Vancouver, WA, USA, 8–14 December 2019. [CrossRef]
81. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019. [CrossRef]
82. Shao, C.; Giacoumidis, E.; Matalla, P.; Li, J.; Li, S.; Randel, S.; Richter, A.; Faerber, M.; Kaefer, T. Advanced Equalization in 112 Gb/s Upstream PON Using a Novel Fourier Convolution-Based Network, Submitted to the European Conference on Optical Communication (ECOC 2024). Available online: https://arxiv.org/pdf/2405.02609 (accessed on 1 March 2024).

83. Lew, J.; Shah, D.; Pati, S.; Cattell, S.; Zhang, M.; Sandhupatla, A.; Ng, C.; Goli, N.; Sinclair, M.D.; Rogers, T.G.; et al. Analyzing Machine Learning Workloads Using a Detailed GPU Simulator. In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2019), Wisconsin, WI, USA, 24–26 March 2019; pp. 151–152. [CrossRef]
84. Marculescu, D.; Stamoulis, D.; Cai, E. Hardware-Aware Machine Learning: Modeling and Optimization. In Proceedings of the 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), New York, NY, USA, 5–8 November 2018; pp. 1–8. [CrossRef]
85. Freire, P.J.; Osadchuk, Y.; Spinnler, B.; Napoli, A.; Schairer, W.; Costa, N.; Prilepsky, J.E.; Turitsyn, S.K. Performance Versus Complexity Study of Neural Network Equalizers in Coherent Optical Systems. *J. Light. Technol.* **2021**, *39*, 6085–6096. [CrossRef]
86. Asghar, M.Z.; Abbas, M.; Zeeshan, K.; Kotilainen, P.; Hämäläinen, T. Assessment of Deep Learning Methodology for Self-Organizing 5G Networks. *Appl. Sci.* **2019**, *9*, 2975. [CrossRef]
87. Xu, Z.; Sun, C.; Ji, T.; Manton, J.H.; Shieh, W. 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* **2019**, *27*, 36953–36964. [CrossRef] [PubMed]
88. Bucila, C.; Caruana, R.; Niculescu-Mizil, A. Model compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 20–23 August 2006; pp. 535–541. [CrossRef]
89. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* **2020**, *129*, 1789–1819. [CrossRef]
90. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**. [CrossRef].
91. Chang, A.X.; Culurciello, E. Hardware accelerators for recurrent neural networks on FPGA. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS 2017), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4. [CrossRef]
92. Srivallapanondh, S.; Freire, P.J.; Spinnler, B.; Costa, N.; Napoli, A.; Turitsyn, S.K.; Prilepsky, J.E. Knowledge Distillation Applied to Optical Channel Equalization: Solving the Parallelization Problem of Recurrent Connection. In Proceedings of the 2023 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 5–9 March 2023; p. Th1F.7. [CrossRef]
93. Robert, R.; Yuliana, Z. *Parallel and High Performance Computing*; Manning Publications: Greenwich, CT, USA, 2021; Volume 704, ISBN 978-161-729-646-8.
94. Hayashi, S.; Tanimoto, A.; Kashima, H. Long-Term Prediction of Small Time-Series Data Using Generalized Distillation. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2019), Budapest, Hungary, 14–19 July 2019; pp. 1–8. [CrossRef]
95. Gray, R.M. Vector quantization. *IEEE Assp Mag.* **2019**, *1*, 4–29. [CrossRef]
96. Pourghasemi, H.R.; Gayen, A.; Lasaponara, R.; Tiefenbacher, J.P. Application of learning vector quantization and different machine learning techniques to assessing forest fire influence factors and spatial modelling. *Environ. Res.* **2020**, *184*, 109321. [CrossRef] [PubMed]
97. Wang, X.; Takaki, S.; Yamagishi, J.; King, S.; Tokuda, K. A Vector Quantized Variational Autoencoder (VQ-VAE) Autoregressive Neural F_0 Model for Statistical Parametric Speech Synthesis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *28*, 157–170. [CrossRef]
98. Rasul, K.; Park, Y.J.; Ramström, M.N.; Kim, K.M. VQ-AR: Vector Quantized Autoregressive Probabilistic Time Series Forecasting. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024), Vienna, Austria, 7–11 May 2024. [CrossRef]
99. Ozair, S.; Li, Y.; Razavi, A.; Antonoglou, I.; Van Den Oord, A.; Vinyals, O. Vector quantized models for planning. In Proceedings of the International Conference on Machine Learning (ICML 2021), Virtual, 18–24 July 2021; pp. 8302–8313. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.