Machine Learning for Multi-Layer Open and Disaggregated Optical Networks

Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electrical, Electronic and Communications Engineering
(34^{th}cycle)

Machine Learning for Multi-Layer Open and Disaggregated Optical Networks

By

Ihtesham Khan

******

Supervisor(s):
Prof. Vittorio Curri

Doctoral Examination Committee:
Prof. Francesco Musumeci , Referee, Politecnico di Milano
Prof. Francesco Da Ros, Referee, Technical University of Denmark - DTU

Politecnico di Torino
2022

I would like to dedicate this thesis to my loving parents and wife

Acknowledgements

First of all, I would like to thank my advisor Vittorio Curri for giving me a chance to join the PLANET group and start this Ph.D. I want to be thankful for his mentorship and continuous guidance in the world of academia.

Next, I would like to thank all the affiliates of the OptCom group, mainly thanks to Andrea Carena for his constant support. Along with Andrea Carena, I also want to acknowledge Paolo Bardella for his kindness and support. I feel highly indebted for all the feedback and suggestion they gave me on my day-to-day workout.

I also wish to thank all the industrial persons who assisted me in understanding the complex concepts and provided me with the confidence to develop novel solutions with real-world applicability.

I also would like to thank all the PLANET team members from Dept. of Electronics and Telecommunications for their friendship, and the good times we spent together. Thanks Emanuele Virgilito, Andrea D'Amico, Rasoul Sadeghi, M. Umar Masood, Giacomo Borraccini, Bruno Correia, Elliot London and also former team members Muhammad Bilal and Alessio Ferrari.

Finally and most importantly, none of this would have been possible without my family's support and unconditional love to support me in achieving new heights. Special thanks to my mother for her continued and unparalleled emotional support, my father for his patience and dedication, and my wife for her endless support and always being there during these three years.

Abstract

In this thesis, data-driven techniques for multi-layer optical networks are described. The fast growth in global IP traffic and the advancement in the communication technologies, such as coherent optical transmission, elastic optical networks, and the opening of software-defined optical networks (SDON), introduce many tunable parameters, making the design and operation of optical networks more complex.

In this framework, exact system modeling using closed-form formulations is very challenging, and typically, a "margin" is deployed while adopting the analytical models. The deployment of specific margins leads to the under-utilization of resources and eventually enlarges the network operational cost. To cater to this limitation of analytical models, the telecom industry is strongly pushing the optical community to move towards intelligent optical networks that can perform autonomous and flexible network management and optimize network resources utilization. Thus, smart optical networks are fundamental to the modern optical network to assess the potentialities better and exploit various technologies' capabilities.

Moving towards intelligent optical networks and data centers, large-scale photonic switches and wavelength selective switches play a prominent role due to their wide-band capabilities, minimal latency, and low power consumption. These distinctive properties increase the possibilities of using photonic integrated circuits based network elements, especially photonic switches, and hence it generates a demand for a generic softwarized model for control states and quality of transmission (QoT) degradation to enable full control by a centralized controller.

In developing an intelligent optical network, the basic framework of SDON is the leading enabling technology in the direction of intelligent optical networks, comprising three principal planes, the data plane, control plane, and application plane. The assisting cognitive module is logically centralized and is part of the control plane. In this novel playground of SDON, machine learning-assisted QoT

estimation in optical networks is presented. The machine learning model's training on the generalized signal-to-noise ratio (GSNR) response to specific spectral load configurations of the already deployed *in-service* network, and consider its realization to predict the QoT of an *agnostic/un-used* network. The synthetic dataset is retrieved from lightpath QoT responses against random spectral loads of *in-service* network. This data is generated during the operative phase of the *in-service* network by measuring the optical line system (OLS) response in terms of GSNR for various spectral load configurations

Furthermore, a framework of machine learning-assisted photonic devices management is given, in the context of SDON. This generic topology-agnostic *blind* framework exploits a machine learning inverse design approach to obtain the softwarized control of any $N \times N$ photonic switching system by giving the permutations of the signals (λ_1, λ_2, λ_3....λ_n) at the output ports. Moreover, a direct design method is also presented to predict the QoT degradation due to the penalty encounter by crossing the optical switching elements. The previously obtained controls are exploited to predict the QoT degradation in terms of optical signal-to-noise ratio (OSNR) Penalty.

Finally, several supervised machine learning models are also proposed using the two most common open-source machine learning libraries, TensorFlow© and *scikit-learn*©. The performance assessment of these developed models is also performed in terms of assisting the QoT estimation and also for photonic devices management.

Contents

List of Figures **x**

List of Tables **xv**

List of Scientific Contributions **xvi**

1 Introduction **1**

1.1 Motivation 1

1.2 Introduction to Optical networks 3

1.3 Evolution towards software-defined open and disaggregated optical networks 4

1.4 Software-defined intelligent optical networks 8

1.5 Outline of the thesis 10

2 Use of Artificial Intelligence for Open and Disaggregated Optical Networks **12**

2.1 Artificial Intelligence 12

2.2 Review of machine learning techniques adopted in optical networks 14

2.2.1 Supervised learning 14

2.2.2 Unsupervised learning 23

2.2.3 Semi-supervised learning 25

2.2.4 Reinforcement learning 26

2.3 Multi-layer applications of machine learning in optical networks . . 26
2.3.1 Machine learning at physical layer 26
2.3.2 Machine learning at network layer 36
2.3.3 Machine learning at Photonics or below layer-0 42

3 Machine Learning for Optical Networks 45

3.1 Physical layer abstraction of transparent open optical network . . . 46
3.2 GSNR as QoT-E Metric 48
3.3 Approaches for QoT Estimation 50
3.4 Simulation model and synthetic data generation 52
3.4.1 Analysis of GSNR Dataset 55
3.5 Machine learning techniques for QoT-E 57
3.5.1 Decision Tree Regressor 61
3.5.2 Random Forest Regressor 62
3.5.3 Multi-layer Perceptron Regressor 62
3.5.4 Boosted Tree Regressor 63
3.5.5 Deep Neural Network Regressor 63
3.5.6 Wide Deep Neural Network 63
3.6 Performance analysis of machine learning models 64
3.6.1 Scikit-learn© Based Learning Methods for QoT-E 64
3.6.2 TensorFlow© Based Learning Methods for QoT-E 66
3.7 Synergies use of machine learning agent and QoT-E engine of network 69
3.7.1 Operational GSNR margin and statistical analysis of GSNR 70
3.7.2 Transfer learning agent 74
3.7.3 QoT-E correction using transfer learning agent 77

4 Machine Learning for Photonic Devices 79

4.1 Elementary switching module and topologies 81
4.1.1 2×2 crossbar switch 81
4.1.2 Multistage Rearrangeable Non-Blocking switching architectures 82
4.1.3 Problem complexity 83
4.1.4 Topologies under analysis 83
4.2 Simulation model and dataset acquisition 86
4.2.1 Routing model 86
4.2.2 Transmission model 86
4.3 Machine learning for routing management of photonic devices . . . 87
4.3.1 Decision tree regression 89
4.3.2 Random Forest Regression 89
4.3.3 Boosted tree regression 90
4.3.4 Linear regression 90
4.3.5 Deep neural network 91
4.4 Performance analysis of machine learning models 92
4.5 Machine learning for QoT evaluation of photonic devices 96
4.5.1 Data generation and statistical analysis of OSNR penalty . . 97
4.6 Inverse and direct machine learning modeling for routing and QoT . 99
4.6.1 Machine learning routing and QoT agent 100
4.7 Machine learning assisted control of wideband switching fabrics . . 103
4.7.1 Ultra-wideband switching system 103
4.7.2 Simulation environment and dataset generation 106
4.7.3 Detail analysis and case study 113

5 Conclusions and Future Work **116**

5.1 Summary 116

5.2 Next Steps 118

Appendix **119**

Acronyms **121**

References **128**

List of Figures

1.1 Aggregated and disaggregated networks 5

1.2 Single-vendor legacy line system. 6

1.3 Multi-vendor software-defined open line system. 7

1.4 The general software-defined intelligent optical network architecture. 8

2.1 Representing deep learning as a subset of machine learning 13

2.2 Machine learning classification 14

2.3 Decision tree 15

2.4 Random forest 17

2.5 Boosted tree mechanism 17

2.6 Neural network activation functions common examples 18

2.7 Deep neural network architecture 19

2.8 Wide-deep neural network architecture 20

2.9 Convolutional neural networks 21

2.10 Feed-forward neural network vs. recurrent neural network 22

2.11 (a) Support vector machine (b) K-means clustering (c) Gaussian Mixture Model 24

2.12 Machine learning applications in optical networks 27

2.13 Machine-learning-assisted SDN-enabled optical networks 43

3.1 Schematic description of an optical network as a topology of ROADM nodes connected by OLSs. The inset shows a general setup for an OLS that, in this case, is supposed to be open. 47

3.2 Abstraction of Optical Network 49

3.3 Model Orchestration 51

3.4 Network Topologies 53

3.5 GSNR statistic of all channels of *Louisville* → *Memphis* path 55

3.6 Overall, GSNR measurements for a single path Louisville → Memphis of an *agnostic/unused* network in the frequency domain. The green dots are the mean values over the entire sample for each channel; the error bars are equal to the standard deviations. The purple line shows the nominal values for this path in the frequency domain. In blue and orange, the maximum and the minimum for each channel are outlined, respectively. The dashed red line indicates the overall GSNR minimum of 10.81 dB. 56

3.7 Machine Learning Module 57

3.8 Hyperparameters optimization of tree-based models using Skopt . . 59

3.9 Hyperparameters optimization of neural network-based models using Skopt 60

3.10 W-DNN loss function over the training Steps 64

3.11 *SKL* Based Learning Methods 65

3.12 *TF* Based Learning Methods 65

3.13 (a) *SKL* Based Learning Methods (b) *TF* Based Learning Methods . 66

3.14 Distribution of $\Delta GSNR$ for the simulated links of USA network using *MLPR* 67

3.15 (a) Single run training time of *SKL* based learning methods (b) Single run training time of *TF* based learning methods 67

3.16 Distribution of $\Delta GSNR$ for the simulated links of USA network using *wdnn* 68

3.17 Normal Q-Q plot of overall GSNR measurements for the path Louisville → Memphis 71

3.18 GSNR measurements of randomly selected channels of Louisville → Memphis path 72

3.19 Analytical GSNR margin estimation for channel-58 path *Louisville* → *Memphis*: green, red, blue, and black vertical lines represent the values of the distribution mean and the nominal, operative, and worst-case GSNR, respectively. The dashed line sketches a Gaussian shape with the same mean and standard deviation of the considered distribution. 73

3.20 Model Schematic: The Machine Learning module assists the QoT engine (GNPy) 75

3.21 Description of the Machine Learning Module. 76

3.22 ΔGSNR distributions obtained with and without ML module for all the investigated path of the *unused* USA network 78

4.1 Abstraction of the optical switch in a SDN-controlled optical network. 80

4.2 Illustration of Bar and Cross states of a 2×2 elementary switching element (CrossBar switch). 82

4.3 Multistage switching networks topologies under analysis 84

4.4 Graphical representation of all possible $N!$ output states for a $N \times N$ fabric. 85

4.5 Generic $N \times N$ transmission model 87

4.6 Schematic of machine learning framework. 89

4.7 Parallel architecture of DNN with hidden layers. 91

4.8 (a) Mean Square Error of ML models (b) Training Time of ML models. 92

4.9 (a) DNN loss function vs. the training steps for Beneš 8×8 architecture. (b) Percentage of correct predictions vs. normalized training dataset size. The normalization is performed with respect to the total generated dataset dimension for the considered N×N fabric, see data in Table 4.1. (c) Percentage of correct predictions vs. hidden layer size for the considered switching configurations (d) Single switch training time vs. hidden layer size. 93

4.10 Heatmap showing normalized error in prediction of control signals using DNN. 95

4.11 Circuit representation for the two device under test 97

4.12 Statistical Analysis of OSNR Penalty 98

4.13 Control Unit block model 98

4.14 Parallel architecture of a deep neural network 99

4.15 Probability density functions of ΔOSNR for each port of the 6x6 Beneš switch. 101

4.16 Probability density functions of ΔOSNR for each port of the 8x8 Beneš switch. 102

4.17 (a) Optical switching element (OSE) black-box model. (b) MZI switching circuit. (c) and (d) Dependence of transmission properties in BAR and CROSS state on wavelength and temperature. 104

4.18 (a)-(b) MZI switch structures: single coupler vs. double coupler. (c)-(d) Corresponding transmission bandwidth properties. 106

4.19 Transmission dependence on wavelength and thermal tunability for the second order coupling MZI device tested in the paper. 107

4.20 Simulation model schematic and characteristics 112

4.21 Statistical analysis of OSNR Penalties for each output port. 113

4.22 Probability density functions of ΔOSNR for each port of the 8x8 Beneš switch. Average values μ and variances σ indicated for the individual cases are expressed in decibel. 114

4.23 OSNR Penalty distribution for 32 nominally equivalent control states generating the output pattern [7, 6, 3, 8, 5, 4, 1, 2]. A label from 1 to 32 has been assigned to each control state according to the order it is generated by the proposed algorithm. 115

List of Tables

2.1 Summary of Machine learning Applications at Physical layer . . . 34

2.2 Summary of Machine learning Applications at Network layer . . . 41

2.3 Summary of Machine learning Applications at Photonics or below layer-0 . . . 44

3.1 Simulation Parameters . . . 51

3.2 Train set network paths . . . 53

3.3 Test set network paths . . . 54

3.4 Networks Topology Details . . . 55

3.5 Machine learning Models Detail . . . 61

4.1 Dataset Statistics . . . 87

4.2 Machine learning Models Detail . . . 90

4.3 Summary of ML prediction results related to control states . . . 96

4.4 Dataset for QoT Evaluation . . . 97

1 Networks Topology Details . . . 120

List of Scientific Contributions

sec:intro A list of the scientific contributions, with special focus on publications, carried out during the PhD are described here.

Original publications included in this PhD thesis

Peer Reviewed International Journals

[1] D'Amico, A., Straullu, S., Nespola, A., Khan, I., London, E., Virgillito, E., ... & Curri, V. (2020). Using machine learning in an open optical line system controller. Journal of Optical Communications and Networking, 12(6), C1-C11.

[2] Khan, I., Bilal, M., & Curri, V. (2020). Assessment of cross-train machine learning techniques for QoT-estimation in agnostic optical networks. OSA Continuum, 3(10), 2690-2706.

[3] Khan, I., Bilal, M., Masood, M. U., D'Amico, A., & Curri, V. (2021). Lightpath QoT computation in optical networks assisted by transfer learning. Journal of Optical Communications and Networking, 13(4), B72-B82.

[4] Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., & Curri, V. (2021). Automatic Management of N× N Photonic Switch Powered by Machine Learning in Software-defined Optical Transport. IEEE Open Journal of the Communications Society.

[5] Usmani, F, Khan, I, Masood, MU, Ahmad, A, Shahzad, M, & Curri, V. (2021). Convolutional neural network for quality of transmission prediction of un-

established lightpaths. Microw Opt Technol Lett. 2021; 63: 2461– 2469. https://doi.org/10.1002/mop.32996

[6] Usmani, F, Khan, I, Siddiqui M , Khan M , Bilal M , Masood, MU, Ahmad, A, Shahzad, M, & Curri, V. (2021). "Cross-feature trained machine learning models for QoT-estimation in optical networks," Opt. Eng. 60(12) 125106 (3 December 2021) https://doi.org/10.1117/1.OE.60.12.125106

[7] Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., & Curri, V. (2021). Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., & Curri, V. (2022). Performance Evaluation of Data-Driven Techniques for Softwarized and Agnostic Management of N×N Photonic Switch. Opt. Continuum, 1(1) https://doi.org/10.1364/OSAC.428567.

[8] Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., & Curri, V. (2021). Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., & Curri, V. (2022). Optimized management of ultra-wideband photonics switching systems assisted by machine learning. Opt. Express, 30(3) https://doi.org/10.1364/OE.442194.

Contributions to International Conferences

[1] D'Amico, A., Straullu, S., Nespola, A., Khan, I., Abdelfattah, S., Virgillito, E., ... & Curri, V. (2019, September). Machine-learning aided OSNR prediction in optical line systems. In 45th European Conference on Optical Communication (ECOC 2019) (pp. 1-4). IET.

[2] Khan, I., Bilal, M., & Curri, V. (2020, July). Advanced formulation of QoT-estimation for un-established lightpaths using cross-train machine learning methods. In 2020 22nd International Conference on Transparent Optical Networks (ICTON) (pp. 1-4). IEEE.

[3] Khan, I., Bilal, M., Siddiqui, M., Khan, M., Ahmad, A., Shahzad, M., & Curri, V. (2020, July). QoT estimation for light-path provisioning in un-seen optical networks using machine learning. In 2020 22nd International Conference on Transparent Optical Networks (ICTON) (pp. 1-4). IEEE.

[4] Khan, I., Chalony, M., Ghillino, E., Masood, M. U., Patel, J., Richards, D., ... & Curri, V. (2020, July). Machine learning assisted abstraction of photonic integrated circuits in fully disaggregated transparent optical networks. In 2020 22nd International Conference on Transparent Optical Networks (ICTON) (pp. 1-4). IEEE.

[5] Khan, I., Chalony, M., Ghillino, E., Masood, M. U., Patel, J., Richards, D., ... & Curri, V. (2020). Effectiveness of machine learning in assessing qot impairments of photonics integrated circuits to reduce system margin. In 2020 IEEE Photonics Conference (IPC) (pp. 1-2). IEEE.

[6] Khan I., Bilal M., & Curri, V. (2020) Cross-Train: Machine Learning Assisted QoT-Estimation in Un-used Optical Networks. In: Ma M. (eds) Proceedings of the 4th International Conference on Telecommunications and Communication Engineering. ICTCE 2020. Lecture Notes in Electrical Engineering, vol 797. Springer, Singapore. https://doi.org/10.1007/978-981-16-5692-7_9

[7] Khan, I., Masood, M. U., Tunesi, L., Ghillino, E., Bardella, P., Carena, A., & Curri, V. Softwarized and Autonomous Management of Photonic Switching Systems Using Machine Learning. In:2021 International Conference on Optical Network Design and Modeling (ONDM), pp.1–6

[8] Khan, I., Tunesi, L., Chalony, M., Ghillino, E., Masood, M. U., Patel, J., ... & Curri, V. (2021, March). Machine-learning-aided abstraction of photonic integrated circuits in software-defined optical transport. In Next-Generation Optical Communication: Components, Sub-Systems, and Systems X (Vol. 11713, p. 117130Q). International Society for Optics and Photonics.

[9] Tunesi, L., Giannuzzi, G., Khan, I., Patel, J., Ghillino, E., Curri, V., ... & Bardella, P. (2021, March). Automatic design of NxN integrated Benes optical switch. In Silicon Photonics XVI (Vol. 11691, p. 1169110). International Society for Optics and Photonics.

[10] Khan, I., Masood, M. U., Tunesi, L., Bardella, P., Ghillino, E., Carena, A., & Curri, V. (2021, May). Machine Learning Assisted Management of Photonic Switching Systems. In CLEO: QELS_Fundamental Science (pp. JTu3A-32). Optical Society of America.

[11] Usmani, F, Khan, I, Masood, MU, Ahmad, A, Shahzad, M, & Curri, V. (2021). "Evaluating Cross- feature Trained Machine Learning Models for Estimating QoT of Unestablished Lightpaths," 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021, pp. 1-6, doi: 10.1109/ICECCE52056.2021.9514154.

[12] Khan, I., Masood, M. U., Tunesi, L., Bardella, P., Ghillino, E., Carena, A., & Curri, V. (2021, July) "A Data-Driven Approach to Autonomous Management of Photonic Switching System," 2021 IEEE Photonics Society Summer Topicals Meeting Series (SUM), 2021, pp. 1-2, doi: 10.1109/SUM48717.2021.9505780.

[13] Khan, I., Masood, M. U., Tunesi, L., Bardella, P., Ghillino, E., Carena, A., & Curri, V. (2021, July) "A Neural Network-Based Automatized Management of N ×N Integrated Optical Switches," in OSA Advanced Photonics Congress 2021, OSA Technical Digest, paper NeF2B.2.

[14] Khan, I., Tunesi, L., Masood, M. U., Bardella, P., Ghillino, E., Carena, A., & Curri, V. , "Machine Learning Assisted Model of QoT Penalties for Photonics Switching Systems," in Photonics in Switching and Computing 2021, W. Bogaerts, K. Morito, S. Ben Yoo, M. Fiorentino, K. Ishii, and B. Offrein, eds., OSA Technical Digest (Optical Society of America, 2021), paper M2A.3.

[15] Khan, I., Tunesi, L., Masood, M. U., Bardella, P., Ghillino, E., Carena, A., & Curri, V. , "Machine learning Assisted Accurate Estimation of QoT Impairments of Photonics Switching System on 400ZR," in Asia Communications and Photonics Conference 2021, C. Chang-Hasnain, A. Willner, W. Shieh, P. Shum, Y. Su, G. Li, B. Eggleton, R. Essiambre, D. Dai, and D. Ma, eds., OSA Technical Digest (Optica Publishing Group, 2021), paper T2B.2.

[16] Khan, I., Tunesi, L., Masood, M. U., Bardella, P., Ghillino, E., Carena, A., & Curri, V. , "Autonomous Control Model for C+L Multi-band Photonic Switch System using Machine Learning," in Asia Communications and Photonics Conference 2021, C. Chang-Hasnain, A. Willner, W. Shieh, P. Shum, Y. Su, G. Li, B. Eggleton, R. Essiambre, D. Dai, and D. Ma, eds., OSA Technical Digest (Optica Publishing Group, 2021), paper T4A.163.

[17] Khan, I., Tunesi, L., Masood, M. U., Bardella, P., Ghillino, E., Carena, A., & Curri, V. , "Machine Learning Driven Model for Software Manage-

ment of Photonics Switching Systems," 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 1-6, doi: 10.1109/GLOBECOM46510.2021.9685878.

[18] Khan, I., Tunesi, L., Masood, M. U., Bardella, P., Ghillino, E., Carena, A., & Curri, V., "Optimal control of Beneš optical networks assisted by machine learning", Proc. SPIE 12028, Next-Generation Optical Communication: Components, Sub-Systems, and Systems XI, 120280I (3 March 2022); https://doi.org/10.1117/12.2608595

Other contributions published during this PhD thesis

[1] Khan, I., Ahmad, A., Masood, M. U., Malik, A. W., Ahmed, N., & Curri, V. (2020). Impact of data center placement on the power consumption of flexible-grid optical networks. Optical Engineering, 59(1), 016115.

[2] Masood, M. U., Khan, I., Ahmad, A., Imran, M., & Curri, V. (2020, June). Smart Provisioning of Sliceable Bandwidth Variable Transponders in Elastic Optical Networks. In 2020 6th IEEE Conference on Network Softwarization (NetSoft) (pp. 85-91). IEEE.

[3] S. Alam, M. U. Masood, I. Khan, A. Ahmad, S. Ghafoor and V. Curri, "Modeling Off-line Routing and Spectrum Allocation Problem in Elastic Optical Network," 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021, pp. 1-6, doi: 10.1109/ICECCE52056.2021.9514112.

[4] Shahzad, F., Khan, I., Masood, M. U., Ahmad, A., Imran, M., Ruffini, M., & Curri, V. (2021, June). Impairment-aware Virtual Network Embedding Using Time Domain Hybrid Modulation formats in Optical Networks. In 2021 International Conference on Optical Network Design and Modeling (ONDM) (pp. 1-6). IEEE.

[5] S. Alam, I. Khan, M. U. Masood, A. Ahmad, S. Ghafoor & V. Curri, "Routing and Spectrum Allocation Heuristic for Sliced Elastic Optical Network System," 2021 IEEE Photonics Society Summer Topicals Meeting Series (SUM), 2021, pp. 1-2, doi: 10.1109/SUM48717.2021.9505961.

Chapter 1

Introduction

Significant improvements have been made in optical networks capacity, reliability, and flexibility in the last few decades. These improvements enable the expansion of optical networks from long haul to metro and finally fiber-to-the-home (FTTH). This vast expansion and dynamic network infrastructures demand advancement towards the intelligent optical network that can operate autonomously at different levels of operations. This thesis explores supervised machine learning techniques that can assist in operating optical and photonics networks at different levels. Chapter 1 introduces optical communication and other evolving technologies that pave a path for an intelligent optical network. Then in Chapter 2, artificial intelligence, its different classes, and its multi-layer applications in the modern optical networks are reported. Later in Chapter 3, a framework for machine learning-assisted optical network management is discussed, and several applications are demonstrated. Then, in Chapter 4, a framework for machine learning-assisted photonic devices management is presented. Finally, in Chapter 5, the conclusion is given, and possible future directions are reviewed.

1.1 Motivation

Recently, the remarkable increase in the global IP traffic (nearly a 26% compound annual growth rate (CAGR) increase in global IP traffic between 2017 and 2022), compelled mainly by the introduction of 5G technology, internet of things (IoT), and

cloud services have marked up high demands and new requirements for the capacity enhancement and reliability of optical networks [1]. To serve this rapid increase in internet users, global internet and optical communication have made incredible advancements, both at services and technology levels such as coherent optical technologies for wavelength-division multiplexing (WDM) optical transport [2], an elastic optical network (EON) [3] and the beginning of software-defined networking (SDN) paradigm [4]. The advancement towards these technologies improves the capacity, data rate, and reliability of the network but introduces a high number of tunable parameters (single/multi-carrier transmission, baud rate, adaptive modulation format, forward error correction (FEC) and adaptive channels spacing, etc.), which make the design and operation of optical networks more complex.

In this framework, exact system modeling using closed-form formulations is very challenging, and typically, a "margin" is deployed while adopting the analytical models. The deployment of specific margins leads to the under-utilization of resources and eventually enlarges the network operational expenditures (OPEX). To cater to this shortcoming of analytical models, it is necessary to move towards intelligent or cognitive optical networks that can perform autonomous and flexible network management. Besides this, the manual control of the present complex network operating systems, typically by human administrators, is more prone to error and increases OPEX.

To this aim, different technologies and techniques are exploited by the telecom industry to start building intelligent optical networks that can improve end-to-end communication both at the service (Quality of Service (QoS)) and the transmission (QoT) levels, using self-learning and real-time management system implementations at the network layer. The cognitive ability of intelligent optical networks empowers them to self-configure, self-optimize, and manage the network operation autonomously. In this thesis, first, we describe the evolution of classical optical networks towards intelligent optical networks. Later, we discuss the possible enabling technologies and techniques that mainly assist in developing intelligent optical networks.

1.2 Introduction to Optical networks

Optical networks are the central backbone communication infrastructure, mainly consist of optical fiber and other transmission equipment such as amplifiers and optical add/drop multiplexers. In optical networks, data is transferred through light channels (lightpaths (LPs)) over optical fiber. Optical networks have some characteristic properties such as low loss, high data rate, and high bandwidth. These unique characteristics make them a good candidate for reliable, high capacity, and long haul communication.

The main breakthrough in the optical transmission system is its advancement from the point-to-point transmission to completely meshed networks, where LP can be added/dropped or rerouted to any given node. The modern optical network fulfills the high-bandwidth requirement of network layers and provides other network operations such as routing, different protection schemes, and quick fault restoration. The enabling of these network functionalities and the performances of optical networks are closely linked to the evolution in optical fiber, optical network elements (NE), and photonic technologies. In the past few decades, the extraordinary advancements in these fields highly enhanced the capacity, reliability, and transmission rate of optical networks.

The first-ever optical communication system was implemented as a point-to-point link between Turin (Italy) and Long Beach (US, California) to enable a telephone call in 1977 [5, 6]. In 1986, the development of Erbium-Doped Fiber Amplifiers (EDFAs) [7] and the significant reduction of fiber losses granted a notable boost in the optical reach of point-to-point communication. The innovation in transceivers steadily increased the data rates per channel (CAGR of 20%) [8], beginning from intensity modulation using direct detection (IMDD) transceivers (few Mbps) to 400 Gbps transmission channels using dual-polarization with quadrature amplitude modulation (QAM) formats. Additionally, the development in the digital signal processing (DSP) techniques simplifies the optical line system (OLS) by enabling the in-line dispersion compensating units (DCUs).

The introduction of WDM techniques further increased the capacity of optical communication systems. The WDM technology employs multiple optical channels around different optical frequencies and transmits them through the same optical fiber. This enables efficient fiber bandwidth utilization by holding numerous optical

channels, enhancing the overall volume of transferred data. In the 1990s, with the introduction of commercial WDM systems, per-fiber capacity doubled with each passing year (100% CAGR). In 2000 and afterward, this growth in fiber capacity has rapidly decreased to a 20% CAGR [8]. Thus, in commercial systems following the same trend, the total sum of WDM channels capacity remained the same for roughly two decades, having reached nearly 100 parallel channels. Nevertheless, today's introduction of technologies such as band-division multiplexing (BDM) have yielded a promising solution to expand the fiber bandwidth beyond the existing C-band; for example, C+L band systems increase the total number of WDM channels close to 200 [9], and commercial systems with these features are currently available in the market [10].

Finally, the most revolutionary advancement in optical communication systems was the development of a technology that can add/drop and route an LP at different network nodes. This technology simplifies channels deployment significantly by replacing the manually rearranging LP through optical demultiplexers and patch panels [11–14] by an automated, agile and robust method [15]. The NE that brought all these functionalities into play are optical add/drop multiplexers (OADMs). OADMs were initially developed using optical filters, which allowed an independent LP to be added/dropped or bypassed at a specific network node. Traditional OADMs have restricted adaptability to different traffic and channel configurations because of their built-in fixed optical configurations. In order to cater to this severe adaptability problem, reconfigurable optical add/drop multiplexers (ROADMs) have been suggested. Currently, ROADMs are massively adopted in most of the state-of-the-art optical network infrastructure, empowering completely agile, flexible and dynamic networking.

1.3 Evolution towards software-defined open and disaggregated optical networks

The key revolutionary step in optical communication systems is the beginning of DSP-based transceivers, which can support multilevel modulation formats exploiting coherent technology [16, 17]. Before introducing DSP-based transceivers, optical communication systems were widely operated with direct-detection (DD)

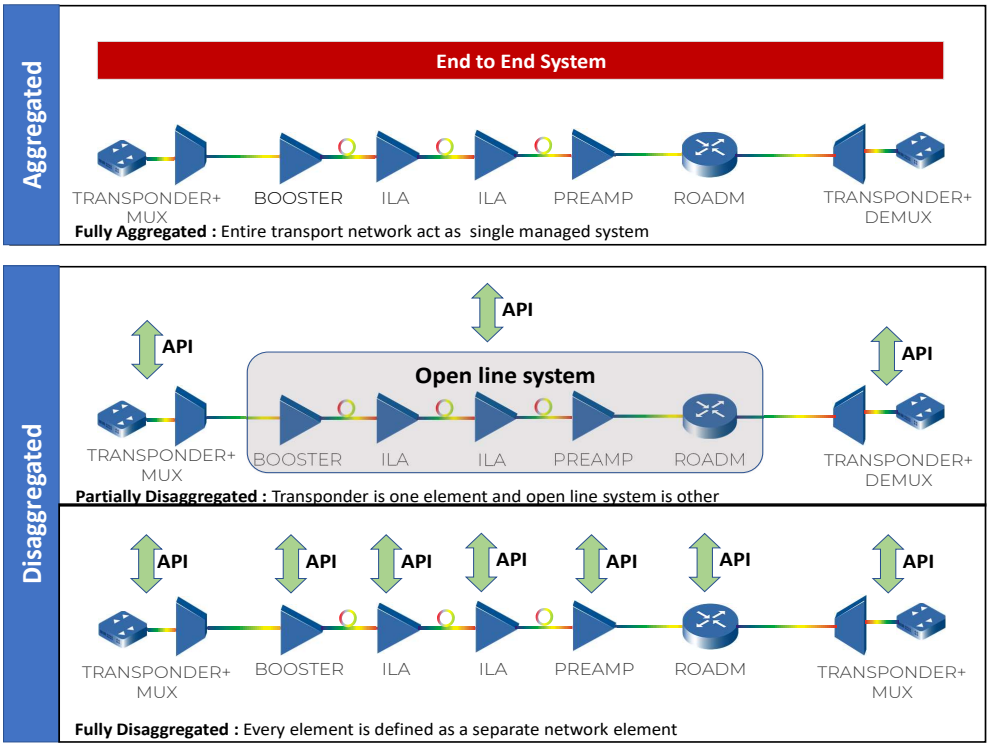


Fig. 1.1 Aggregated and disaggregated networks

transceivers. The links of optical networks, under DD transceivers implementations, operate as close data pipelines, which do not accept any flexibility in configuration [18, 19]. In this framework, transparent LP routing was restricted and must be configured during the network design stage. Besides this, the optical-electro-optical (OEO) traffic regeneration in nodes was strongly driven because of these intrinsic technological constraints. Consequently, the required transparency and elasticity at the logical network layer were not assured at the transmission layer.

In the previous decade, the massive advancement in the agile digital-to-analog (DAC) and analog-to-digital (ADC) converters, in conjunction with the innovation in DSP modules, has aided a reasonable degree of flexibility to the optical transmission system [20]. Additionally, using electronic dispersion compensation (EDC) [21, 22], and DSP enabled equalization [23], the receiver can swiftly compensate all kinds of linear propagation effects, mainly chromatic dispersion (CD) and polarization mode dispersion (PMD).

These technologies paved a road map for developing DSP-based transceivers, starting a new era of flexible optical transceivers [24, 25] that can handle adaptive

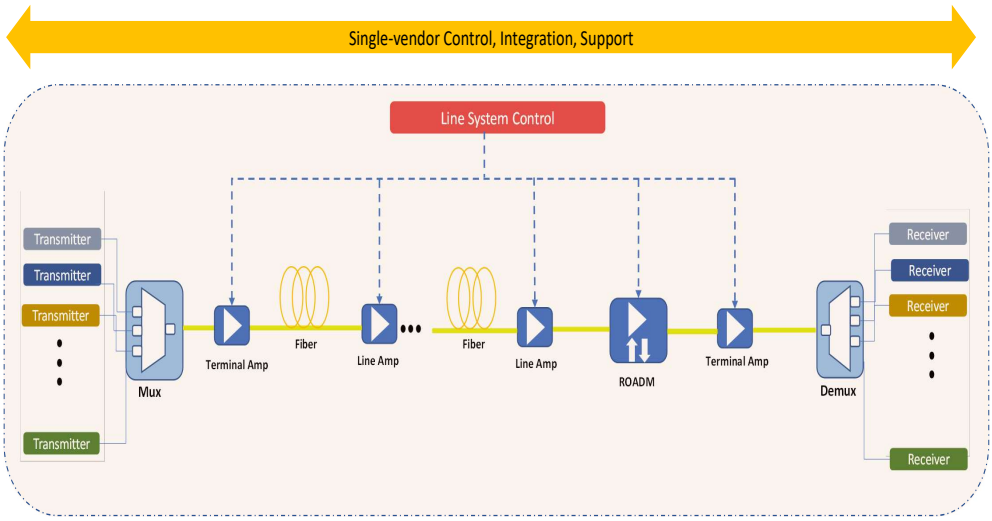


Fig. 1.2 Single-vendor legacy line system.

modulation formats and get rid of DCUs from the optical links of a network without adding any extra penalties [26]. The removal of DCUs from optical links has added extra flexibility by enabling transparent wavelength routing. This ability ultimately made a remarkable improvement in the ROADMs architecture, which can now deliver reconfigurable and transparent optical networking [12].

Moreover, the new era of DSP-enabled transceivers allowed another leap by opening the possibility of a new paradigm of disaggregation in the optical communications system, by decoupling the line system (mainly optical fibers and amplifiers) from line terminal (transceiver) shown in Fig. 1.1. This disaggregated optical network system also enables the optimization of the total cost of network infrastructure and removes the barrier of vendor lock-in by introducing the concept of open-line systems [27] and optical white boxes [28].

The single vendor legacy line system is represented in Fig. 1.2. In these vendor-locked systems, a pre-defined vendor controller was used for managing the optical line system and line terminal. This classical vendor lock-in system did not provide much margin and flexibility to the network operators for customization. The operator should entirely rely on the vendor-supplied management software at each operation level, limiting flexibility during the design and management of optical networks. During the last decade, this single vendor traditional optical networks evolved towards open and disaggregated optical networks, pictorially depicted in Fig. 1.3; an open and disaggregated optical network decouples the OLS and line terminal. Fig. 1.3 rep-

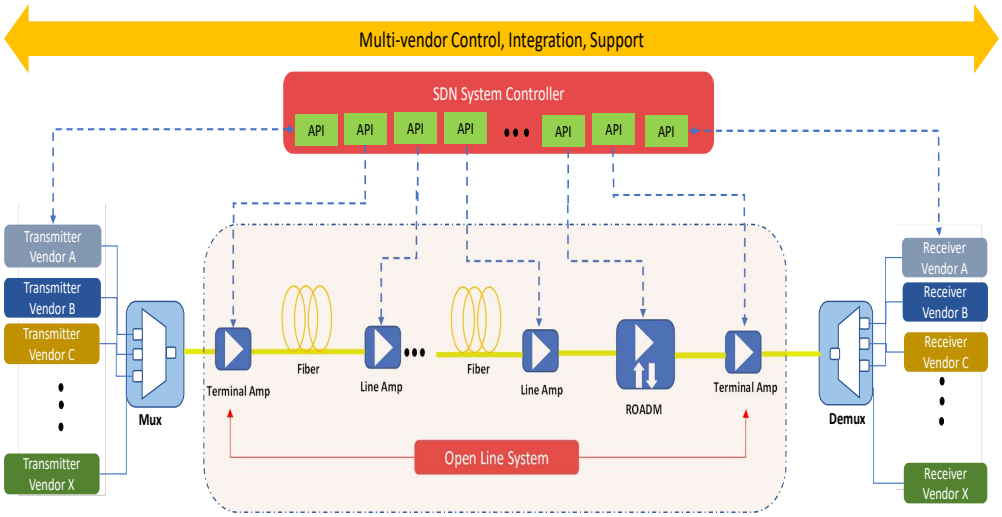


Fig. 1.3 Multi-vendor software-defined open line system.

resents software-defined open line system, where each NE is independently managed and operated through open application programming interfaces (APIs).

On top of these software interfaces, the network operating system can be controlled and orchestrated by a centralized software-defined networking (SDN) controller that is quite flexible; each network operator can customize its core engine based on its requirements. The use of generic interfaces for both control and management allows multi-vendor operation in the same network. For the development of generic interfaces and data models, several studies are taking place from the last couple of years, such as OpenConfig (supported by Google) [29], the Telecom Infra Project (TIP) (supported by Facebook) [30], and OpenROADM (supported by AT&T) [31]. In addition to SDONs, a new technology called spectrally elastic optical networks (EON) was proposed in 2008. The EON provides extraordinary flexibility and scalability in spectrum allocation compared to the traditional fixed grid optical networks [32].

In this softwarized and flexible architecture, the EONs provide flexibility to the network controller to scale up or down resources according to the traffic requests in order to utilize the available spectrum [3, 33] efficiently. At the same time, the SDN implementation enables the management of each NE within a virtualized environment, permitting a disaggregated approach to the network, enabling openness and virtual network slicing. Thus, the modern state-of-the-art optical networks are genuinely flexible, reconfigurable, and softwarized. In this work frame, the optical

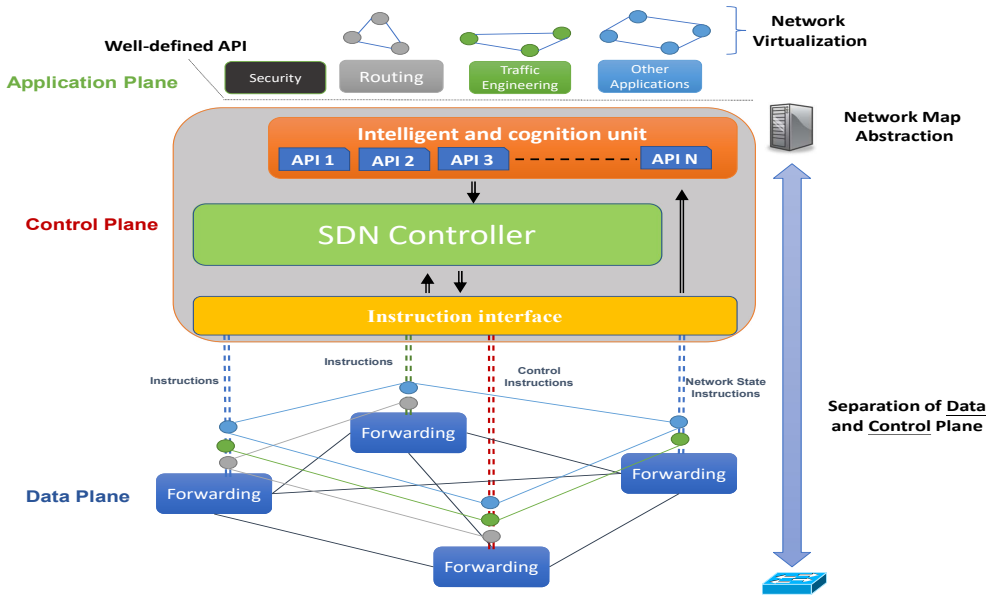


Fig. 1.4 The general software-defined intelligent optical network architecture.

networks are exceptionally dynamic, and it is challenging to manually control and manage the networks. This raises high demand to introduce a vendor-agnostic and intelligent network management system that can autonomously manage and automate different network operations. To this end, it is necessary to introduce a reasonable degree of intelligence in the network operating system, enabling the network to learn and make decisions in real-time.

1.4 Software-defined intelligent optical networks

Intelligent optical networks have a *smart* management and control system, which automatically learns from the present network state and uses this learned knowledge to decide how to adapt the network controls to optimize the overall network performance and enable a quick recovery in case of network failure. The enabling cognition building block is a component of the SDON control plane depicted in Fig. 1.4. The state-of-the-art SDN framework decouples the control and data planes; the cognitive module is a part of the control plane [34]. The basic framework of SDN comprises three principal planes; data, control, and application plane.

The data plane is typically known as an infrastructure plane and is the primary level in the SDN architecture. The data plane is responsible for processing and delivering packets with local forwarding devices such as physical and virtual switches. The employed physical switches are hardware-based, typically implemented as open network hardware like NetFPGA [35]. Similarly, virtual switches are software-based abstractions of hardware switches operating on general operating systems like Linux. The three most well-known implementations of virtual open switches are Open vSwitch [36], Indigo [37], and Pantou [38].

On the other hand, the control plane is mainly the *brainpower* of the SDN architecture, that can perform several functionalities like controlling and managing network resources, dynamically revising forwarding information, and operating the network organization. The key component of the control plane is the centralized logical controller [39–43], which in practice acts like the networking operating system (NOS). The control plane embedded controller acts as a mediator which can control all types of communication between forwarding tools and applications. On one side, the controller exploits the network state information to provide an abstraction of the data plane to the application plane user. On other side, the controller interprets the request generated at the application plane and transfer it to the data plane for forwarding. In addition, the controller also performs other important functionalities, such as routing and traffic engineering, storing information about the network topology, configuring the devices by setting the required working point, etc. The SDN controller usually exploits three different communication interfaces that enable it to interact with different interfaces such as southbound interfaces, northbound interfaces, and eastbound/westbound interfaces.

- **Southbound interfaces**: The southbound interfaces are operated to connect the control plane with the data plane. They enable the data plane forwarding components to interchange the information about network states and control policies with the controller of the control plane. Nowadays, OpenFlow [44], introduced by open networking foundation (ONF), is the leading and well-known southbound open standard interface. Other southbound interfaces are also developed in the last few years, like protocol-oblivious forwarding (POF) [45], network configuration protocol (NETCONF) [46], and OpFlex [47], etc.

- **Northbound interfaces**: The northbound interfaces are operated to connect the control plane with the application plane. The application plane utilizes the

northbound interfaces to take advantage of the network abstraction supplied by the control plane to covey the information about network operation and assist in the automation and flexible management of SDN networks. Recently, the ONF has defined the standard northbound interfaces and a generic communicating architecture [48].

- **Eastbound/Westbound interfaces**: The eastbound/westbound interfaces operate between different controllers of large scale SDN networks having more than one controller. During the implementation of large-scale SDN based networks, multiple network controllers are deployed to process the massive data flows. To better schedule network operations, the controllers operating for different networks communicate using eastbound/westbound interfaces such as SDNi [49], communication interface for distributed control plane (CIDC) [50], and east-west bridge [51], etc.

Finally, the application plane is the higher level in the SDN framework, responsible for dealing with applications. These applications can offer various kinds of services and provide different platforms for managing different services. Typically, the application plane exploits the northbound interfaces to acquire mandatory information about network states related to a particular service or application. After receiving the required information, the application plane can apply the needed changes to the network operations.

1.5 Outline of the thesis

This thesis proposes a data-driven framework, which exploits the multi-level applications of machine learning (ML) in optical and photonics network. The ML techniques are first used to predict the quality of transmission estimation (QoT-E) of a LP. Along with this, ML algorithms are efficiently used to develop a generic topological and technological agnostic model for the complete management of photonic integrated circuits (PIC)-based switching system.

In this manner, as an opening step, in Chapter 2, artificial intelligence, its different classes, and its multi-layer applications in the modern optical network are reported.

In Chapter 3, a framework of machine learning-assisted optical network management in terms of QoT-E is discussed, and several applications are demonstrated.

Then, in Chapter 4, a framework of machine learning-assisted photonic devices management is presented. Finally, in Chapter 5, the conclusion is given, and possible future directions are reviewed.

The main novelties described in this thesis incorporate:

- A framework for ML-assisted QoT-E of LP in SDON. The proposed framework can be used for planning and also LP path computation
- A framework for machine learning-assisted management of any $N \times N$ photonic switching system; routing level controls prediction and QoT impairments estimation.
- Design and management framework based on machine learning-assistance for any $N \times N$ ultra-wideband (UWB) photonic switching system; QoT impairments prediction and deterministic routing level controls estimation.

Chapter 2

Use of Artificial Intelligence for Open and Disaggregated Optical Networks

2.1 Artificial Intelligence

Artificial intelligence (AI) is the key enabler of mimicking intelligent computing devices or machines to simulate human thinking capabilities and behavior. With the simulated cognitive ability, the machine can perform different logical tasks, which requires perception, learning, and reasoning. The AI-based system typically works in three phases: (i) learn and store understanding, (ii) utilize the already learned knowledge to resolve complex problems, and (iii) develop new understanding via experience during its life span [52]. The applications of AI-based systems are a vast field of study having different subfields such as ML, computer vision, natural language processing, and expert systems. ML is a subset of AI that empowers a computer system to learn without being explicitly programmed. The learned knowledge and understanding can be utilized to perform classification or predictions or other schemes of interest [53].

The term ML is not a new field of study as it was proposed for the first time by "Arthur Samuel" in 1952 [54]. The ML is the study of computer algorithms that can improve automatically through experience and by the use of data [55]. The ML-related algorithms were initially developed in the early 70s. However, due to the lack of computing resources over the past decades, ML algorithms were not

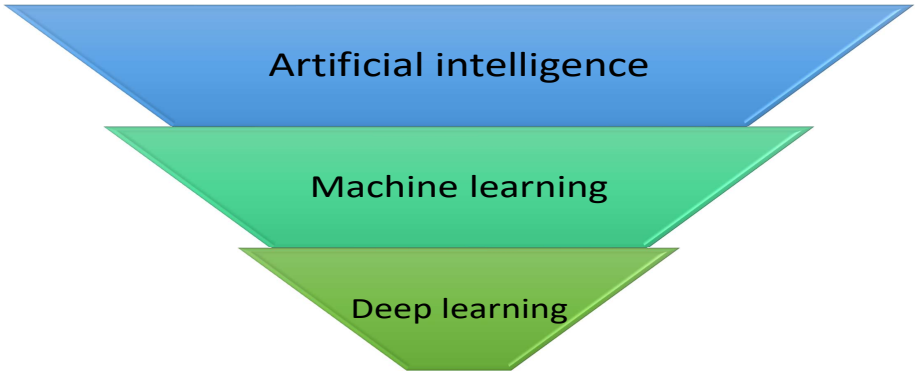


Fig. 2.1 Representing deep learning as a subset of machine learning

practically possible to execute. Typically, ML algorithms need good computational resources and a significant amount of data that provides cognition to the machine to solve the problem. Nowadays, the incredible boost in the silicon integrated chips and the latest evolving technologies, 5G and the IoT provide tremendous computational power and a massive amount of data which revive the ML field. These ML enabling technologies have also given a pathway to initiate the deep learning (DL) technique. DL is a subset of ML (see Fig. 2.1) that demonstrates exceptional performance in different fields or domains [56].

Nowadays, ML has a wide variety of applications and is expanding very quickly with each passing day. In our daily life, without realizing it, we are employing ML while using Google Maps, Google Assistant, Facebook, Amazon, Netflix, YouTube, etc. The ML applications are not limited to these daily life apps, but also have real-world implementations in medical diagnosis, traffic prediction, automated driving, stock market trading, email spam filtering, speech recognition, and computer vision [57–59]. Nevertheless, in this thesis, we make an effort to use different ML techniques for multi-layer open and disaggregated optical networks. Specially, we exploit the applications of various ML schemes and demonstrate their effective operation in optical networks (Chapter 3), and photonics devices (Chapter 4).

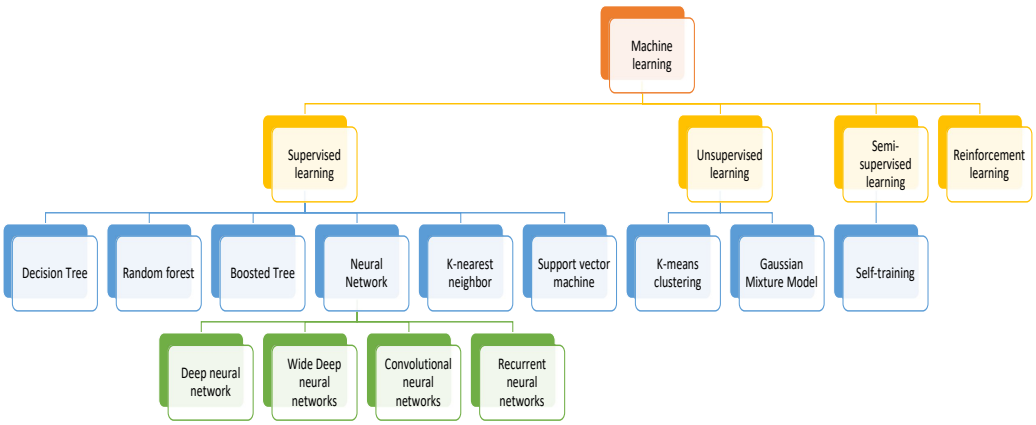


Fig. 2.2 Machine learning classification

2.2 Review of machine learning techniques adopted in optical networks

In the last few years, the evolution towards elastic and SDONs truly motivates the exploitation of ML in the domain of optical and photonics networks. The adoption of these latest technologies makes modern optical networks more flexible and dynamic, opening the possibility for softwarized and automated network operating systems. In this context, by applying ML, it possible to bring automation to network resource provisioning, network optimization while providing smartness to the SDON controller through analysis on the present and previously retrieved network data.

This section's contribution is described in two main folds: (i) an introduction of the main classes of ML and popular ML techniques in each class (see Fig. 2.2). (ii) a review of ML techniques and their operational usage in the area of optical and photonics networks. An additional intuitive assessment of the existing literature is reported, which describes the use cases of ML to the optical networks and how/why/where it plays a notable title role in particular areas of optical networks.

2.2.1 Supervised learning

Supervised learning is an ML approach that manipulates a full set of labeled data during training. In supervised learning, for each input sample F_i, there is the

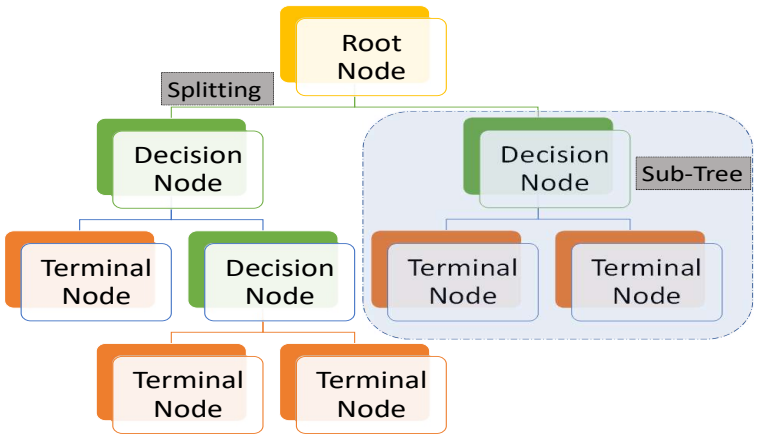


Fig. 2.3 Decision tree

desired label L_i. Supervised learning has two classes based on the labeled data type provided; the labeled data could be discrete (classification problem) or continuous (regression problem). The main goal of supervised learning algorithms is to train the model that can classify the data or predict results correctly. The full set of labeled data provided to supervised learning algorithms is separated into three chunks with a different ratio; a training set (use for training the model), a validation set (use for validating the model), and a test set (unknown samples which are used for prediction or classification). In (Chapter 3), and (Chapter 4) this ML approach is explored for the optimization of some novel regression problems in optical networks and photonic devices, respectively.

2.2.1.1 Decision tree

The decision tree belongs to the class of the supervised ML approach. It provides direct relationships between the input and response variables [60]. A decision tree constructs a tree based on several decisions made by analyzing different aspects of data set features and eventually leads to a response variable. The construction of the typical tree starts from a root node, where the inputs are provided. This root node is further split based on certain conditions into decision nodes. This procedure of splitting a single node into several nodes is called splitting. The nodes that cannot be split up into further decision nodes are called leafs or terminal nodes, as shown in Fig. 2.3. The decision tree has two main basic tuning parameters: *minimum samples leaf* (minimum number of samples required to be at a leaf node)

and *maximum depth* (longest path between the root node and the leaf node). The minimum number of samples required to be at a leaf node

A standard regression cost function representing the mean square error (MSE), used for most of the regression problem is as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y - y')^2 \tag{2.1}$$

Here y, denotes the ground truth (actual value) while y' represents the predictive value. N represents the total number of test samples.

2.2.1.2 Random forest

The random forest regressor is a type of supervised ML algorithm that creates an *ensemble* of regression trees using a *bagging* technique [61]. *Bagging* creates various subsets from the training data set chosen randomly with replacement. The random forest is a step extension over *bagging* because it not only takes a random subset of data but also a random selection of features rather than using all the features to train several decision trees. The final outcome of the random forest is made by merely averaging the predictions of each decision tree shown in Fig. 2.4. The two main tuning parameter of random forest are *minimum samples leaf* and *maximum depth*.

2.2.1.3 Boosted Tree

The boosted tree is also a type of supervised ML algorithm that creates an *ensemble* of regression trees using the *gradient-boosting* technique. It works by combining various regression trees models, particularly decision trees, using *gradient-boosting* [62]. Boosted tree model can be represented using Equation 2.2, where the final regressor f is the sum of simple base regressor r_i shown in Fig. 2.5. In Fig. 2.5, a boosted tree mechanism is shown in which several base trees are combined to get a full boosted regressor.

$$f(x) = r_0 + r_1 + r_2 + ... + r_i \tag{2.2}$$

Similar to other tree approaches, boosted tree also has two main tuning parameters; *minimum samples leaf* and *maximum depth*. Additionally, the configuration parame-

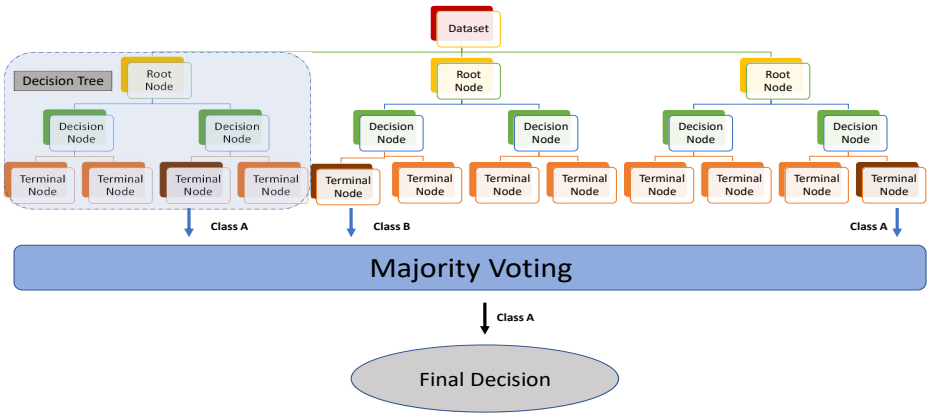


Fig. 2.4 Random forest

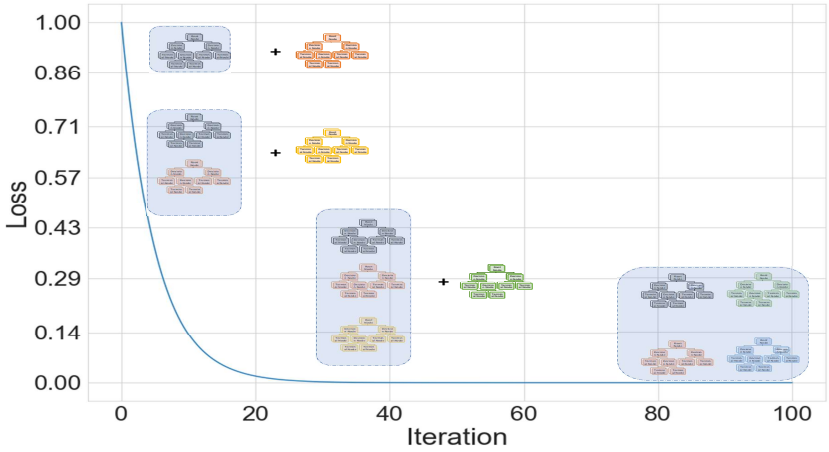


Fig. 2.5 Boosted tree mechanism

ters such as *learning rate* along with L_1 regularization is also considered in some cases.

2.2.1.4 Neural network

Neural Network (NN) is an ML model inspired by the human nervous system to process information. It comprises the input, hidden, and output layers, where the layers are sets of neurons. NN typically learns with a feedback process where the predicted output is compared with the actual output. The difference between them is then calculated. The error gradient is computed for every preceding layer using a

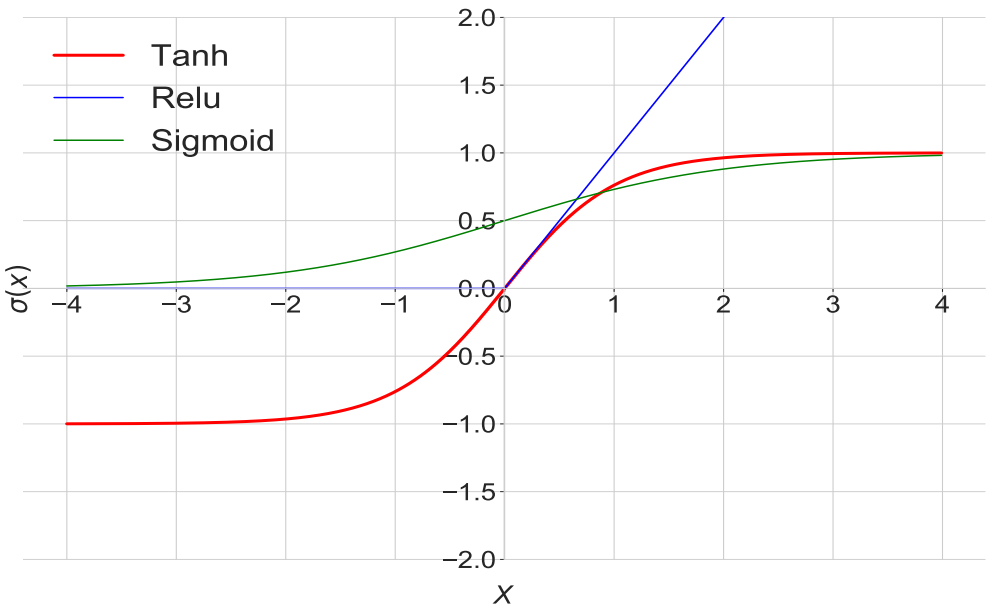


Fig. 2.6 Neural network activation functions common examples

back-propagation algorithm to adjust the weights using a stochastic gradient descent algorithm.

- **Activation function:**
  The typical NN is made of several interconnected neurons at different layers. The neuron, the primary cognitive unit of NN, is characterized by its weight, bias, and activation function. The neuron uses the weights (w) and biases (b) to perform a linear transformation on the given input (x).

$$y = (x \cdot w) + b \tag{2.3}$$

  On top of this linear transformation, the activation function is applied, which operates nonlinear transformation to the x.

$$y = \text{Activation}(\sum(x \cdot w) + b) \tag{2.4}$$

  Introducing this particular non-linearity enables the NN model to learn the complex patterns from the given x. So the selection of activation functions has a significant prominent role in the definitive performance of NN.

$$\text{Sigmoid} : \sigma(x) = \frac{1}{1+e^{-x}} \tag{2.5}$$

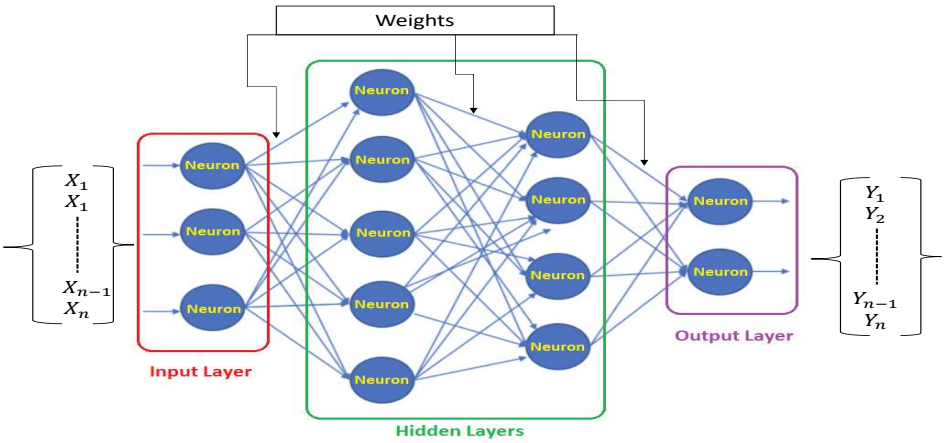


Fig. 2.7 Deep neural network architecture

$$\text{Tanh} : \sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.6}$$

$$\text{Relu} : \sigma(x) = \max(0, x) \tag{2.7}$$

Traditionally, sigmoid and hyperbolic-tangent(Tanh) are widely utilized non-linear activation functions for the neurons augmented in the hidden layer of the model. Nevertheless, in the modern community of data sciences, the rectified linear unit (Relu) has turned out to be the most common alternative as an activation function. The typical graph representation of the three reported activation functions is shown in Fig. 2.6.

The main inefficiency in the sigmoid and Tanh non-linear activation functions is the rapid vanishing of their gradients as $|x|$ becomes bigger, and thus, the output of the activation function moves toward saturation. As the gradient vanishes, the weights and biases are updated slowly with minimal value, affecting the training of NN adversely. To overcome this gradient vanish problem, Relu is adopted, as the gradient does not fade away with an increase in x, leading to better NN performance [63].

1. **Deep neural networks:**
   The deep neural network (DNN) is a NN having multiple hidden layers connecting the input and output layers [64]. Each particular layer of DNN consists of multiple neurons, the computational and learning unit of neural networks shown in Fig. 2.7. The typical NN has several configuring parameters such

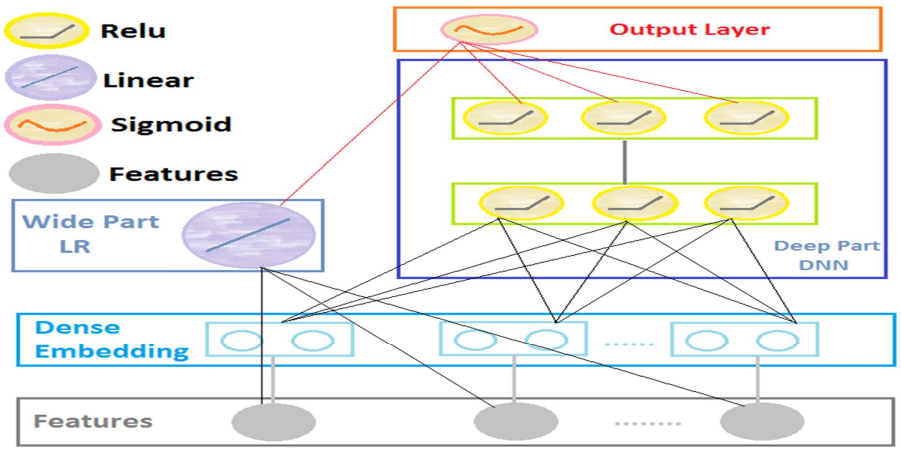


Fig. 2.8 Wide-deep neural network architecture

as *training steps*, *learning rate* and L_1 regularization. The basic function of L_1 regularization is to prevent DNN from over-fitting. In addition to this, non-linear activation functions are used that allows translation of the given input features into the desired prediction labels [65].

Finally, the most important parameter is the number of *hidden-layers*. DNN can be tuned on several numbers of *hidden-layers* and neurons in order to achieve the best trade-off between precision and computational time.

2. **Wide Deep neural networks:**
   The wide-deep neural network (W-DNN) is a type of DNN architecture that combines the strength of *memorization* with *generalization* [66]. Generally, W-DNN is synergically trained on *wide* linear model such as *linear regression* (LR) for memorization and for generalization on *DNN*. The output of wide LR and generalized DNN are combined at the output layer to get the final prediction shown in Fig. 2.8. The output layer is mostly loaded with *sigmoid* function, as *sigmoid* produces activation values in a particular range so that the output layer will always be activated. During the training of W-DNN, the wide and deep parts are jointly trained at the same time.

3. **Convolutional neural networks:**
   The convolutional neural networks (CNN) is a subset of NN and is inspired by the organization of the animal visual cortex. Typically, CNN is a well-acknowledged approach to perform best with image data. CNN comprises three types of layers: convolution (Conv), pooling, and fully connected layers [67].

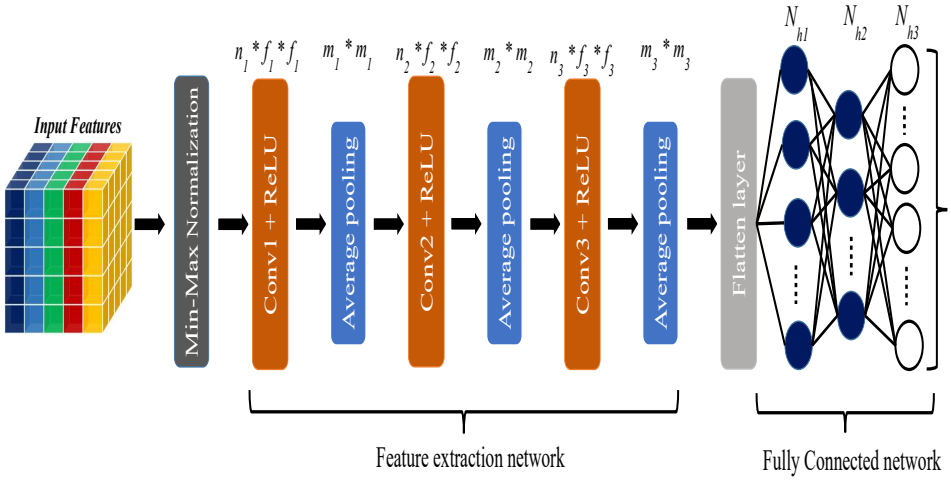


Fig. 2.9 Convolutional neural networks

The Conv and pooling layers are combined to create a feature-extraction network. Conv-layers are used as a feature detector to extract significant features from the input data for better prediction. In addition to this, each Conv-layer mostly uses *Relu* activation function to accelerate the training process. Moreover, the average-pooling layers are placed between succeeding Conv-layers to carry out spatial-pooling. The primary purpose of pooling layers is to lower the spatial size of input feature maps, leading to a reduced number of parameters and computational complexity. It is worth mentioning that the reduction in spatial size is equivalent to the kernel size of the pooling layer that is $N \times N$, and it reduces the spatial size of input features by a factor of N. Thus, each layer produces a compact and informative description of input features. The output of the feature-extraction network is a 3-dimension representation of the input data. Moreover, the flattening layer is used to convert the 3-dimensional representation into a 1-dimensional array of feature vectors, and then this feature vector is finally passed to the fully connected network.

4. **Recurrent neural networks:**
   The recurrent neural networks (RNN) is a leading NN mainly used when sequential training data is available. It is the unique type of NN that remembers its inputs using its inner memory units [68]. The RNN internal memory units enable it to memorize essential information about the given input data. This memorizing feature of RNN assists it in the training process, which

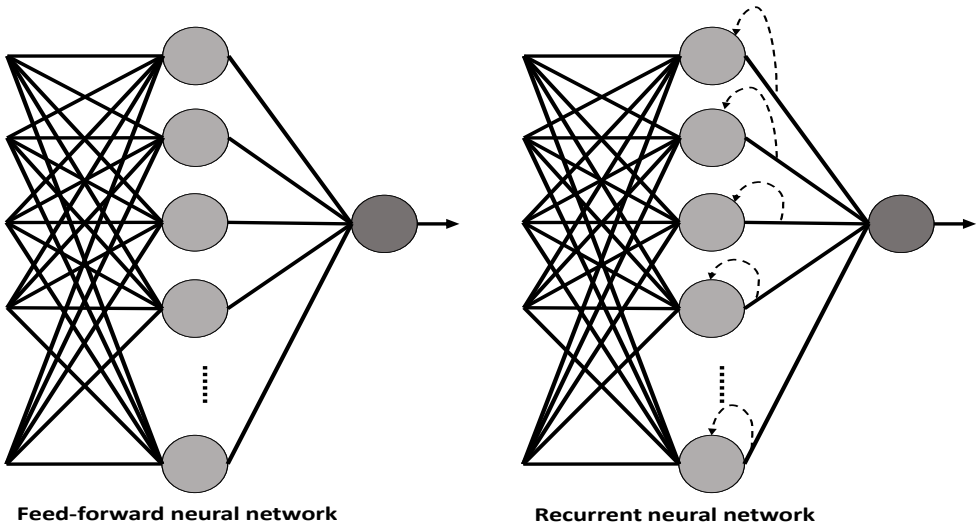


Fig. 2.10 Feed-forward neural network vs. recurrent neural network

consequently increases the prediction capability exploiting sequential data like text and fiscal data, time-series predictions, natural language processing, audio, videography, meteorology, etc. The working idea of RNN is different from feed-forward NN. In a feed-forward NN, the information only goes in a single forward path, starts from the input layer, all the way through the hidden layers, and finally reaches the output layer. The feed-forward NN has no memory unit; as a result, they consider the present input during the training process. Unlike feed-forward NN, RNN has memory elements in which the information cycles through a loop shown in Fig. 2.10. This information running through loops enables RNN to consider the present input and its previous inputs for its training.

2.2.1.5 K-nearest neighbor

K-nearest neighbors (KNN) is a type of supervised ML technique. KNN makes predictions based on feature similarity by calculating the distance between the new data point and training data points. The two main tuning parameters required for KNN are the number of nearest neighbors (K) and the distance metric.

2.2.1.6 Support vector machine

Support vector machine (SVM) is a supervised ML technique that caters to regression problems where continuous output is predicted. The following essential parameters are used to configure SVM:

- **Kernel**: It is used to map data from lower dimensions to higher dimensions at lower computation costs. It is beneficial in finding the best hyper-plane.
- **Hyperplane**: It is a line or decision boundary that can separate n-dimensional space into groups.
- **Support Vectors**: Two parallel lines are drawn with ε distance from the hyperplane to define a margin shown in Fig. 2.11a.

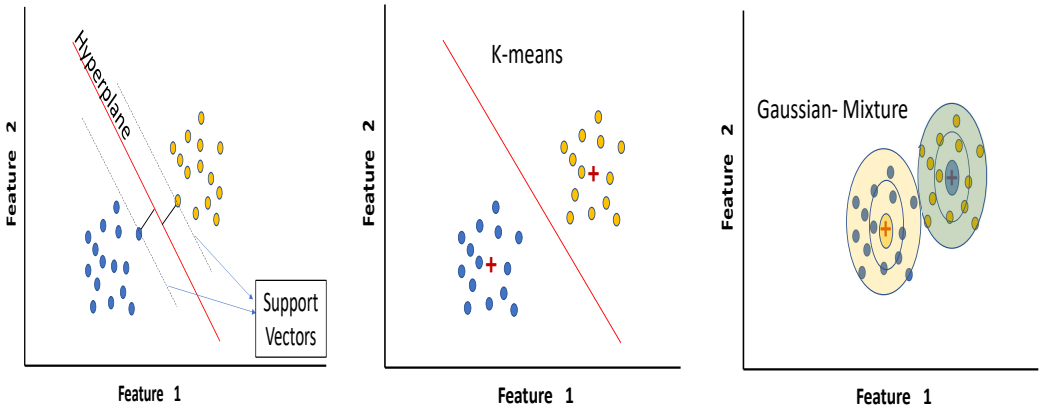
In SVM, the maximum allowable error within a tolerable range is defined by the ε value. The main goal is to find a function $f(x)$ that deviates by a value not greater than ε for each training point from the output prediction. The best fit line is the one with the maximum number of data points.

2.2.2 Unsupervised learning

This unsupervised learning is a subset of the ML technique that does not require any labeling dataset for the training. In general, unsupervised learning is used for three main tasks clustering (categorizing unlabeled data based on their resemblances or differences), association (find relations between variables in unlabeled data), and dimensional reduction (reduces the number of unlabeled data features/inputs in a huge dataset). This thesis only exploits the applications of supervised learning procedures in the domain of multi-layer optical networks.

2.2.2.1 K-means clustering

K-means clustering is the one of the renowned clustering procedure. It is an iterative technique in which data points x(1), x(2), x(3)..... X(N) are split up into K clusters in a specific manner such that the sum of the squared errors for data points contained by a cluster is minimized. In this iterative process, the mid point of



(a) (b) (c)

Fig. 2.11 (a) Support vector machine (b) K-means clustering (c) Gaussian Mixture Model

every single cluster is initially estimated, and then the N data elements are allocated to the group with the minimal Euclidean distance. The next step is updating the center of each cluster by taking the average using the location of the data points placed in that cluster. The iterative computing center and data allocation procedure are repeated until points do not change significantly or a given number of maximum iterations are reached shown in Fig. 2.11b.

2.2.2.2 Gaussian Mixture Model

Gaussian mixture model (GMM) is the most extensively used probabilistic method for clustering. It typically suggests a finite number of Gaussian distributions which exemplify a cluster. After suggesting a particular distribution, it starts gathering the data points which best fit into that suggested distribution. The important parameters used for model configuration are the coefficient of mixing, mean, and covariance of every Gaussian distribution. Assuming a dataset with features F_i; GMM suggests a mixture of K Gaussian distributions (K = number of clusters) having a specific mean vector and variance matrix. The given unknown parameters are determined using an algorithm called expectation-maximization.

- **Expectation-Maximization:** Expectation-maximization (EM) is a statistical algorithm that is famous for finding the appropriate model parameters. Let's assume K_i number of clusters for a particular sample space and in response K_i Gaussian distributions having mean μ_i and covariance $\sum_i$. In addition, the number of points or density of the distribution is symbolized with P_i. Now in

order to find the values of these parameters as to characterize the Gaussian distributions, the EM works in two phases.

- **E-step:** In this step, EM calculates the probability r_i for each data point x_i that it belongs to a specific cluster/distribution.

$$y_i^{cluster} = \frac{p_i \text{ that } x_i \text{ belongs to a distribution}}{\text{sum of } p_i \text{ that } x_i \text{ belongs to } K_i} \tag{2.8}$$

- **M-step:** After the E-step, the complete data is used to update the μ, $\sum_{cluster}$, and P. This is an iterative process that updates the probability for each data point to allocate it to a particular distribution and is repeated to maximize the log-likelihood function shown in Fig. 2.11c.

$$P = \frac{N_{dp}}{\text{total number of data points}} \tag{2.9}$$

$$\mu = \frac{1}{N_{dp}} \left(\mu_i \sum_i y_i^{cluster} \right). \tag{2.10}$$

$$\sum_{cluster} = \frac{1}{N_{dp}} \left((x_i - \mu_{cluster})^t (x_i - \mu_{cluster}) \sum_i y_i^{cluster} \right). \tag{2.11}$$

where N_{dp} is number of data points allocated to cluster.

2.2.3 Semi-supervised learning

Semi-supervised learning is an ML approach that is a hybrid of supervised and unsupervised learning. It generally tackles problems in which most of the sample space is unlabeled; limited total labeled data samples are obtainable. This specific type of learning is mainly practical when labeled data samples are not so widespread or too costly to acquire, but easily accessible unlabeled data can be exploited by this learning method to improve the overall system performance.

The most common type of semi-supervised technique is self-training [69]. It is an iterative method that at first only applies supervised learning with labeled data samples. Later, in each iteration, few of the unlabeled data elements are labeled using the outcomes of the model acquired from the supervised learning method.

After labeling these unlabeled data points, they are combined with the already known labeled data to start the training under a similar supervised learning method.

2.2.4 Reinforcement learning

Reinforcement learning is a goal-oriented ML technique that empowers an agent to learn by exploring different environment states and refining agent learning abilities by using evaluative feedback stated as the reward. Like supervised learning, reinforcement learning also applies mapping between input features and output labels. However, in supervised learning, the feedback supplied to the driving model is in the form of correct controls that help the model learn the behavior with respect to the actual one. Reinforcement learning utilizes rewards and punishments as feedback for positive and negative behavior. Unlike unsupervised learning, reinforcement learning differs due to the objectives. The primary objective of unsupervised learning is to understand resemblances and discrepancies between the given data samples. In reinforcement learning, the target is to achieve an appropriate action model that would maximize the overall accumulative reward of the agent.

2.3 Multi-layer applications of machine learning in optical networks

The modern-day optical networks are deployed with several types of network monitors, which provide different kinds of system information such as QoT in terms of bit error rate (BER), monitoring of traffic, fault alarms and devices temperature, etc. This large amount of network data makes ML more practical to operate for different optical network applications at different layers shown in Fig. 2.12. Typically, ML techniques exploit the available data to train its cognitive engine, and after training, it can quickly solve complex non-linear problems.

2.3.1 Machine learning at physical layer

At the physical layer of an optical network, various functionality effectively demands the use of ML assistance. These typically include evaluating transmis-

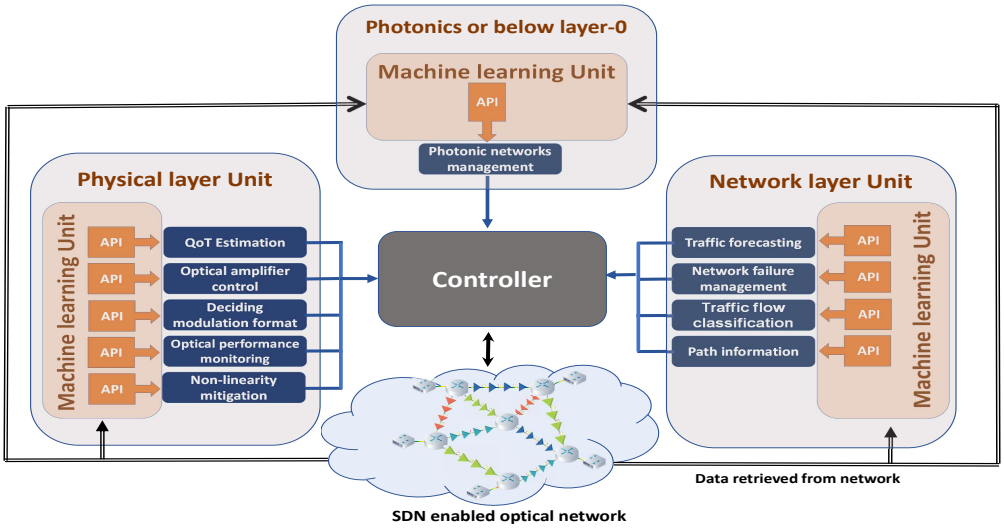


Fig. 2.12 Machine learning applications in optical networks

sion system performance in terms of QoT, which is also a prerequisite for path computation, controlling the optical amplifier gain, modulation format decision, and compensating for fiber non-linearity in several dedicated cases [70, 71]. In the following section of this thesis, a brief explanation of the operational usage of ML techniques applied to the physical layer is reported.

2.3.1.1 QoT-Estimation

The computation of the QoT for the deployment of a new LP in transparent optical networks has techno-economic importance for any network operator. The theory of QoT commonly describes different physical layer parameters, mainly BER/Q-factor or optical signal-to-noise ratio (OSNR). These parameters affect the signal received at a given receiver and provide a numerical gauge to ensure that the level of QoT is enough for the given receiver threshold. In addition to this, the following reported parameters are primarily influenced by numerous transmission variables (baud rate, modulation format, coding rate, LP path computation, etc). Consequently, optimization of such a variety of possible parameters is pretty challenging and makes it almost impossible for the network operator to manually configure all the possible combinations for the deployment of a particular LP.

As of today, the current QoT-E approaches for LP deployment can be divided into two major classes: (i) The vendor-provided data describing the system and

network elements such as static characterization of devices (e.g., amplifier gain and noise figure in the frequency domain, connector loss, etc.) is used to implement an accurate QoT-E in vendor-specific systems. Many analytical models are available for calculating the QoT by using the vendor data that characterizes the OLS components. However, this static data-based approach may not be accurate as the components experience gradual degradation due to aging, leading to progressively unreliable QoT-E after a certain period. (ii) The second approach is based on the telemetry data considering only the present network status. Deducing an OLS agnostic functioning in an open environment, the OLS controller mainly depends upon telemetry data actualizing from the optical channel monitor and the erbium doped fiber amplifiers EDFAs. This particular approach works by utilizing the telemetry of the network's current state to estimate an accurate QoT. In contrast to the previous approach, this method does not rely on the static characterization of the devices' parameters. It thus removes the uncertainty in the QoT-E accuracy due to the device aging factor discussed in the former approach. Nonetheless, the problem with this method is that the QoT response, mainly its OSNR component, depends highly on the spectral load configuration, which leads to a considerable uncertainty in the QoT margin [72]. On the contrary, ML assisted QoT estimator worked in an entirely agnostic way and evolved as a promising method to predict the QoT of LP autonomously.

The ML paradigm has already been effectively used for QoT-E from a different perspective. Few major contributions are reported in this thesis, such as cognitive-case-based-reasoning (CBR) technique is suggested in [73], exploiting various features of deployed LPs such as physical path, total path length, the total number of propagating LP per link and utilize Q-factor as a label. A new traffic request is satisfied by the most similar case of already stored features based on the means of the Euclidean distance in normalized features space. The associated Q-factor depth is compared with a predefined system threshold. The experimental results of [73] are demonstrated in [74], exploiting the data acquired from a real testbed.

A NN model is proposed in [75–78], which uses several features of a given LP such as total length of the LP, the count of EDFAs pass over by LP, the overall length of the link, destination node degree, and the number of channels accommodated by the LP, while the exploit label predicts the Q-factor. The training of NN is done using real-time network data to enable the predictive model consecutive updates. In [79] random forest binary classifier is proposed, which gets features set that

incorporates the number of links, the total length, longest link length of the LP, the transmitted traffic volume, and the operated modulation format. The classifier model characterizes the probability of a certain BER of unestablished LP in terms of exceeding the system threshold. Similarly, in [80], the authors proposed a random forest classifier together with KNN and SVM to discover the best classifier in terms of QoT label. The analysis results in [80] shows that the SVM performs better than the other two, although with a more expensive computational cost.

Furthermore, in [81], the authors proposed a ML-based technique called Gaussian processes non-linear regression (GPR) that experimentally demonstrated that the proposed GPR is trained on data acquired for a particular set of optical WDM (24GBaud QPSK) network configurations. The trained model is then tested to predict some other set of network configurations. The manipulated features used for GPR training are input signal power, symbol rate, inter-channel spacing, and transmission length to predict accurate BER. Lastly, in [82], the author experimentally demonstrated the feasibility of integrating an intelligent QoT estimator in the control of a real testbed.

2.3.1.2 Optical amplifier control

The present-day internet traffic is mainly dynamic and heterogeneous; consequently, it needs more flexible and dynamic LP deployment. The dynamic system can add or drop LPs very fast, and thus it is challenging for the network operator to sustain physical-layer stability. In this context, the most challenging task is the real-time configuration of EDFAs, whose power profile is strongly dependent on spectral load. For example, when a new LP is turned on or when an existing LP is turned off, substantial irregularity with the signals power is observed, particularly with the presence of multiple EDFAs. Hence, autonomous and self-control preamplification of signal power is needed to prevent unnecessary power inconsistency after EDFA amplification.

In recent years ML algorithms have emerged as the most promising procedure for flexible tuning of the EDFA working point based on the signal input power. In the studies like [83–87], the authors experimentally demonstrated different cognitive and ML methods for controlling different EDFA parameters (noise figure (NF) and gain (G)) by adjusting its operating point in its operating region. In [83], a cognitive procedure is suggested, which is employed in dynamic network environments. A

database of amplifier gains of already deployed LPs is collected, along with LP characteristics such as the links count, overall link lengths, and finally the receiver-side OSNR. During the new LP request, the most appropriate in LP characteristics are retrieved from the database along with associated gains profiles, and a fresh alternative of gains is produced by perturbing the retrieved values. Later, the OSNR is estimated with the new gain profile and put in the storage of the existing database. Finally, the gain profiles related to the maximum OSNR are used for configuring the amplifier gains for the requested new LP deployment. In [84], real-time EDFA operating point management is performed using cognitive gain control via the generalized multi-protocol label switching (GMPLS) for a single amplifier, which is extended to cascaded amplifiers in [87]. In [85, 86], the authors utilized NN over an experimental setup with single and cascaded amplifiers to characterize the EDFAs operating point with interpolation errors below 0.5 dB.

2.3.1.3 Deciding modulation format

The present-day optical networks are rapidly evolving towards the flexible grid/EON infrastructure. The EON paradigm has discovered a unique optical network architecture that provides LPs based on the actual traffic demands. This flexibility makes the LP provisioning problem more challenging than traditional fixed-grids WDM. The complexity in LP provisioning is mainly due to adapting bandwidth utilization, frequency of carrier, and modulation format, which enables the transmission of LP on the necessary bit rate and total distance. Implementing an adaptive modulation format also made the coherent optical transmission more complex as it is not always possible to decide modulation format (DMF) in advance.

The use of ML algorithms assists the coherent transmission in different ways. In [88], the authors assess the performance of six unsupervised clustering methods to differentiate among five different formats (BPSK, QPSK, 8-PSK, 8-QAM, 16-QAM) mainly on confusion matrix terminology and the received-side OSNR. The authors in [89] proposed a Bayesian EM technique to calculate the different possible clustering groups in the defined Stokes space description of the signal received to detect the modulation format. The authors in [90, 91], proposed a NN model which exploits amplitude histograms' as a feature in order to differentiate between different modulation formats such as PM-QPSK, 16-QAM, and 64-QAM with a 0% error rate in differentiating. In [92], a NN is synergically used along a genetic algorithm to

further refine the classification ability of NN models. The results in [92] show the classification among six distinct modulation formats (NRZ-OOK, ODB, NRZ-DPSK, RZ-DQPSK, PM-RZ-QPSK, and PM-NRZ-16-QAM).

2.3.1.4 Optical performance monitoring

The modern-day optical networks are deployed with several network monitors, especially performance monitors, which provide different system informations. Optical performance monitoring (OPM) is broadly considered a leading empowering technology for SDN. Through OPM, an SDN controller can guarantee robust and reliable networking. Typically OPM reports the system performance and transmission parameters like BER, Q-factor, CD, PMD, during the total lifetime of the LP. The SDN controller exploits the knowledge about the recorded parameters and can be used to perform different networking functions, such as adapting signal launch power and modulation format, varying data rates, rerouting, and spectrum assignment of LP, etc. The traditional OPM procedures are not as sophisticated as to allow monitoring of multiple transmission parameters simultaneously and independently, mainly due to the difficulty in separating analytically the effects of different impairments. Consequently, a significant number of monitors are needed to capture the effect of each transmission impairment, causing an increase in overall system cost. An additional critical problem of conventional OPM devices is that their efficient positioning is required to obtain system information.

In recent years, ML-based models are proposed as the potential enabler for multi-purpose monitoring tools in optical networks with very low cost. The majority of ML-based OPM methods use transmission variables like baud rate, OSNR, CD, PMD and polarization-dependent loss (PDL) as features for the training of its cognitive engine. Different approaches are used in the literature to extract the features for the training of ML agents. In [81, 93–96], the author fed the NN with features such as Q-factor, variance, root mean squared jitter (RJRMS), and crossing level amplitude, extracted from the power eye diagrams. Similarly, in [97], an eye-diagram and phase portrait with two-dimensional representation was used for the same purpose. In [94], asynchronous constellation diagrams are also included to take into account the symbols transition. In [95] [96], histograms of the asynchronously sampled signal amplitudes are also used for the feature extraction process. The extraction of the feature before providing it to NN reduces the complexity of the model and also

requires few realizations of data to train the model. On the contrary, in [98], [99], the features are not extracted before feeding into the NN, so the author used DL architecture to extract the features using more hidden layers of NN.

2.3.1.5 Non-linearity mitigation

The decrease in the strength of the optical signal over a certain distance is mainly due to the attenuation it suffers during fiber propagation. To overcome this limitation, optical amplification needs to be carried out after a certain length as to increase the transmission distance. During this operation, the optical amplifiers increase not only the signal power but also boost the noise, which results in amplified spontaneous emission (ASE). Along with this, the transmission through the fiber introduces non-linear effects such as the Kerr effect [100]. In the transmission set-up, the penalty in fiber propagation due to attenuation, CD, and non-linear Kerr effect can be approximated as additive Gaussian disturbance [101–103] and is typically called as non-linear interference (NLI). The NLI mainly affects the correct symbol detection at the receiver. Generally, due to the circular symmetrical Gaussian distribution of noise, by minimizing the Euclidean distance, optimal symbol disclosure is achieved among the received symbols r_s, as they have smooth linear decision boundaries. On the contrary, with memoryless non-linearity, the associated noise with the received symbols r_s has no circular symmetry. This non-linearity strongly affects the constellation cluster diagram, as it becomes elliptical, substituting the simpler circular symmetry. This specific case makes correct symbol detection much more challenging, as the optimal symbols recovery stategy is no more based on the Euclidean distance minimization. The use of ML methods such as SVM, KNN, GMM, and kernel density estimator assists in the optimal symbol recovery at the receiver end.

In [104], the initial propagating power in the fiber has attained a gain of almost 3 dB by engaging GMM together with EM toward 14Gbaud DP 16-QAM transmission across a dispersion compensated fiber link of 800 km. In [105], the KNN classifier is proposed to compensate network penalties in dispersion managed, dispersion unmanaged and zero-dispersion links, under modulation scenario implementing 16-QAM transmission. In [106], NN is adopted for achieving nonlinear equalization scheme at 16-QAM OFDM transmission system. Similarly, in [107], a NN network-assisted equalizer reduced the computational cost using generalized matrix inversion.

SVM technique is proposed in [108, 109]. In [108], fast Newton-based SVM is adopted to decrease inter-subcarrier mixing in 16-QAM OFDM transmission, while in [109], SVM classifier is managed to classify decision boundaries of an M-PSK constellation. In [110], GMM is proposed to outplace the traditional decoding unit, while in [111–116], the authors proposed NN for equalization. The feature used as an input to NN is a vector of received symbols, and the label or output is the equalized optical signal, having lowered inter-symbol interference (ISI). In [117], the k-means clustering method is used to propose an effective equalization technique with minimal complexity for a 64-QAM coherent transmission system.

Table 2.1 Summary of Machine learning Applications at Physical layer

Machine learning Application	Machine learning Technique	Features/Input	Labels/Output	Data type	References
QoT-E	CBR	Physical path, total path length, total number of propagating LP per link	Q-factor	Synthetic	[74]
				Real	[74]
	NN	Total LP length, count of EDFAs crossed by LP, the total length of link, destination node degree, number of channels accommodating by LP	Q-factor	Synthetic	[79], [76]
				Real	[77]
	NN	Source node plus destination node, link occupancy, modulation format, length of path, data-rate	BER	Real	[78]
	Random forest	Number of links, total length, longest link length of the LP, traffic volume, operated modulation format	BER	Synthetic	[79]
	Random forest, KNN, SVM	Launch power of signal, modulation format, data rate, total link length, span length	BER	Synthetic	[80]
	GPR	Input signal power, symbol rate, inter-channel spacing, and transmission length	BER	Synthetic	[81]
OAC	CBR	Number of links, total link length, NF, G-flatness	OSNR	Real	[83]
	NN	Input and output power of EDFA	EDFA working point	Real	[85], [86]

	CBR	Input and output power of single EDFA	EDFA working point	Real	[84]
	CBR	Input and output power of cascaded EDFA	EDFA working point	Synthetic	[87]
DMF	Clustering algorithms	Stokes space parameters	Modulation formats	Real	[88]
	Bayesian EM	Stokes space parameters	Modulation formats	Real	[89]
	KNN	Received symbols	Modulation formats	Real	[118]
	NN	Amplitude histograms	Modulation formats	Real	[90], [91]
	NN	Amplitude histograms	Modulation formats	Synthetic	[92]
OPM	NN	Q-factor, variance, closure, RJRMS, and crossing level amplitude (extracted from the power eye diagrams)	OSNR,PMD,CD	Synthetic	[93–95]
	Deep learning	Eye diagrams and amplitude histogram parameters	OSNR,PMD,CD	Real	[98]
	NN	Q-factor, variance, RJRMS, and crossing level amplitude (extracted from the power eye diagrams)	Modulation format	Real	[81]
	Kernel based ridge regression	Eye diagrams and phase parameters	PMD,CD	Real	[97]
Non-linearity mitigation	GMM together with EM	Received symbols	OSNR	Real	[104]
	KNN	Received symbols	BER	Real	[105]
	NN	Received symbols	Q-factor	Synthetic	[106]
	NN	Received symbols	Self-phase modulation	Synthetic	[113]
	Fast newton-based SVM	Received symbols	Q-factor	Synthetic	[108]
	SVM	Received symbols	Decision boundaries	Real	[109]
	GMM	Equalized symbols	Decoded symbols	Real	[110]

	NN	Equalized symbols	Equalized signal with lowered ISI	Real	[111–116]
	K-means	Received constellation	Constellation clusters	Real	[117]

2.3.2 Machine learning at network layer

Like the physical layer, ML also shows a promising role at the network layer, such as assisting new LP provisioning or restoration of the already deployed LP, network failure detection and localization, also in some cases, advance forecasting of failure, efficient traffic classification to meet QoS and SLAs, etc [70, 71]. In the following section, few applications of ML at the network layer are reported.

2.3.2.1 Traffic forecasting

During the optical network design phase, the accurate traffic forecast, decreases the over-provisioning of LPs to a reasonable extent. Moving towards the operation phase, the network operators can effectively utilize the network resources by doing traffic engineering with the help of the available real-time data. As the ML techniques are data-driven, they could be implicitly used for the prediction of traffic, which is a fundamental task both in the planning and operational design of optical networks.

In [119, 120], the authors proposed a supervised learning technique such as auto-regressive integrated moving average (ARIMA) [121] to predict the traffic in the context of virtual network topology (VNT) design and configuration. A network planner and decision-maker (NPDM) unit based on ARIMA models is used for predicting traffic for the configuration on VNT. Similarly, in [122] and [123], the authors also proposed traffic prediction models for the configuration of VNT exploiting NN. The proposed NNs are fed with the available source-destination traffic matrix. The NN predicts another traffic matrix which is then applied to a decision-maker unit (DMN). The DMN decide whether the VNT requires to be reconfigured or not. The author in [124], proposed a DL technique for the traffic estimation and resource allocation for an intra-datacenter network. The results in [124] show that DL outperforms the simple NN technique in terms of efficiency but is costlier in terms of complexity. In [125] the authors proposed a cognitive network management unit based on ML algorithm for the traffic prediction matrix used mainly for VNT

configuration. However, like the previous reference, the authors in [126] proposed Bayesian inference to predict the traffic for VNT configuration.

In [127] the authors adopted a general structure for traffic modeling using call data records (CDR). Different clustering-based algorithms such as non negative matrix factorization (NMF) and collective NMF (C-NMF) are used to obtain beneficial augmented relation in the real datasets provided by network operator. The insight information is used to enable the network operators to utilize their network assets better. In [128] and [129] the authors mainly adopted a genetic algorithm (GA) for virtual topology design and configuration. Additionally, they used reinforcement learning to update the fitness function of GA.

2.3.2.2 Network failure management

The network operator always desires to detect and pinpoint the event of a failure in advance. This advanced detection of failure is critical for the operators because it avoids any network out-of-service (OOS), allowing to meet the service level agreements (SLAs). Failure management of a network can be done in three levels; (i) The initial level is the detection of failure this is generally digging up the set of LPs that are disturbed by the occurrence of failure; this enables network operators to only reroute the traffic of the affected LPs. (ii) Failure localization is the second level that mainly assists in launching the recovery process that tries to re-maintain the pre-failure network status. (iii) Final level is to discover the actual cause of failure, such as short-term traffic congestion, devices malfunctioning, and, in some cases, abnormal behavior of network management system (NMS). This is useful for taking the necessary steps to avoid future failure and making suitable restoration procedures.

The modern optical network has numerous monitoring systems that continuously produce the network status history. This information is adequate to empower data-driven techniques for different applications such as failure management. The ML-based systems can be involved to detect the failure location and specify the type of failure. The authors in [130], proposed the kriging method to locate the correct place of malfunction along all parts of the network. In [130], the already available data of deployed LP having some failure records are exploited to locate the failure of all other LPs operating in a network. In [131], the authors adopted a Bayesian network that locates a failure with the LPs and labels the leading cause of failure.

The authors exploited the real field measured BER and the power of LP received at the ending nodes as an input to the suggested Bayesian classifier. The performance of the Bayesian classifier demonstrates that only 0.8% of the tested cases were wrongly classified. In [132], the author considered a multi-layer gigabit passive optical network (GPON)/fiber to the home (FTTH) network. The physical network topology consisting of optical network terminals (ONTs), optical network units (ONUs), and fibers are indicated as layer-1. The optical device layer is termed layer-1, where the malfunctioning or failure between the devices is shaped in layer-2 utilizing acyclic directed graphs connected through layer-1. In [132], a Bayesian network and EM algorithms are proposed to quantify the uncertainties at layer-2 failures using the approximation of conditional probability distributions of the subset of correct operational data. However, another subset is those statistics that are either not reported or have incorrect measurements. The proposed scheme is used to estimate this subset of data to identify the cause of failures in the network and assist in self-pronouncement. Similarly, in [133], a combination of the Bayesian model and EM is employed for fault identification in GPON/FTTH infrastructures. In [134], the authors adopted a synergic regression and classification technique and proposed two algorithms termed BANDO and LUCIDA. The authors first adopted BANDO running on each node of the network for the abnormal BER detection. The BANDO takes the previous information related to LPs BER and exposes any unexpected shifts in the BER level that may happen due to some device failure. After getting the possible notification from BANDO and taking the preliminary information related to LPs BER and power, the LUCIDA running on the network controller classifies failure.

Similarly,in [135], two smart algorithmic rules are presented, named Testing OptIcal Switching at connection SetUp time (TISSUE) and FailurE causE Localization for optIcal NetworkinG (FEELING). The first algorithmic rule, TISSUE gets the estimated BER of LP at every traversing node of a network toghether with the theoretical BER. It initiates assessing the differentiation of the estimated BER slope and theoretical BER slope. If the difference exceeds a particular threshold, a failure is predicted. While the FEELING algorithm exploits the decision tree and SVM for failure localization. The FEELING multi-class classifier takes the features such as the power readings around the central channel frequency and also the power reading along with other cut-off points of the spectrum. The decision tree and SVM classify the failure cause localization. The authors in [136] also proposed several

ML algorithms for failure detection and cause classification. The ML models such as SVMs, random forests, and neural networks are trained on the data related to BER. The results in [136] showed the failure detection and its cause classification, which enables the network administrator to detect the failure in advance and allows taffic rerouting over the affected LP, as to minimize OOS. Finally, in [137], the authors adopted NN and statistical regression to predict failures using the input parameters such as currents drawn, the gain of the amplifiers, power levels and shelf temperatures.

2.3.2.3 Traffic flow classification

The expansion of modern network infrastructure and the latest technology like IoT introduces various types of diverse internet traffic. The network infrastructure should efficiently classify the various generated internet traffic flows before allocating the network resources. It is necessary to make the proper sequence of the internet traffic flows according to their priority, enabling effective resource allocation and ensuring QoS protocols against each traffic flow in order to maintain the SLAs.

In [138], the author observed several classes of packet loss during optical transmission. The authors proposed a hidden Markov model (HMM) together with EM algorithm to classify the packet loss into two main categories, i.e., congestion loss or contention loss. The authors in [139] proposed a NN to categorize traffic flows in data center networks. The NN exploits the IP addresses of the source and destination nodes, port specified for source and destination, protocols operating on the transport layer, size of packets, and flow timings between the initial 40 packets as features.

2.3.2.4 Path computation

While allocating network resources for new arriving traffic requests, a suitable path should be selected to effectively deploy the new traffic request according to the required QoS and also maintain the QoS of already deployed network traffic. Conventionally, routing methods like Dijkstra, Bellman-Ford, Yen algorithms are mainly used for path computation. These traditional path computation algorithms rely on well-defined cost functions such as the shortest path to reach from source to destination, minimizing the overall delay, minimizing power consumption, etc.

The physical and network layer parameters are generally used for path computation in optical networks. The accurate estimation of physical layer parameters is essential because they are mainly managed to select the suitable optical path already reported in section 2.3.1.1. The network layer contributions are summarized in this section. Path computation can generally be considered a multi-layer process, and ML methods can assist this practice at each layer. The authors in reference [140] presented a path computation mechanism for optical burst switched (OBS) networks to reduce the probability of burst loss. The authors tried to handle it like a multi-armed bandit problem (MABP) and suggested a reinforcement learning (Q-learning) method to solve it. The authors adopted the Q-learning method as other techniques are not scaleable for solving complicated problems. A MABP concept derives from a gambling theme; a participant aims to pluck one arm of the slot machine's over several arms to maximize the total reward. Using the same theory, in [140], the path selection between a given source to destination pair is characterized as plucking one of the machines arms, having the premium in terms of decrease in the probability of burst-loss.

Similarly, the authors in [141] proposed a clustering algorithm (fuzzy c-means clustering (FCM)) for QoS aware path computation. The FCM based module is implemented in the SDON controller to improve network performance with the introduced cognitive ability. The FCM manipulates features such as traffic requests, LP lengths, OSNR, modulation formats, BER etc., and provides each LP with the finest available physical layer parameters.

Table 2.2 Summary of Machine learning Applications at Network layer

Machine learning Application	**Machine learning Technique**	**Features/Input**	**Labels/Output**	**Data type**	**References**
Traffic forecasting	ARIMA	Traffic matrix records	Traffic matrix prediction	Synthetic	[119, 120]
	NN	Source-destination traffic matrix	Traffic matrix prediction	Synthetic	[122, 123]
	DL	Intra-datacenter traffic matrix	Intra-datacenter traffic matrix prediction	Real	[124]
	Reinforcement learning	Solutions achieved from GA	Virtual topology update	Synthetic	[128, 129]
NFM	Kriging method	Data of already deployed LP	Locate the failure of all other LPs operating in a network	Real	[130]
	Bayesian network	Real field measured BER and power of LP	Label the leading cause of failure	Real	[131]
	Bayesian network and EM	Missing of GPON/FTTH network data	Predict the missing subset of the dataset	Real	[132, 133]
	BANDO	Previous information about LPs BER	Abnormal BER Detection	Real	[134]
	LUCIDA	Notification from BANDO and taking the preliminary information related to LPs BER and power,	Classifies failure		
	TISSUE	Estimated BER of LP at each traversing node of a network and the theoretical BER	Predict failure	Real	[135]
	FEELING	Power readings around the channel central frequency, levels of power around cut-off site of the channel spectrum	Failure cause localization		
	SVMs, Random Forests, and NNs	Data related to BER	Failure detection and cause classification	Real	[136]

	NN and statistical regression	Current drawn, G of the amplifier, power levels, shelf temperature	Detect failure	Real	[137]
TFO	HMM and EM	Data related to packet loss	Classify the packet loss	Real	[138]
	NN	Source IP plus destination IP, port specifying source and destination, protocols defined for transport layer, sizes of packet, finally intra transmission duration of initial 40 packets	Categorize traffic flows in data center network	Real	[139]
Path computation	Q-learning	Path for each source-destination pair and wavelength selection strategy	Reduce the burst-loss probability	Synthetic	[140]
	FCM	Traffic requests, LP lengths, OSNR, modulation formats, BER	QoS aware path computation	Synthetic	[141]

2.3.3 Machine learning at Photonics or below layer-0

Nowadays, network elements (NE) are progressively exploiting photonic integrated circuits (PICs) to perform complex functions at the photonic level avoiding the bottleneck of optoelectronic conversion. This trend increases the management complexity of such devices. Different ML techniques are effectively used for the proper governance of network devices compared to the traditional analytical models, which are costly in computation at a large scale and can not be executed in real-time.

In [142] the authors proposed an algorithm powered by the NN to calibrate a 2×2 dual-ring assisted mach-zehnder interferometer (DR-MZI) based switching system. The NN is trained to identify the initial state of the DR-MZI. The authors in [142] focused on a 6-nm spectrum segment and 1202 data points, given as an input to NN. The output of NN consists of four variables, i.e., coupler transmission coefficient t, the intrinsic loss factor a, differential phase dp, and the round-trip phase φ_{MZI}. In [143], the authors reported and experimentally demonstrated a full self-learning and reconfigurable photonic signal processor based on an optical neural network (ONN) chip. The ONN chip is encompassed 48 phase shifters and 20 MZIs. The

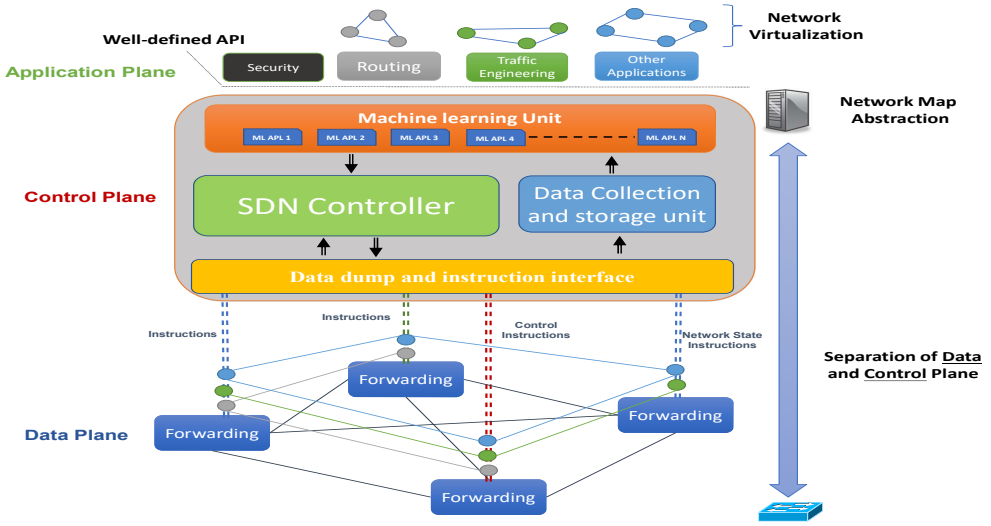


Fig. 2.13 Machine-learning-assisted SDN-enabled optical networks

overall design of ONN is composed of an input layer and an output layer with identity activation functions ($f(z) = z$). The proposed chip performs three main functions by self-learning using the numerical gradient descent algorithm: multi-channel optical switching, optical multiple-input-multiple-output de-scrambling, and tunable optical filtering. In [144], it is proposed to use the deep reinforcement learning (DRL) technique to reconfigure silicon photonic flexible low-latency interconnect optical network switch (Flex-LIONS) according to the traffic characteristics in high-performance computing (HPC) systems. The DRL agent is implemented in the SDN controller. At each system step t, the controller states data s_t containing the inter-top-of-rack connectivity and traffic matrices (obtained through telemetry). The agent uses CNN to get critical features from s_t and subsequently calculate a reconfiguration policy π_t. Furthermore, a novel reinforcement ML-based framework called DeepConf is introduced in [145], for automatically learning and implementing a range of data center networking techniques. This framework is aided by a unique intermediary network state that forms different sets of reinforcement learning models for different tasks of data center networking. The DeepConf unit obtains state s_t (traffic matrix) at training step t and uses this state knowledge the agent makes a policy decision

Table 2.3 Summary of Machine learning Applications at Photonics or below layer-0

Machine learning Application	**Machine learning Technique**	**Features/Input**	**Labels/Output**	**Data type**	**References**
Photonic devices management	NN	Initial state of the DR-MZI selected from 6-nm spectrum segment with 1202 data points	Coupler transmission coefficient t, the intrinsic loss factor a, differential phase dp, and the round-trip phase φ_{MZI}	Real	[142]
	NN	Four independent 10 Gbit/s non-return-to-zero (NRZ) signals	Multi-channel optical switching, optical multiple-input-multiple-output de-scrambling, and tunable optical filtering	Real	[143]
	DRL	Controller states data s_t (inter-top-of-rack connectivity and traffic matrices)	Reconfiguration policy π_t	Synthetic	[144]
	DeepConf (reinforcement learning)	Input traffic matrices	Reconfiguration policy	Synthetic	[145]

After the general argument in this part, it is pretty clear that ML algorithms can be effectively utilized for different practical interests. Hence, this leads a road map toward the development of a cognitive control plane for the next era of modern optical networks having various ML API running, which assists different functionality at different network levels shown in Fig. 2.13.

Chapter 3

Machine Learning for Optical Networks

Introduction

Recently, the remarkable increase in the global IP traffic, compelled by the introduction of 5G technology, the IoT, and cloud services, has marked-up high demands and new requirements for the capacity enhancement and reliability of optical networks [146]. To serve the rapidly increasing number of internet service users, the technologies of optical networks are continuously evolving. To cater to this dramatic upsurging number of internet service users, the key operator demands the full utilization of network resources of existing infrastructure to maximize returns on CAPEX investments. To achieve this, the data transport layer needs to be optimized to reach the maximum available capacity. The center key factor for optimal exploitation of data transport is maximizing the capacity of WDM transmission techniques along with network disaggregation. These attributes evolved to technologies: EON and SDN paradigm in an optical network. The EONs provide flexibility to the network controller to scale up or down resources according to the traffic requests in order to utilize the available spectrum efficiently [3, 33]. In addition to this, the SDN implementation enables the management of each NE within a virtualized environment, so permitting a disaggregated approach to the network exploitation, enabling openness and virtual network slicing. These features pave a roadway for a partial or full

disaggregation of optical networks: a disaggregated network includes subsystems that are controlled independently by relying on common data structures and API.

The foundation step towards flexible and disaggregated networks is the abstraction of the WDM optical transport as a topology graph weighted by the GSNR degradation on transparent LPs, introduced by each crossed NE and subsystem, mainly by OLS including fibers and amplifiers [147, 148]. Each OLS is handled by an OLS controller running in the control plane [149], which sets the amplifier working point and consequently determines the GSNR degradation. The more accurate the nominal working point, the better the capability to rely on the overall LP GSNR. So, a lower system margin is requested in LP deployment and, consequently, larger traffic can be handled, enabling better exploitation of the installed equipment. In addition to this, a reliable softwarization of optical transport also helps in the automatic recovery of network failures by reducing downtime.

In this chapter 3, a framework for ML-assisted optical networks is discussed to estimate the QoT of LP. The chapter begins with the physical layer abstraction of the optical network in Section 3.1. After that, the QoT metric in terms of GSNR is discussed in Section 3.2. Later, the different approaches used for estimating the QoT are discussed in Section 3.3. The Section 3.4 reports the network simulation model along with dataset generation and its analysis. In Section 3.5, the proposed ML models for QoT-E and the optimization of hyperparameters are discussed. After this, the performance analysis of all the proposed models is given in the Section 3.6. The analysis demonstrated in Section 3.6 only consider the ML models to predict the QoT-E of the LP. After this analysis, in Section 3.7, a more realistic scenario is reported in which the ML model is synergically used with the network QoT estimator engine in order to assist the QoT estimator engine in estimating the QoT of LP.

3.1 Physical layer abstraction of transparent open optical network

A transparent optical transport network is a structure of connected ROADM nodes, where traffic requests are added/dropped or routed, as shown in Fig. 3.1. Topology links are bidirectional OLSs made of fiber pairs and in-line amplifiers – EDFA [150]. As links are virtually symmetrical, an unidirectional approach is

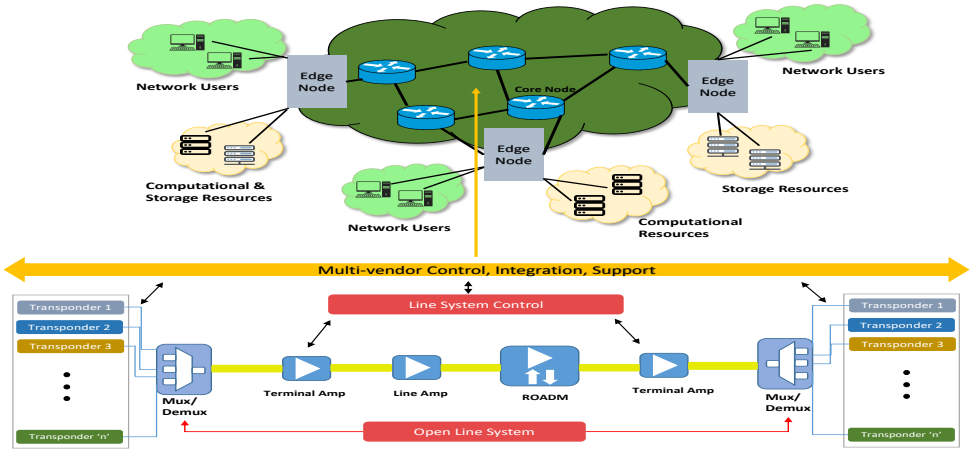


Fig. 3.1 Schematic description of an optical network as a topology of ROADM nodes connected by OLSs. The inset shows a general setup for an OLS that, in this case, is supposed to be open.

considered in this analysis. Following the disaggregated approach, each OLS is independently and autonomously controlled by an OLS controller that sets the operational point of in-line amplifiers to minimize QoT degradation. [151, 149]. The OLS controller's settings are based on the available parameters describing the physical layer, among which the most important are the fiber connector losses and the ripples of in-line amplifiers Gs and NFs. In particular, these ripples vary with the spectral load. So, OLS controllers may guarantee operation at the nominal working point with a certain degree of uncertainty on the actual GSNR degradation.

On the optical infrastructure, LPs are deployed as transparent optical circuits on the WDM flexible grid [152] connecting transceivers; hence, they support dual-polarization multiple level modulation formats. In this framework, a higher operative GSNR enables a larger deployable modulation cardinality, which provides a more significant deployable rate. Thus, the key operation in LP deployment is to select the feasible modulation format, so maximizing the rate according to the available LP QoT. Such an operation performed by the network controller requires an accurate QoT-E, as uncertainties on the actual LP QoT require system margins and a consequent reduction in the deployed traffic.

Generally, LPs suffer from several propagation impairments: ASE noise from in-line amplifiers, linear and nonlinear fiber propagation effects and ROADMs filtering effects. Thanks to DSP techniques in coherent transceivers, fiber propagation linear

effects can be managed together with additive Gaussian disturbances (NLI) and phase noise, both nonlinear and generated by self- and cross-channel nonlinear crosstalk.

In general, the DSP module at the receiver effectively compensates for the phase noise using the carrier phase estimator algorithm. This is necessary for QoT-E in transceivers scenarios with large-constellation modulation formats, designed for short-reach [153].

Finally, the ROADMs filtering effects and crosstalk also introduce some level of degradation to the QoT, generally considered as an extra loss and additional Gaussian noise source. ROADM impairments are out of the scope of this work.

3.2 GSNR as QoT-E Metric

Nowadays, the core networks are typically operated by transceivers using QPSK, 8-QAM and 16-QAM constellations, or hybrid formats [154]. In such implementations transparent LPs can be effectively modeled as additive white Gaussian noise (AWGN) channels, affected by ASE noise and NLI Gaussian disturbances. Typically, the QoT over an AWGN channel is summarized by the signal-to-noise ratio (SNR), that for LPs is typically identified as GSNR:

$$\mathrm{GSNR} = \frac{P_{\mathrm{Rx}}}{P_{\mathrm{ASE}} + P_{\mathrm{NLI}}} = \left(\mathrm{OSNR}^{-1} + \mathrm{SNR}_{\mathrm{NL}}^{-1}\right)^{-1}, \tag{3.1}$$

where $\mathrm{OSNR} = P_{\mathrm{Rx}}/P_{\mathrm{ASE}}$ is the optical signal to noise ratio detectable by optical channel monitors (OCM)s, $\mathrm{SNR}_{\mathrm{NL}} = P_{\mathrm{Rx}}/P_{\mathrm{NLI}}$ is the nonlinear SNR due to NLI only and so only observable on the DSP recovered constellation. P_{Rx} is the power of the channel at the receiver, P_{ASE} is the power of the accumulated ASE noise and P_{NLI} is the power of the accumulated NLI.

Following a disaggregated approach, the NLI introduced by each fiber span can be separated into two main contributions: the self-channel NLI (SC-NLI) and the cross-channel NLI (XC-NLI) depending on the input spectrum [155]. Several mathematical models have been proposed for the NLI evaluation with a disaggregated approach, with different accuracy levels, e.g., [156, 157]. For an accurate spectrally-resolved evaluation, NLI calculation must consider the simultaneous effect of stimulated raman scattering (SRS) [158]. So, the NLI per span can be evaluated by a proper

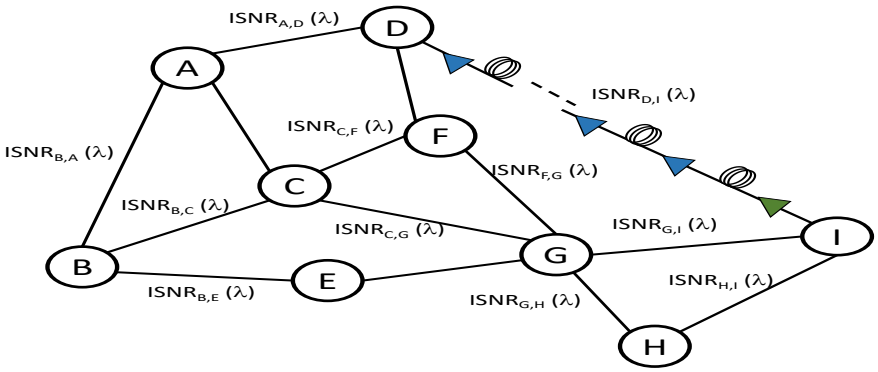


Fig. 3.2 Abstraction of Optical Network

algorithm following a spectral disaggregated approach superimposing the effects of each channel. This is the approach exploited in the present work by using the Gaussian noise simulation available in python (GNPy) library. For the ASE noise accumulation, the considered G and NF are for a typical amplifier set. So, a data structure effectively abstract WDM optical transport network as a topology weighted graph G(V,E), where graph-vertices (V) are switches and ROADMs, while graph-edges (E) are OLSs. Weights on edges are the GSNR characterizing each OLS as measured by the OLS controller. Fig. 3.2 shows an example, where inverse-SNR (ISNR)=1/GSNR. Over such a data structure, LP GSNR can be evaluated by *navigating* the route and accumulating the metrics of interest for the wavelengths under test:

$$GSNR = \frac{1}{\sum_i^{OLS} \frac{1}{GSNR_i}} \,. \tag{3.2}$$

The accuracy in GSNR computation depends on the mathematical models for fiber propagation, amplifiers, and the accuracy in the knowledge of physical layer parameters. The perturbation from nominal values used for OLS controlling and for QoT-E yields an uncertainty on the LP GSNR, which requires system margins and reduces the deployable traffic, with respect to the nominal amount. The main uncertainties are the fiber connector losses defining the fiber input power, and consequently, the NLI and the in-line amplifiers G and NF determining the ASE noise, with a dependence on the spectral load.

In this chapter, the simulation performed for comparing different ML models only considers amplifiers G ripple and NF as a primary source of uncertainties, while for transfer learning agent fiber connector losses are also considered along with amplifiers G ripple and NF.

3.3 Approaches for QoT Estimation

This section lists different possible approaches for obtaining knowledge of the OLSs characteristics, with each method allowing a different reduction in the GSNR uncertainty.

In the first approach, the data available from the system and network components such as static characterization of devices (e.g., amplifier G and NF in the frequency domain, connector loss, etc.) is used to implement an accurate QoT-E in vendor-specific systems. Regarding this particular approach, a considerable number of analytical models are available for calculating the GSNR by using the data and characterizing the OLS components. However, this static data-based approach may not be accurate as the components experience gradual degradation due to the aging factor, leading to progressively unreliable QoT-E after a certain period.

A second approach is based on the telemetry data considering only the present network status. Deducing an OLS agnostic functioning in an open environment, the OLS controller mainly depends upon telemetry data retrieved from the OCM and the EDFAs. In this particular approach, it is reasonable to use the telemetry of the network's current state to estimate an accurate QoT. In contrast to the previous approach, there is no reliance on the static characterization of devices parameters and thus the uncertainty in the QoT-E accuracy due to the device aging factor is removed. Nevertheless, the issue with this particular method is that the GSNR response, mainly its OSNR component, is highly dependent upon the spectral load configuration, which introduces a considerable uncertainty in the QoT margin [72].

The third approach considers a dataset that collects the QoT responses against random spectral loads of the *in-service* network. This data is generated during the operative phase of the *in-service* network by measuring the OLS response in terms of GSNR for various spectral load configurations. This particular case provides an ideal playground to apply ML. A ML method using a training dataset composed of spectral load realizations of the *in-service* network yields an accurate QoT-E for each newly generated spectral load realization for *agnostic/unused* network scenarios. In contrast to the second approach, where only telemetry data is considered, this approach utilizes the QoT-E based on the GSNR response to specific spectral load configurations, decreasing the uncertainty in GSNR predictions of *agnostic/unused* network. Additionally, this method does not need any knowledge of the OLSs

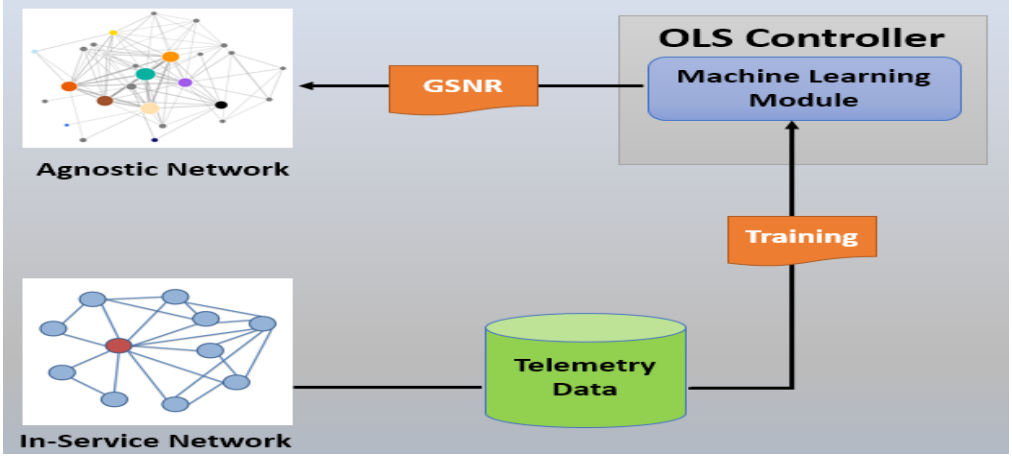


Fig. 3.3 Model Orchestration

physical parameters in opposition to the first approach. Thus, this method provides an ideal platform to apply ML methods in an *agnostic/unused* network scenario.

In this thesis, the main focus is on the third approach, considering a ML model trained on the GSNR response to specific spectral load configurations of the already deployed *in-service* network, and considering its realization to predict the QoT of an *agnostic/unused* network, as shown in Fig. 3.3.

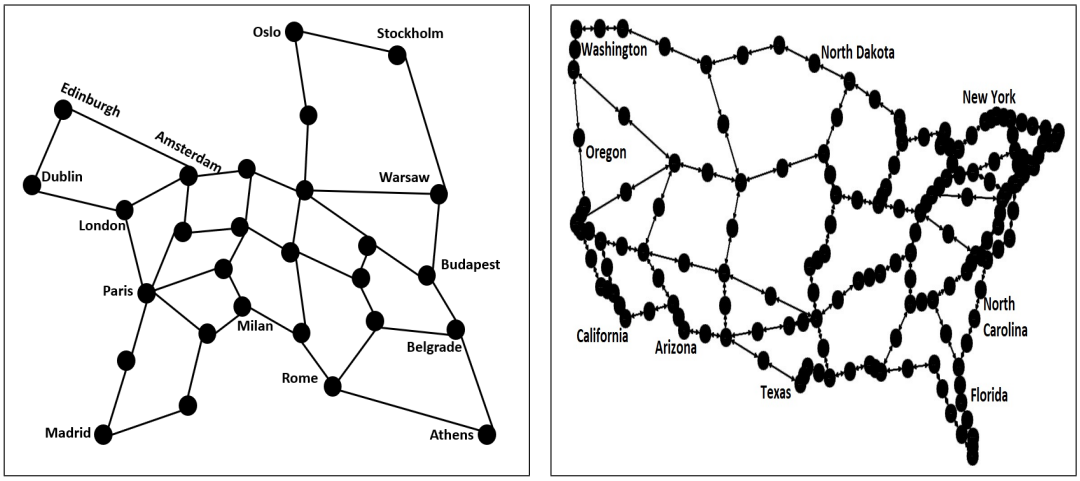
Table 3.1 Simulation Parameters

Parameters	
Launch Power/ Channel	0 dBm
Dispersion (D)	16.0 $ps/nm/km$
Attenuation coefficient (α)	0.3 dB/km
Channel Spacing	50 GHz
Span Length	80 km
WDM Comb (C-Band)	76
Baud Rate	32 $Gbaud$
Amplifier Noise Figure	[6 - 11] dB [159]
Nominal Amplifier Noise Figure	8.75 dB
Amplifier Gain Ripple	Variation of 1 dB
Nominal Amplifier Gain Ripple	Flat
Fiber Type	Standard SMF
Insertion losses	Exponential Distribution ($\lambda = 4$)[160]
Nominal Insertion losses	0.3 dB [161]

3.4 Simulation model and synthetic data generation

This section describes the simulation performed to model a particular network and its transmission components. Along with this modeling, the mimicked datasets detail are also reported, which are generated synthetically for two different networks; *in-service* and *agnostic/unused* network by using the GNPy library.

A typical SDN empowered backbone optical network is considered, in which edges are modeled by OLSs, while nodes are defined as ROADM sites. The considered OLS is controlled targeting the maximum GSNR [162] according to the local-optimization global-optimization (LOGO) strategy [149]. The considered OLS controllers and QoT-E might rely on the nominal values for NE parameters. Although both datasets are synthetically generated, the *in-service* network dataset is obtained by reading data from transceivers and monitors for each deployed LP, specifically the channel power and OSNR from monitors, GSNR from transceivers and the number of crossed spans from the controller. Therefore, in this investigated hypothetical scenario, the *in-service* network dataset contains the exact channel powers, OSNRs and GSNRs values, without any uncertainty. On the other hand, the *agnostic/unused* network dataset includes inaccuracies due to network parameter uncertainties. To mimic reality, a synthetic dataset is generated considering random connector losses and in-line amplifiers NF and G ripples. In particular, at first the connector losses are randomly set, while a Monte-Carlo analysis is performed on different OLS spectral load and consequent ripples [163], in order to collect the dataset. Insertion losses are determined by an exponential distribution with $\lambda = 4$ as described in the study [160]. To generate synthetic data, the GNPy library [164, 159] is used. The GSNR computation of the GNPy library is spectrally resolved and is based on the generalized Gaussian noise (GGN) model for NLI. The GGN model always considers worst effect for the NLI by supposing Gaussian-modulated interfering channels [158, 159]. Because of the computational effort needed to generate a proper dataset, the considered OLSs carry only 76 channels over the standard 50 GHz grid on the C-band, having total bandwidth close to 4 THz. A substantial difference in results is not expected when considering standard 96 channels on the entire C-band. The transceivers at 32 GBaud, shaped with a root-raised-cosine filter, are considered. The OLS launch power is defined by the booster at the ROADM output, as to set 0 dBm per channel, while in-line amplifiers are set at transparency, i.e., the nominal G completely recovers fiber losses. All OLSs are supposed to operate on standard



(a) EU network topology (b) USA network topology

Fig. 3.4 Network Topologies

single-mode fiber (ITU-T G.652) with a span length of 80 km. These hypotheses are used for both the *in-service* and *agnostic/unused* network, as both considered networks exploit the same optical transport technologies. Network parameters are summarized in Tab. 3.1.

Table 3.2 Train set network paths

EU: Training					
Source	**Destination**	**Spans**	**Source**	**Destination**	**Spans**
Vienna	Warsaw	7	Brussels	Bucharest	30
Amsterdam	Berlin	8	Frankfurt	Istanbul	34

The generated dataset is obtained by perturbating the above frame of reference by varying the parameters of EDFA: NFs, amplifier G ripples, and fiber connector losses. The selection of NF is made by a uniform distribution varying between 6 dB to 11 dB, while the amplifier G ripples are varied uniformly within a 1 dB interval.

(Note that such a wide range of NF is typical in commercial devices at spectral loading far from full load). The considered spectral loads in dataset generation is a subset of the entire 2^{76} possible loads, where 76 is the total number of WDM channels. In the considered subset of spectral loads, each source-to-destination (s $\rightarrow$ d), pair has 1024 realizations of random loads ranging from 34% to 100% of total bandwidth occupation. Given the (s $\rightarrow$ d), the add/drop variation of the spectral load has been reproduced by randomly selecting amplifier ripples. Nevertheless, it always assumes a full spectral load when the NLI impairment is calculated, as it represents

Table 3.3 Test set network paths

USA : Testing					
Source	**Destination**	**Spans**	**Source**	**Destination**	**Spans**
Milwaukee	Minneapolis	6	Denver	Detroit	29
Louisville	Memphis	7	Kansas City	Las-Vegas	30
Boston	Buffalo	8	Las-Vegas	Little Rock	30
Charleston	Charlotte	8	Albuquerque	Atlanta	33
Chicago	Cincinnati	12	Birmingham	Bismarck	33
Greensboro	Hartford	14	Austin	Baltimore	34
Atlanta	Austin	20	Miami	Milwaukee	35
Charlotte	Chicago	20	Abilene	Albany	36
Columbus	Dallas	20	Bismarck	Boston	36
Dallas	Denver	20	Detroit	El Paso	38
El Paso	Fresno	21	Hartford	Houston1	38
Minneapolis	Nashville	21	Baton Rouge	Billings	41
Buffalo	Charleston	22	Billings	Birmingham	41
Houston	Jacksonville	23	Albany	Albuquerque	45
Memphis	Miami	24	Los Angeles	Louisville	46
Baltimore	Baton Rouge	25	Fresno	Greensboro	54
Little Rock	Long Island	25	Long Island	Los Angeles	61
Jacksonville	Kansas City	26			

the worst-case scenario. The first dataset is generated for the EU network topology which consists of 4096 data realizations for four (s –> d) pairs (1024 combinations for each (s $\rightarrow$ d) pair) and is used as an *in-service* network as shown in Tab. 3.2, while for the *unused* network, the dataset is generated for the USA network topology that includes thirty five (s $\rightarrow$ d) pairs having 35,840 data realizations as shown in Tab. 3.3. The considered networks are characterized by distinct topologies shown in Fig. 3.4a and Fig. 3.4b, adopting the same fiber type and transmission equipment. Still, they are different in terms of random amplifiers parameters, i.e., ripples on NF and G and fiber insertion losses. The basic parameters for both network topologies are shown in Tab. 1. The distribution of *GSNR* for a single particular path *Louisville* $\rightarrow$ *Memphis* for all the considered realizations of varying spectral load of WDM channels is depicted in Fig. 3.5, which includes the resulting GSNR values obtained varying spectral loads, WDM channels, G ripples, NFs and insertion losses. A single path is analyzed as the other paths do not provide a substantial difference in the statistical characteristics of GSNR.

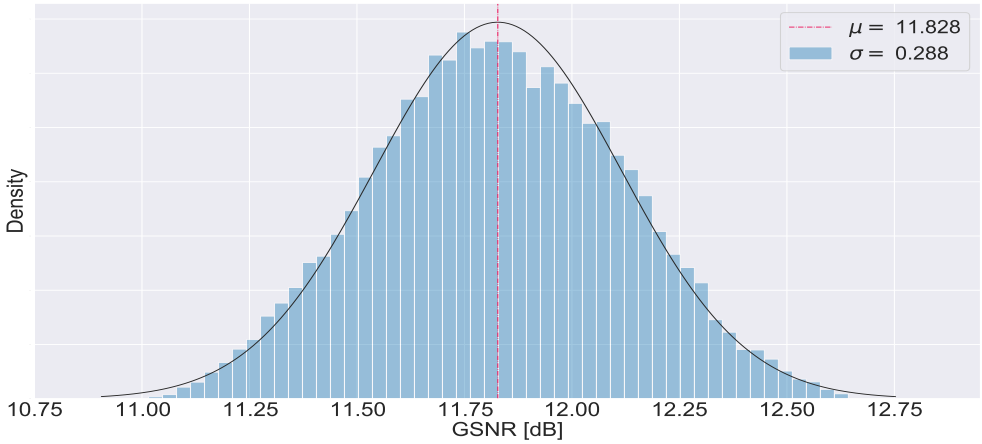


Fig. 3.5 GSNR statistic of all channels of *Louisville* → *Memphis* path

Table 3.4 Networks Topology Details

Topology Details		
Parameters	**EU: Training [165]**	**USA : Testing[166]**
Number of Nodes	28	100
Number of Links	41	171
Average path distance (km)	2014.06	2541.75
Maximum path distance (km)	3051.10	5481.07
Minimum path distance (km)	669.30	568.33
Average number of spans per Link	19.75	27.49

3.4.1 Analysis of GSNR Dataset

After retrieving the dataset, the statistical analysis of the GSNR against different spectral load configurations for particular (s → d) is made to estimate the uncertainty in GSNR calculations. A single path *Louisville* → *Memphis* is selected in the test dataset for the statistical analysis of the GSNR. Only one path is selected for analysis as the other paths do not provide any substantial difference in the statistical GSNR characteristics.

Firstly, the distribution of *GSNR* for this particular path is depicted in Fig. 3.5. In this regard, the channel under test (channel number 10) is selected for all the test realization in which it is always *ON*. Along with this analysis, a few primary considerations arise by computing the average of the GSNRs for every channel over all the test realizations of this specific path, presented in Fig. 3.6. The average values of GSNR allow to model the OLS components, with an average distribution between 11.72 dB, and 12.28 dB, exhibiting standard deviations from 0.20 dB to 0.29 dB respectively.

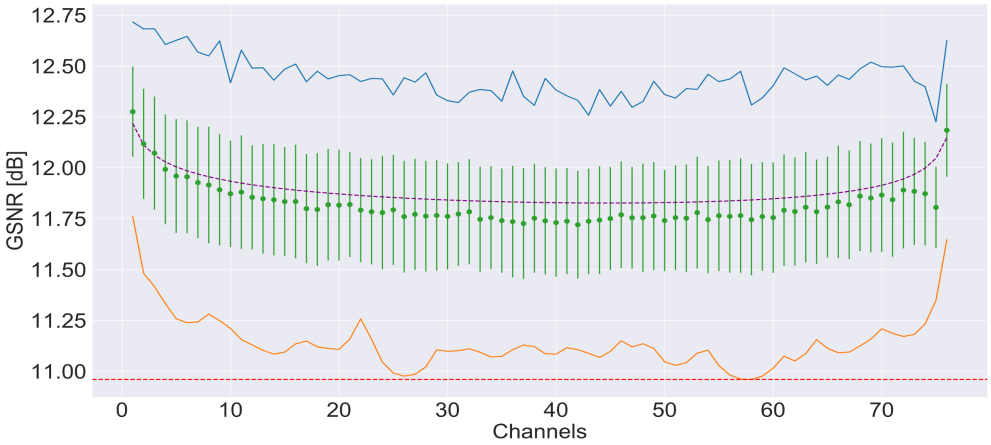


Fig. 3.6 Overall, GSNR measurements for a single path Louisville → Memphis of an *agnostic/unused* network in the frequency domain. The green dots are the mean values over the entire sample for each channel; the error bars are equal to the standard deviations. The purple line shows the nominal values for this path in the frequency domain. In blue and orange, the maximum and the minimum for each channel are outlined, respectively. The dashed red line indicates the overall GSNR minimum of 10.81 dB.

Considering the current scenario, if nothing is known about the GSNR dependency upon frequency, the same GSNR threshold must be enforced for all channels with a magnitude lower than a global expected minimum. In this case, the GSNR value is fixed to the constant threshold of 10.81 dB creating an average margin of up to 1.48 dB over a considered set of realizations for this particular path. Moving towards the following scheme, the availability of stored data that characterize the frequency-resolved GSNR response must be considered: the margin can be truncated by fixing a minimum value for each channel, which must lie below the respective minimum measurement (continuous orange line in Fig. 3.6). However, this solution is sub-optimal; it is the most acceptable reachable result, conservative and agnostic with respect to the definite spectral load configuration against this specific path.

This definitive solution yields a finite improvement, compared to the first case having a value of 1.48 dB, as the average margin would be reduced to 1.30 dB. Still, this second approach is strikingly dependent upon the sample space of GSNR realization: acquiring a minimum reliable value for each channel requires a sufficiently large number of instances. In the above context, both the considered scenarios are not feasible given *agnostic/unused* network, where the statistics distribution of the GSNR with respect to different spectral load configurations is not available.

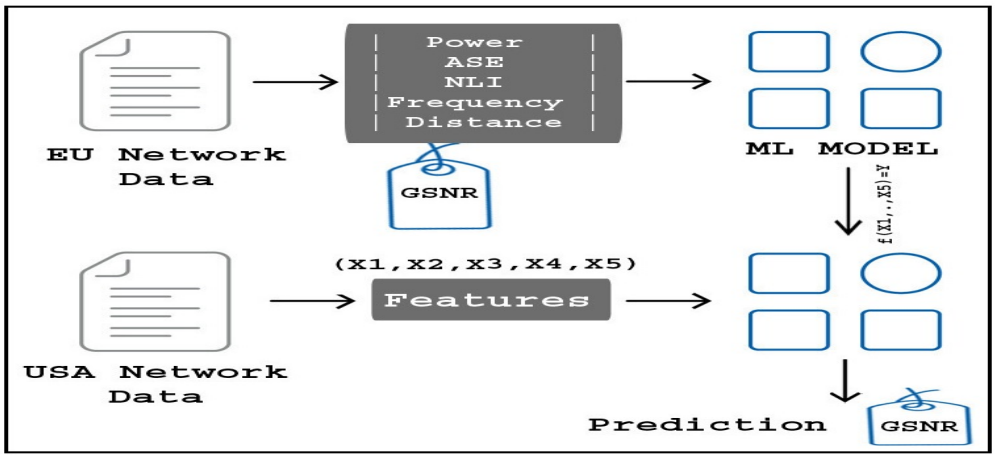


Fig. 3.7 Machine Learning Module

In contrast to this, a ML approach appears to be a promising candidate not only to predict the GSNR against any particular path of a network but can also lead to a decrease in the uncertainty in GSNR predictions, even if the dimension of the sample of *in-service* network is limited.

3.5 Machine learning techniques for QoT-E

The proposed models cognate the features and labels of the candidate LP of the *in-service* network using ML. The manipulated parameters used to define the features (system input) for the ML models include received signal power, NLI, ASE, channel frequency, and distance between source to the destination nodes (integral number of fiber spans traversed). In contrast, the exploited label (system output) is manipulated by the *GSNR* parameter of the candidate LP depicted in Fig. 3.7. The total number of input features for proposed ML models consist of *305* entries, as there are 76 entries against each manipulated parameter *(*$1+4\times 76=305$*)*. The considered models use ML's functionality, which is a powerful tool to find the relationship between the provided features and the desired label [167]. Typically almost all ML-based models perform better when they are trained on standardized datasets [168]. Generally, these sets have zero mean and unit variance.

$$Z=\frac{X-\mu}{\sigma} \tag{3.3}$$

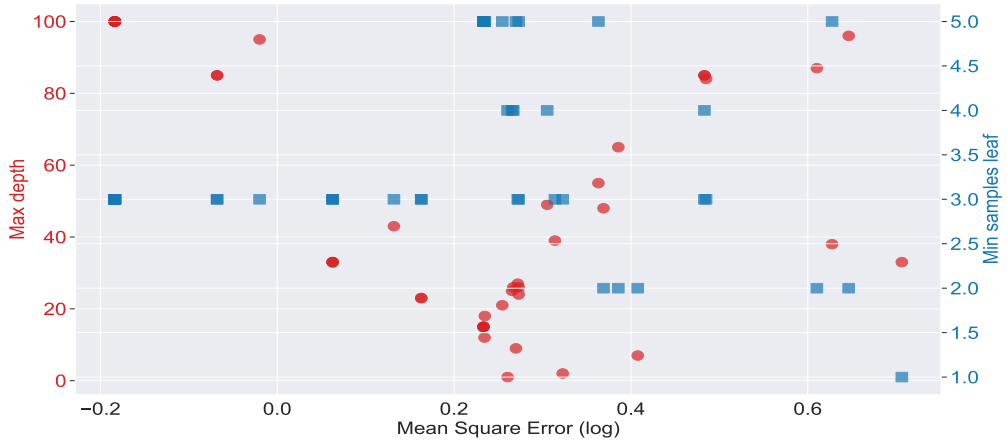
$$\text{MSE}=\frac{\sum_{i=0}^{n}\left(\text{GSNR}_i^{\text{p}}-\text{GSNR}_i^{\text{a}}\right)^2}{n}, \tag{3.4}$$

In the present work, standardization of the dataset is done using z-score normalization, expressed in Equation 3.3, where μ and σ are the mean and standard deviation of each particular feature vector of the dataset. Along with this, model evaluation is done by MSE as a loss function expressed in Equation 3.4, where GSNR_i^{a} and GSNR_i^{p} are the actual and predicted values of any candidate channel for the ith spectral load, respectively, and n is the total number of realizations in the test dataset.

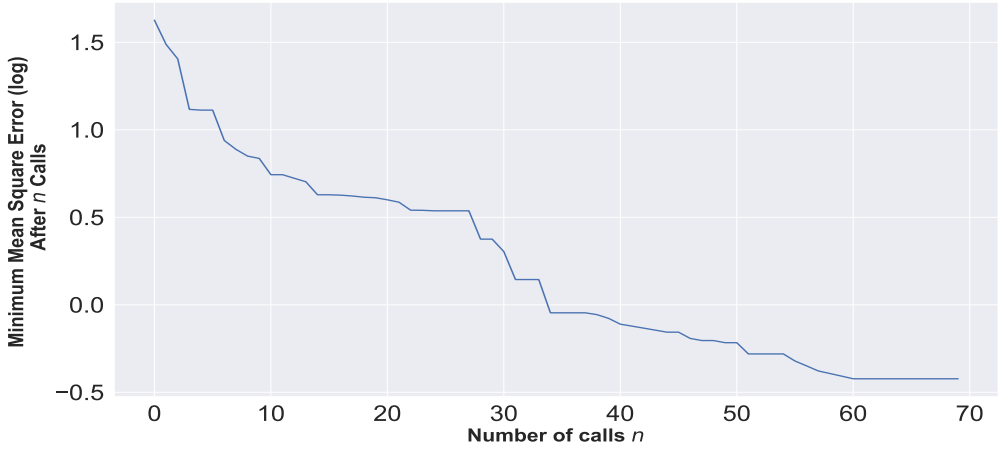
The underlying paradigm of ML allows the inference of functional network characteristics that cannot be easily or directly measured. Generally, the cognition ability of ML models is provided by a series of intelligent algorithms that learn the inherent information based on the training data. The intrinsic information is then abstracted into decision models that guide the testing phase. These trained decision models offer operational advantages by allowing the network to *draw conclusions* and react autonomously.

In the present work, six ML models are proposed for QoT-E. Each proposed model consists of three basic modules: pre-processing, training, and testing. The pre-processing module standardizes the dataset before applying it to the training module. The training module uses the standardized dataset of the EU network, which is considered as *in-service* network reported in Table 3.2 for the training of the proposed models. After training, the testing module explicitly starts testing on the USA network dataset subset, considered as an *agnostic/unused* network, as reported in Table 3.3. The EU network is selected for training as its size is smaller than the USA network, which helps in the extensive testing of the models as the USA network provides more variety in terms of network parameters, especially fiber span length.

The proposed ML-based regression models are developed by using high-level python API of two open-source ML libraries, *scikit-learn*© (SKL) [168] and *TensorFlow*© (TF) [169], which provide a variety of learning algorithms as well as appropriate functions to refine the dataset before using as an input of the ML model. Generally, SKL is a general-purpose or a traditional ML library, while TF is considered a DL library. Using python API of *SKL's*, three possible ML models can be deployed: decision tree regressor, random forest regressor, and multi-layer perceptron regressor. Using *TF's* the alternatives are: boosted tree regressor, deep neural network and wide deep neural network. In this thesis, only regression models are proposed as the considered problem output variable (label) is only numerical (continuous) rather than categorical (discrete).



(a) Max depth and Min sample leaf optimization

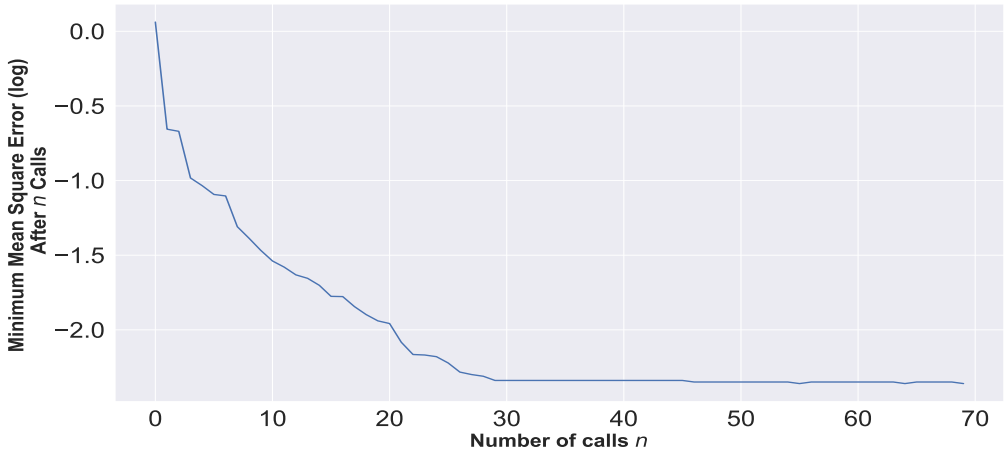


(b) MSE convergence plot for tree model

Fig. 3.8 Hyperparameters optimization of tree-based models using Skopt

Based on the hyperparameters, the proposed models have two primary classes; tree-based models and neural network-based models. The hyperparameters for both types of models are optimized before loading them to a specific model. There are various techniques for hyperparameter optimization of the ML models such as manual search, random search, grid search, automated hyperparameter tuning, etc. In the proposed thesis, the sample space of hyperparameters is first explored using manual search. After that, the manual search optimization is further verified by using the automated hyperparameter tuning scheme of Scikit-optimize (Skopt) open-source library. Skopt utilizes a sequential model-based optimization algorithm to search for optimal configuration of hyperparameters for particular problems in less time [170].

(a) Neurons per *hidden-layer* and number of *hidden-layers* optimization



(b) MSE convergence plot for neural network model

Fig. 3.9 Hyperparameters optimization of neural network-based models using Skopt

The hyperparameters optimization is done in two fashions; initially, it is performed for tree-based models (see Fig. 3.8), and after that, optimization is done for neural network-based models (see Fig. 3.9). In Fig. 3.8a, the parameters like max depth and minimum sample leaf are optimized with respect to the loss function (MSE). Observing Fig. 3.8a, the optimized value for minimum sample leaf is 3, while max depth is equal to 100. The same optimized parameters are considered for all the proposed tree-based models. After that, to further verify the obtained results in Fig. 3.8a, the MSE convergence plot is also shown with respect to the number of calls in Fig. 3.8b. A similar analysis is performed for neural network-based models for the two important parameters, i.e., the number of neurons per *hidden-layer* and the total number of required *hidden-layers* for the considered problem. The results

in Fig. 3.9a report the optimized value for the number of neurons per *hidden-layer* which is equal to 20, while the total number of required *hidden-layers* is equal to 3. The optimized parameters obtained in Fig. 3.9a are considered for all the proposed neural network-based models. Furthermore, to validate the acquired findings in Fig. 3.9a, the MSE convergence plot is shown with respect to the number of calls in Fig. 3.9b. The default and optimized hyperparameters of the proposed ML models are reported in Table 3.5.

Table 3.5 Machine learning Models Detail

Machine learning Model	**API**	**Parameter**	**Value**
Decision Tree Regressor	*scikit-learn*©	*Min samples leaf*	3
		Max depth	100
Random Forest Regressor	*scikit-learn*©	Method	$'Bagging'$
		Min samples leaf	3
		Max depth	100
Multi-layer Perceptron Regressor	*scikit-learn*©	*Hidden layers*	3
		Neurons/hidden layers	20
		Stochastic Gradient Descent Algorithm	*SGD*
		Activation function	$'ReLU'$
		L_1 regularization	0.001
		Learning rate	0.01
		Training steps	1000
Boosted Tree Regressor	*TensorFlow*©	Method	*'Gradient Boosting'*
		Min samples leaf	3
		Max depth	100
		Learning rate	0.01
		L_1 regularization	0.001
Deep Neural Network	*TensorFlow*©	*Hidden layers*	3
		Neurons/hidden layers	20
		Keras optimizer	$'ADAGRAD'$
		Activation function	$'ReLU'$
		L_1 regularization	0.001
		Learning rate	0.01
		Training steps	1000
Wide Deep Neural Network	*TensorFlow*©	*Hidden layers*	3
		Neurons/hidden layers	20
		Keras optimizer	$'ADAGRAD+FTRL'$
		Activation function	$'ReLU+Sigmoid'$
		L_1 regularization	0.001
		Learning rate	0.01
		Training steps	1000

3.5.1 Decision Tree Regressor

To estimate QoT, the decision tree regressor (DTR) model is proposed. The DTR has two main basic tuning parameters; *min_samples_leaf* and *max_depth*. To get the optimum values of these parameters, the number of *min_samples_leaf* = 3 and

max_depth = 100 are selected in order to achieve the best trade-off between precision and computational time in this particular simulation scenario.

3.5.2 Random Forest Regressor

To estimate QoT, the next proposed tree model is random forest regressor (RFR). The (RFR) is a type of ML technique that builds an *ensemble* of simple regression trees using a *bagging* technique. Similar to DTR parameter tuning, the RFR is also configured with *min_samples_leaf* = 3 and *max_depth* = 100 in this simulation scenario.

3.5.3 Multi-layer Perceptron Regressor

The multi-layer perceptron regressor (MLPR) is one of the most commonly used AI networks. Generally, MLPR is not a single perceptron with multiple layers but multiple artificial neurons (perceptrons) with multiple layers. Typically, MLPR has three or more layers of perceptrons. The layers of the MLPR form a directed, acyclic graph where each layer of MLPR is fully connected to the subsequent layer [171]. The proposed MLPR is configured by several parametric values such as *training steps* = 1000, loaded with back-propagation (BP) algorithm along with default *stochastic gradient descent (SGD)*) algorithm optimizer having default *learning rate* = 0.01 and L_1 regularization = 0.001 [172]. The basic function of L_1 regularization is to prevent MLPR from over-fitting. In addition to this, several non-linear activation functions such as *Relu, Tanh, sigmoid* are tested during model building. After testing all of these non-linear activation functions, *Relu* is selected to improve MLPR capabilities, as it outperforms others in terms of prediction and computational load [65]. Finally, the most important parameter is the *hidden-layer*, as MLPR is tuned on several numbers of *hidden-layer* and neurons to achieve the best trade-off between precision and computational time. These two parameters are linked to the complexity of the MLPR, which is tied to the intrinsic complexity of the problem. Although the increase in the number of layers and neurons improves the accuracy of the MLPR up to a certain extent, a further increase in these values has an adverse diminishing return effect that causes over-fitting and increases in the computational time. In this trade-off, MLPR for QoT-E uses *3 hidden-layers*, containing *20 neurons* each.

3.5.4 Boosted Tree Regressor

The boosted tree regressor (BTR) is also a type of ML algorithm that creates an *ensemble* of regression trees using the *gradient-boosting* technique. Similar to other proposed tree regressors parameter, the BTR is also configured with *min_samples_leaf* = 3 and *max_depth* = 100. Further more, BTR is also configured by several other parameters such as default *learning rate* = 0.01 along with L_1 regularization = 0.001 for absolute weights of the tree leafs for the current simulation scenario.

3.5.5 Deep Neural Network Regressor

The DNN is a type of artificial neural network (ANN) with multiple layers between the input and output layers. Each particular layer of DNN consists of multiple neurons, the computational and learning unit of NN. For the QoT-E, the proposed DNN is configured by several parametric values such as *training steps* = 1000, loaded with default *adaptive gradient algorithm (ADAGRAD)* keras optimizer with default *learning rate* = 0.01 and L_1 regularization = 0.001 [173]. The basic function of L1 regularization is to prevent DNN from over-fitting. In addition to this, during model building, *Relu* non-linear activation function is selected, which allows translation of the given input features into the prediction label of our interest with less complexity [65]. Finally the most important parameter remains the number of *hidden-layers*, as similarly to the previously discussed methods, DNN is tuned on several number of *hidden-layers* and neurons in order to achieve the best trade-off between accuracy (in terms of low MSE) and computation cost. In this trade-off, DNN for QoT-E uses *3 hidden-layers*, containing *20 nodes* each.

3.5.6 Wide Deep Neural Network

The wide deep neural network (W-DNN) is a type of DNN architecture that combines the strength of *memorization* with *generalization* [66]. For the QoT-E, the proposed W-DNN is configured with 1000 *training steps*, loaded with default *follow the regularized leader* (FTRL) keras optimizer with default *learning rate* = 0.01 and L_1 regularization = 0.001 against wide *LR* model [174], while generalize *DNN* is loaded with default *ADAGRAD* keras optimizer with default *learning rate* = 0.01 and L_1 regularization = 0.001 [173]. The *Relu* nonlinear activation functions is selected

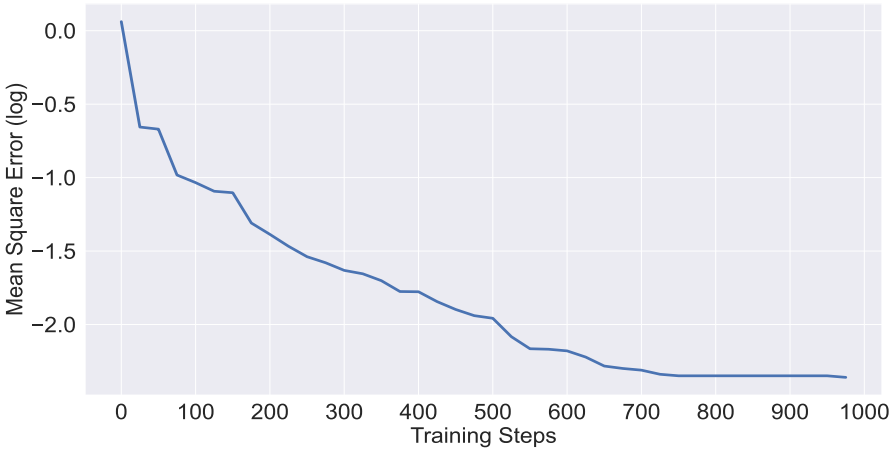


Fig. 3.10 W-DNN loss function over the training Steps

to empowered the deep part, as it outperform others non-linear activation function in terms of prediction [65]. The output layer is fed with *sigmoid* function, as the *sigmoid* produces activation values in a particular range so that the output layer will always be activated. During the training of W-DNN, the wide and deep parts are jointly trained at the same time. The loss function over training steps for W-DNN is observed in Fig. 3.10. The synergic training of both wide and deep architectures optimizes all parameters and the related weights of their sum, taken into account during the training time.

3.6 Performance analysis of machine learning models

The numerical assessment of various ML models developed by using python APIs of *SKL* and *TF* libraries have been performed by considering 4 different paths of EU network for training and 35 different paths of USA network for testing. The prediction power of the each proposed model is estimated by calculating the $\Delta GSNR$, where $\Delta GSNR = GSNR_{Predicted}$ - $GSNR_{Actual}$. The proposed models are simulated on a workstation having specifications, 32 GB of 2133 MHz RAM and an Intel® Core™ i7 6700 3.4 GHz CPU.

3.6.1 Scikit-learn© Based Learning Methods for QoT-E

In this section, the proposed ML models' prediction performance based on *SKL's* API: DTR, RFR, and MLPR is reported . In Fig. 3.11 distribution of $\Delta GSNR$, a prediction error metric is plotted for DTR, RFR, and MLPR for the test samples of

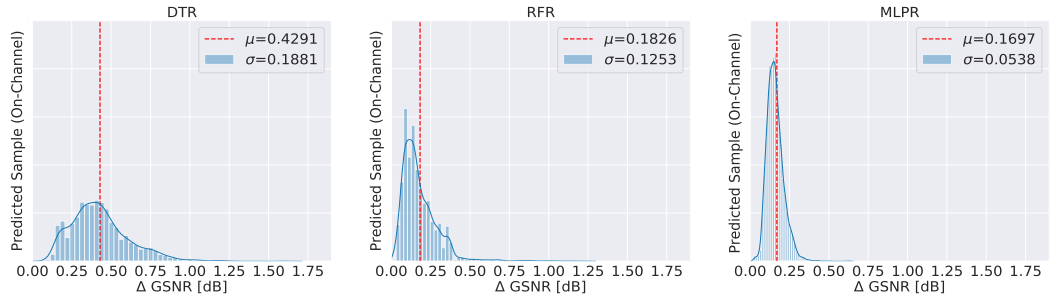


Fig. 3.11 *SKL* Based Learning Methods

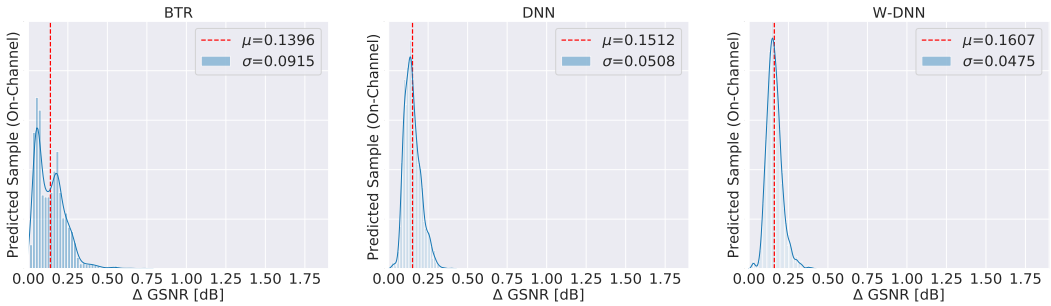


Fig. 3.12 *TF* Based Learning Methods

on channels realization only. For the given simulation scenario, the DTR is unable to find the underlying relationship and irregularities. On the other hand, the RFR took advantage of averaging various decision trees instead of a single decision tree, trained on randomly selected subsets of the training samples. Therefore, the overall performance of the RFR is much better than the DTR. Furthermore, the MLPR performed very well due to its cognitional potentiality provided by the internally configured neurons compared to the DTR and RFR. The above descriptive results are verified by observing the μ) and σ of $\Delta GSNR$ distribution against each proposed model in Fig. 3.11.

Furthermore, observing Fig. 3.13a, the box plot of $\Delta GSNR$ also depicts that the prediction accuracy of MLPR, which outperforms DTR and RFR. In Fig. 3.13a, the central rectangle span specifies the first quartile (Q_1) to the third quartile (Q_3). A segment inside the rectangular box shows the *median* of $\Delta GSNR$ and *"whiskers"* around the rectangular box shows the minimum and maximum values of $\Delta GSNR$. Focusing on MLPR, after observing Fig. 3.11 and Fig. 3.13a,it is quite obvious that MLPR is the best learning method among *SKL* based learning in the present simulation scenario. The results related to the MLPR are further analyzed by showing the bean plot of the $\Delta GSNR$ distribution against all the test paths for

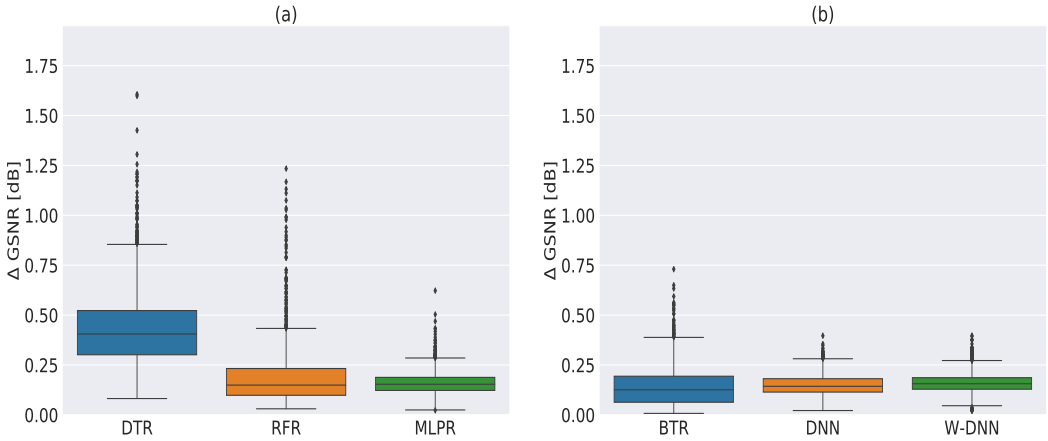


Fig. 3.13 (a) *SKL* Based Learning Methods (b) *TF* Based Learning Methods

the *agnostic/unused* USA network (Fig. 3.14). In Fig. 3.14, the shaded area of the bean plot shows kernel density estimate of the probability density function per point along with a line-scatter plot of all individual data points. The *"whiskers"* above and below the bean show the out bound of $\Delta GSNR$, the black segments show individual observation along with the μ value (*red segment in each bean*) of $\Delta GSNR$ for each test path.

After demonstrating the performance prediction capabilities, further analysis has been done related to the training time of each *SKL* based learning method, shown in Fig. 3.15a. The Fig. 3.15a, shows that the proposed MLPR requires a longer training time compared to RFR and DTR, due to its internal fully connected *hidden perceptrons*. The RFR takes a slighter longer duration than the DTR because of its dependency over the *bagging technique*.

3.6.2 TensorFlow© Based Learning Methods for QoT-E

In this section, a similar analysis for *TF* based learning methods has been performed. The proposed ML models prediction performance based on *TF's* API considers three cases, depicted in in Fig. 3.12: BTR, DNN, W-DNN. Fig. 3.12 shows the distribution of $\Delta GSNR$ for W-DNN, BTR and DNN for the test samples of *ON channels realization* only. For the given simulation scenario, the BTR takes advantage of the *boasting technique*, combining various models of the regression trees and select the new tree that best reduces the *loss function*, instead of choosing a random

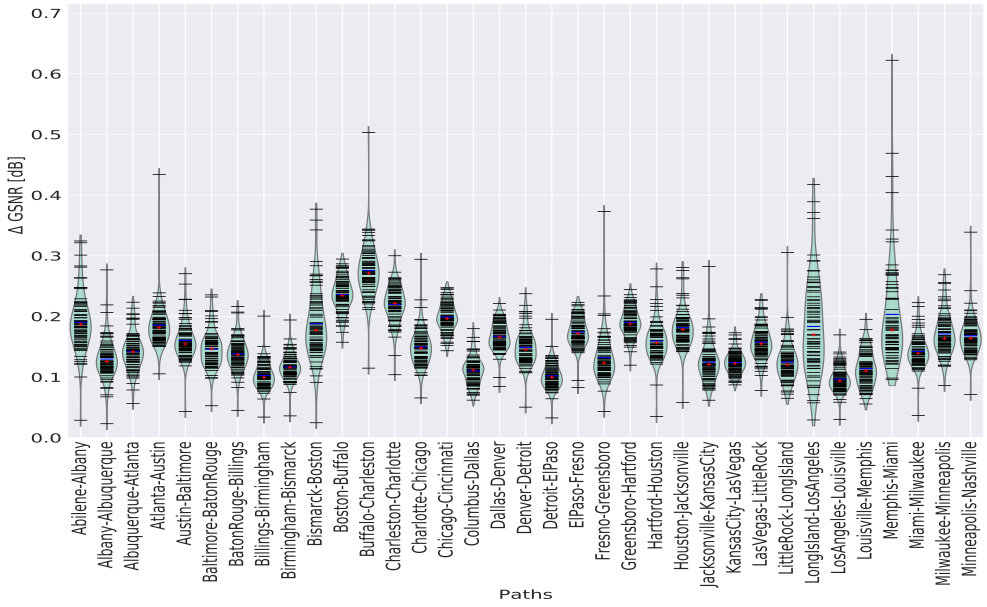


Fig. 3.14 Distribution of $\Delta GSNR$ for the simulated links of USA network using *MLPR*

one. On the other hand, the DNN outperforms BTR due to its cognitional potentiality, provided by the additional internally configured neurons as opposed to BTR. Finally, the W-DNN gives more refined tuning with respect to the DNN, as it combines the LR and DNN functionalities. The wider LR gives more refined tuning with respect to the DNN by characterizing features at the output layer. The above-described results are verified by observing the μ and σ of $\Delta GSNR$ distribution for each proposed model in Fig. 3.12.

In order to better visualize the comparison among W-DNN, BTR and DNN, Fig. 3.13b shows the box plot of $\Delta GSNR$ distribution for each *TF*-based

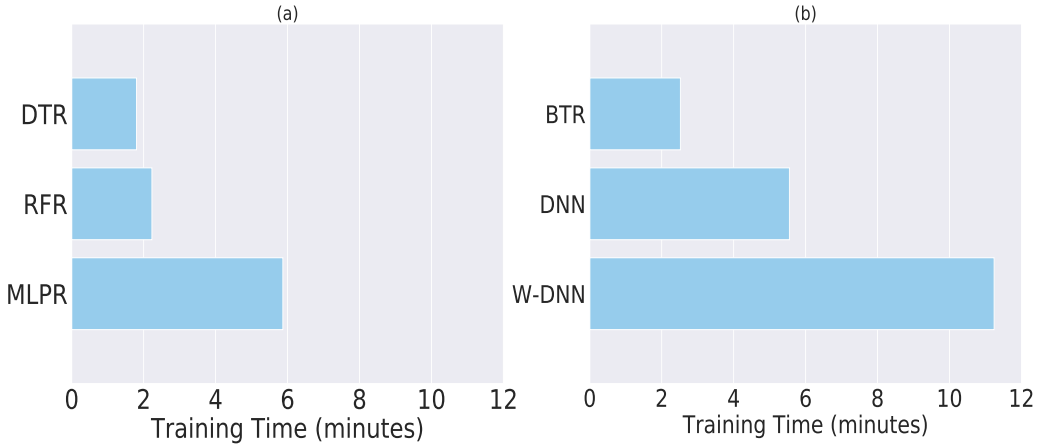


Fig. 3.15 (a) Single run training time of *SKL* based learning methods (b) Single run training time of *TF* based learning methods

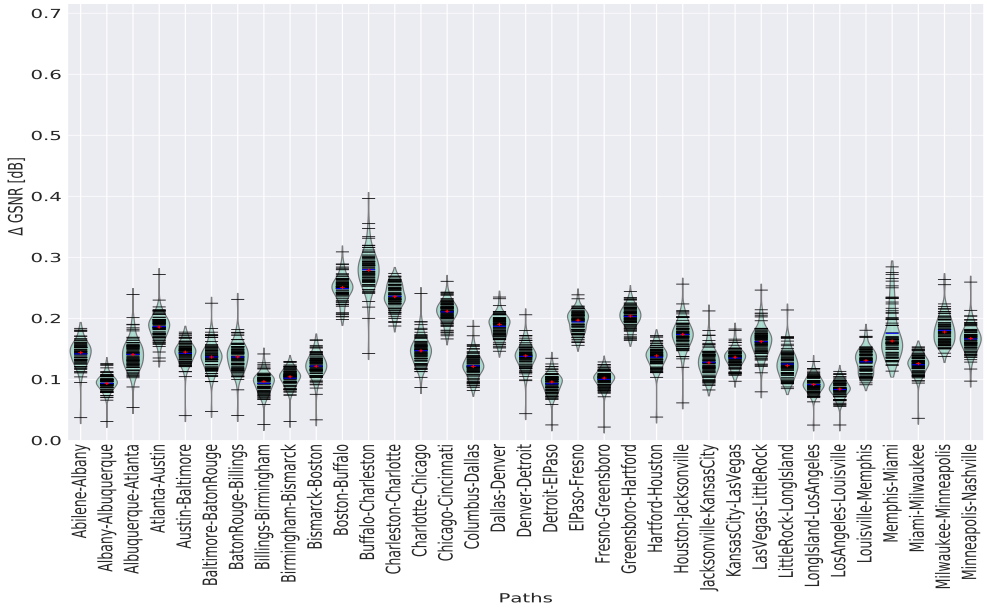


Fig. 3.16 Distribution of Δ*GSNR* for the simulated links of USA network using *wdnn*

learning method. Focusing on W-DNN after observing Fig. 3.12 and Fig. 3.13b, it is quite obvious that W-DNN is the best *TF*-based learning method in the proposed simulation scenario. To further analyze and elaborate the results related to the W-DNN, a bean plot of Δ*GSNR* distribution of all the test paths of the *agnostic/unused* USA network is shown in Fig. 3.16. Furthermore, the analysis of the training time for each *TF* based learning method is depicted in Fig. 3.15b. Fig. 3.15b shows that the proposed W-DNN requires a longer training time compared to the DNN and BTR, due to the synergic training of LR and DNN. The DNN takes a longer duration than the BTR because of its internal *hidden-layers* containing several neuron units. Finally, the prediction ability of the finest models of the two libraries is analyzed, MLPR and W-DNN, against all the test LPs. Observing Fig. 3.16 and Fig. 3.14 the W-DNN shows maximum Δ*GSNR* = 0.40 dB against against *Buffalo* → *Charleston* LP while the MLPR shows maximum Δ*GSNR* = 0.62 dB against *Memphis* → *Miami* LP. The remarkable performance of the W-DNN model is largely due to the synergic training of both the LR and the DNN, which enables it to outperform the traditional MLPR and DNN. Furthermore, the classical MLPR shows a large deviation of Δ*GSNR*, due to its dependency on the gradient-based local search technique, which is prone to unwanted local minima convergence during training.

3.7 Synergies use of machine learning agent and QoT-E engine of network

This section extends the results already presented in the previous section. In this section, the ML model is synergically used with the QoT-E engine of the network. The ML model is similarly trained on a dataset of *in-service* network but this time it is used to correct the error in GSNR computation which is estimated by QoT-E running in the controller of a sister *unused* network using nominal parameters of NEs . The error is typically created due to the variations in the operational point of NEs. The ML model proposed in the present scenario to correct error created due to the variations in the operational point of NEs works as a transfer learning module. The datasets already generated in the previous section are utilized for training and testing the transfer learning module. The main cognitive unit of a transfer learning module is developed by using higher-level APIs of the TF library.

The primary motivation of this study is to decrease the uncertainty in the GSNR computation of an LP and, consequently, to enable reliable path computation to deploy the candidate LP at the minimum margin. The synergistic use of a transfer learning unit with the QoT-E engine may substantially reduce the inaccuracies, so enabling a reduction in the needed margin. A same disaggregated network scenario is considered, in which the network controller may rely on a QoT-E API. The QoT-E accurately evaluates the fiber span losses together with the introduced NLI and computes the G and ASE noise of the amplifiers. If the controller can get a reliable network status, i.e., an *exact snap-shot* of operational parameters for each NE, the QoT-E engine can estimate the GSNR with excellent accuracy, as shown, for instance, in [175, 159]. In the absence of an *exact* description of the system parameters, a network operator relies on their *nominal* descriptions. Typically, nominal values are the average parameters provided by the vendors. The estimator engine utilizes these values and calculates a nominal GSNR with some degree of uncertainty. The present study, relies on a dataset coming from an *in-service* network: after a statistical analysis on the GSNR data, according to the transfer learning paradigm, the resulting set can be used to train a ML agent to assist the QoT-E in managing an *agnostic/unused* sister network. As a sister network, a different topology is considered based on the same hardware: specifically, fiber type and EDFAs.

Along with these assumptions, random and spectrally-flat connector losses are also considered, and a Monte-Carlo analysis of EDFA G ripple and NF is performed. GSNR data is then analyzed to search for a common statistical characterization. The analysis also shows that the per-wavelength GSNR distribution can always be well approximated as Gaussian-distributed. This observation leads to propose a method to set the needed margin on a network, given the standard deviation of GSNR, with a fixed maximum tolerable OOS percentage. Then, the dataset from the *in-service* network is used to train an ML agent running in the controller on the sister network. The ML agent's scope is to correct the GSNR computation for LP QoT on an *agnostic/unused* sister network, whose nominal NE parameters have been perturbed to include a realistic degree of uncertainty that needs to be minimized by the ML. The method is tested on several paths on the *agnostic/unused* network. The results show that the trained ML agent substantially reduce the uncertainty in GSNR computation, consequently reducing the needed margin. Based on the statistical characterization of the GSNR distribution, an original method is proposed which sets the margin by a maximum tolerable OOS percentage, given the GSNR standard deviation.

3.7.1 Operational GSNR margin and statistical analysis of GSNR

In this section, the GSNR dataset of an *agnostic/unused* network and its statistics are analyzed in order to comment on the margin to be considered with respect to the QoT-E. Along with this, different possible approaches to define the margin: considering only variation ranges or relying on a statistical characterization of GSNR is also reported. In a scenario of *agnostic/unused* network, to estimate the GSNR, the network controller can rely only on the nominal description of system parameters reported in Table 3.1. Using only this nominal description of system parameters, the network controller estimates a *nominal* GSNR value. This estimated nominal GSNR has some degree of uncertainty due to the variation in the working points of the NEs Fig. 3.5 shows the GSNR distribution for all WDM channels and all realizations for the path *Lousiville* $\rightarrow$ *Memphis*: it can be observed that it is distributed around the average nominal value according to a probability density function (PDF) that can clearly be approximated as Gaussian. Fig. 3.6 shows the same results for all wavelengths on the same path. In this figure, the exact variation ranges are shown.

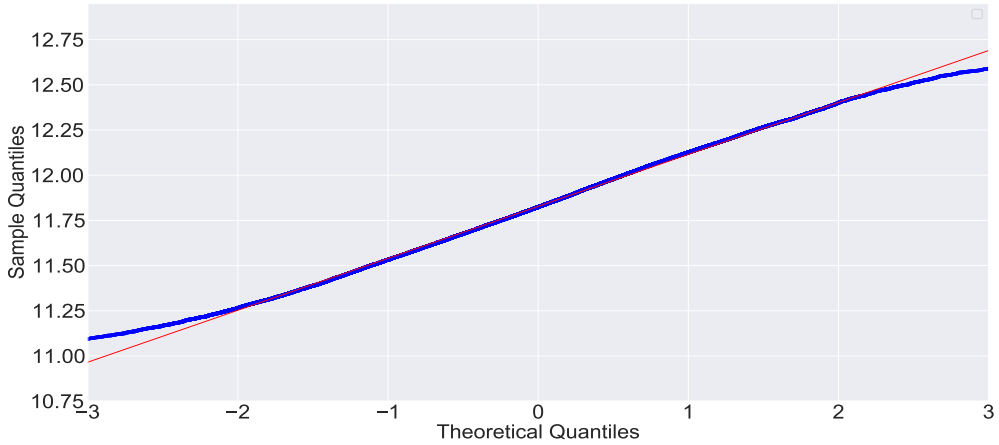


Fig. 3.17 Normal Q-Q plot of overall GSNR measurements for the path Louisville → Memphis

In general, it can be observed that the system uncertainties induce a variation of the actual GSNR ($\text{GSNR}^{\text{actual}}$) with respect to the nominal value ($\text{GSNR}^{\text{nominal}}$) that is supposed to be used for OLS control and computed by the QoT-E for LP deployment. So, in general, it gets an uncertainty

$$\Delta\text{GSNR} = \text{GSNR}^{\text{nominal}} - \text{GSNR}^{\text{actual}} \tag{3.5}$$

in the GSNR computation that must be taken into account. In particular, all cases when $\Delta\text{GSNR} > 0$ are critical because the actual GSNR is smaller than the estimated one and so relies on the QoT-E computation in these cases leads to unwanted OOS. To overcome such an issue, the operative GSNR ($\text{GSNR}^{\text{operative}}$) to be used for a reliable LP deployment is obtained by reducing the nominal one of a given GSNR margin δ: $GSNR^{\text{operative}} = GSNR^{\text{nominal}} - \delta$. Note that in this thesis, the margin needed by ripples in G, NF and variations in connector losses are the primary area of interest. Other uncertainties and hardware aging may require further enlargement of the the deployed GSNR margin. The following sections will discuss how to set the margin, first considering only the worst cases, then statistically approaching the underlying problem. According to the analysis, the main focus is on a specific path, but results can be generalized.

By analyzing the results displayed in Fig. 3.6, it can be observed that the actual GSNR varies around the nominal values with different worst-case *GSNR* per wavelength and if the $GSNR^{\text{operative}}$ is set equal to the worst-case *GSNR*, the margin

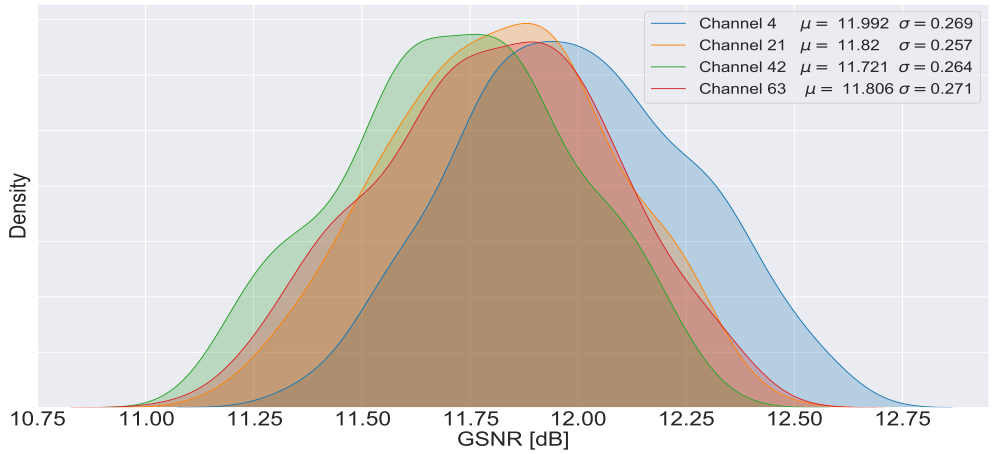


Fig. 3.18 GSNR measurements of randomly selected channels of Louisville → Memphis path

is given by $\delta = GSNR^{\text{nominal}} - GSNR^{\text{operative}}$. According to our dataset analysis, a variation of required δ ranging from 0.8 dB up to 1.8 dB is observed on the path under test. Considering to set a unique value for the margin over the entire exploited band, it means a request for 1.8 dB of margin, that will correspond to a large waste in potential capacity, mainly in the case of flexible transceivers [154]. Deploying a per-wavelength margin partially reduces the issues but keeps an open problem: being the GSNR fluctuation a random process around the nominal value, it is convenient to approach the problem statistically. Lets suppose to know – analytically or numerically – the PDF $f_{GSNR,\lambda}(x)$ for the GSNR fluctuations for every wavelength λ on a given path, the per-wavelength margin δ_λ can be set by a maximum tolerable percentage of OOS cases p_{oos}, according to the following expression:

$$\int_{-\infty}^{GSNR^{\text{nominal}}-\delta_\lambda} f_{GSNR,\lambda}(x)\,\mathrm{d}x \leq p_{\text{oos}}/100\,. \tag{3.6}$$

This approach can be further strengthened if an effective analytical approximation for the GSNR PDF can be found. In the following sections, we show how the per-wavelength GSNR *PDF* is always well approximated by a Gaussian distribution, so Eq. 3.6 becomes closed-form just depending on the per-wavelength mean and variance of GSNR distributions. By observing the GSNR statistics, the aggregated result for the overall GSNR is Gaussian. Even more, also the per-wavelength statistics suggest the same GSNR approximation. This Gaussian-*ity* of aggregated statistics of overall GSNR is verified in Fig. 3.17 by Normal Q–Q (quantile-quantile) plot,

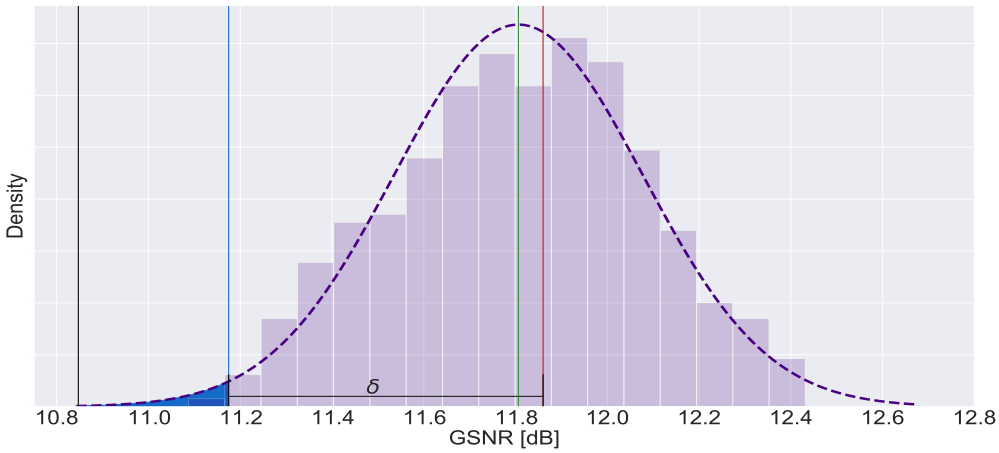


Fig. 3.19 Analytical GSNR margin estimation for channel-58 path *Louisville* → *Memphis*: green, red, blue, and black vertical lines represent the values of the distribution mean and the nominal, operative, and worst-case GSNR, respectively. The dashed line sketches a Gaussian shape with the same mean and standard deviation of the considered distribution.

which is one of the methods frequently used for the dataset Gaussian-*ity* test [176] because of its graphical representation. In parallel to this, the per-wavelength GSNR statistics are shown in Fig. 3.18, where the relative GSNR distribution for several randomly selected channels (4, 21, 42, 63) are plotted along with their μ and σ values. Also, in this case, Q-Q analysis leads to good results, confirming that for practical purposes, the GSNR statistics can be assumed as such. In general, the mean GSNR is not precisely equal to the nominal GSNR, but in this scenario it can be practically considered as such. Furthermore this simplifies the approach by just requiring per-wavelength standard deviation for the full characterization, and the margin setting Eq. 3.6 assumes the following expression:

$$\frac{1}{2}\text{erfc}\left(\frac{\delta}{\sigma_{GSNR}}\right) \leq p_{\text{oos}}/100\,, \tag{3.7}$$

which yields a closed-form expression for margin setting:

$$\delta \geq \sigma_{GSNR}\,\text{inverfc}\,(2\,p_{\text{oos}}/100)\ \text{[dB]}, \tag{3.8}$$

where inverfc is the inverse of the complementary error function erfc. A practical engineering rule could be setting $\delta = 3\ \sigma_{GSNR}$ that corresponds to $p_{\text{oos}} = 0.13\%$. These results open up future investigations on statistical regression on GSNR vari-

ations given by different line uncertainties, as the margin setting needs the GSNR standard deviation.

As an application example, considering the usual path under test and applying the statistical margin setting, with a maximum tolerable percentage of OOS of $p_{\text{oos}} = 1\%$: the margin requests varies ranging from 0.3 to 1.1 dB, depending on the channel. Margin setting is pictorially described in Fig. 3.19 for channel-58 of the *Louisville* $\rightarrow$ *Memphis* path. The blue shaded part shows the maximum allowable OOS, while the red line represents the nominal GSNR value for the considered channel. The black line represents the minimum value of GSNR. The GSNR margin is defined by the difference between the nominal GSNR (red line) and the operative GSNR (blue line).

The OOS penalty in the statistical approach of margin setting creates a road path for proposing a more flexible architecture. As the dataset of the already *in-service* network can be exploit to assist an *agnostic/unused* network controller in the estimation of GSNR. A data-driven technique such as ML architecture can be ideal in this kind of scenario where the operator has a dataset of an already *in-service* network. In this work, a trained ML module running over the controller of an *agnostic/unused* network is proposed. The proposed ML module is trained on the data provided by the already *in-service* network and specifically used to assist the core QoT-E engine of *agnostic/unused* network, to provide a correcting mechanism for the estimation of the QoT shown in Fig. 3.20. In Fig. 3.20, the network controller (USA network) fed its core QoT-E engine (GNPy) with nominal values of the system parameters, which leads to a nominal estimation of the GSNR, with some degree of uncertainty, due to variation in the working point of NE. This error (uncertainty) in the estimation of GSNR is corrected by using trained ML module, which is trained on the stored telemetry data of already *in-service* (EU network).

3.7.2 Transfer learning agent

This section describes the ML module, which is trained on the dataset of the already *in-services* network. Besides, the orchestration of the trained ML module specifying features, labels, and the additional configuration parameters of the ML algorithm is also described.

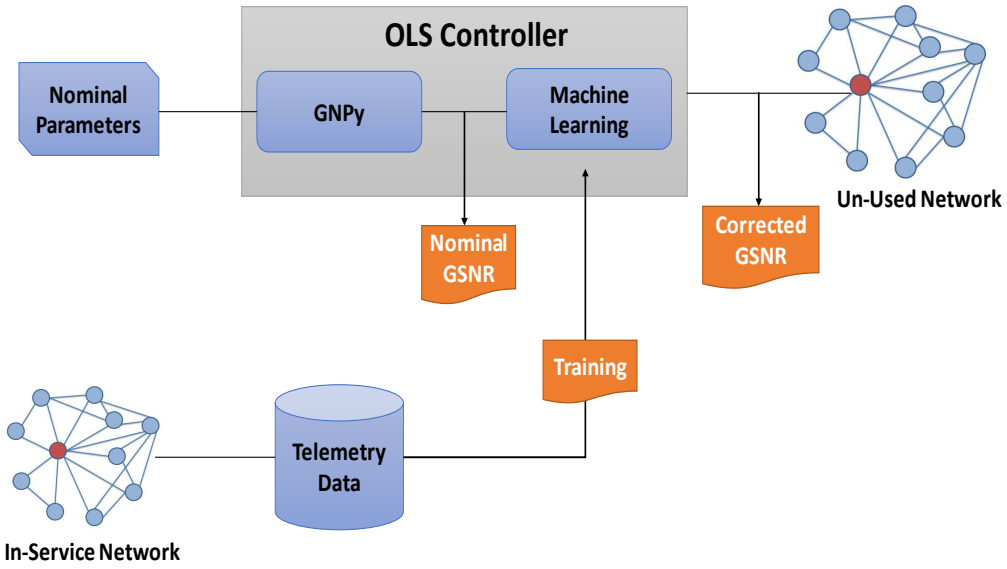


Fig. 3.20 Model Schematic: The Machine Learning module assists the QoT engine (GNPy)

The proposed work presents a trained ML module to assist the core QoT estimator engine in order to correct the estimated *GSNR* by QoT estimator engine of a specific LP before its actual deployment in an *agnostic/unused* network. DNN is selected as the ML algorithm [167], which is a powerful tool that has shown significant results in numerous frameworks as the one under investigation. The manipulated input features (system input) are similar to the previous analysis, i.e., the received signal powers, NLIs, ASEs, channel frequencies, and the number of spans between source to the destination node. The label (system output) in this study is different from the previous analysis and is fixed as a $\Delta GSNR$, the difference between the nominal and the actual GSNR values expressed in Eq. 3.5 (error in GSNR estimation), obtained for each channel: Fig. 3.21 shows the final ML module structure. The total number of input features are similar to the previous analysis consists of *305* entries, the number of LP spans plus the received signal power, the NLI, the ASE and frequency for each channel ($1+4\times 76=305$).

In order to enhance the performance of the DNN algorithm, it is trained on a normalized dataset [168].The proposed DNN is developed by using higher-level APIs of the TF platform [169]. The considered DNN is configured by several parametric values, such as *training steps* = 1000, loaded with default ADAGRAD keras optimizer with default *learning rate* = 0.01 and default L_1 regularization = 0.001 [173]. Moreover, *Relu* has been selected to empower DNN as it outperforms the

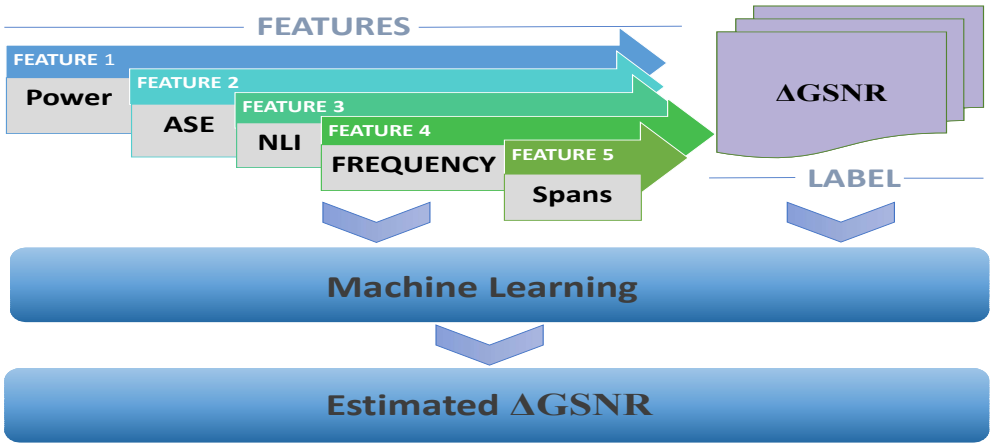


Fig. 3.21 Description of the Machine Learning Module.

others in terms of prediction and computational load [65]. Finally, another important DNN configuration parameters are the number of *hidden-layers* and neurons per *hidden-layers*. Both of theses parameters are already optimized in section 3.5

After the optimization, a DNN with *3 hidden-layers* containing *20 neurons* each is selected. Given these configurations, the training, validation, and testing on three separate subsets of the *in-service* EU network dataset is performed; the conventional rule of 70%,15%, and 15%, respectively, is chosen, as subset proportions. The train set in this scenario consists of data realizations for four (s $\rightarrow$ d) pairs of the *in-service* EU network already described in Sec. 3.4. The *training steps* is set as the stopping factor and MSE as loss function of the model given by:

$$\text{MSE} = \frac{\sum_{i=0}^{n} \left(\Delta\text{GSNR}_i^{\text{p}} - \Delta\text{GSNR}_i\right)^2}{n}, \tag{3.9}$$

where n is the number of tested realizations and, for each tested case i, $\Delta\text{GSNR}_i^{\text{p}}$ and ΔGSNR_i are, respectively, the predicted and calculated errors given by the nominal QoT-E. Once the accuracy level of the model predictions has been reached, the trained ML module can be used together with the core QoT estimator engine (GNPy) to enhance the GSNR estimation of the LP before its actual deployment in an *unused* USA network.

3.7.3 QoT-E correction using transfer learning agent

In this section, the comparison of the GSNR predictions and the consequent required margins in both the QoT estimation with and without the ML model support is made.

In both these scenarios, the statistical characterization of the GSNR distribution holds the Gaussianity. Thus the required margin evaluation can be performed by imposing a fixed percentage of OOS as described earlier. To summarize, due to the degree of uncertainty of any GSNR prediction, system margins are required in order to avoid unwanted OOS events. These margins lead to a resizing of the actual LP capacity. To have a uniform comparison metric that quantifies the performance of different GSNR evaluation methods, the following expression is formulated:

$$\Delta = \sum_{i}^{n} \left| \frac{\mathrm{GSNR}_i^{\mathrm{operative}} - \mathrm{GSNR}_i^{\mathrm{actual}}}{n} \right| \tag{3.10}$$

where n is the sample dimension and the operative $\mathrm{GSNR}^{\mathrm{operative}}$ take into account both the GSNR predicted by the specific method used for the GSNR estimation and the GSNR margin obtained by means of Eq. 3.8. In this investigation, the test set includes six (s $\rightarrow$ d) pairs of the *agnostic/unused* USA network already described in Table 3.3. Moreover, the analysis is further reduced to those realizations that have a lower actual GSNR with respect to the nominal one, as these cases are more critical, since they result in an OOS event if no margin deployment is considered.

In order to provide a clear description of the obtained results, the first focus is on the presentation of the results to a single path, *i.e. Louisville* $\rightarrow$ *Memphis* of the *unused* USA network. Given the actual GSNR synthetic dataset, the ground reference is set as the Δ evaluation obtained by considering the minimum GSNR value. This solution represents a rough approach, and it is presented to provide a reference scale. This method produces an average margin of $\Delta = 0.98\,\mathrm{dB}$ on the worst-case scenario channel. The worst-case scenario is fundamental in this type of analysis as it sets the required margin to guarantee all channels to be in-service. The QoT-E engine significantly refine the previous approach, providing a nominal GSNR value. In this case, without the ML support, the average margin is $\Delta = 0.69\,\mathrm{dB}$ on the single path *Louisville- Memphis* for the worst-case scenario channel requiring a $p_{\mathrm{oos}} = 1\%$.

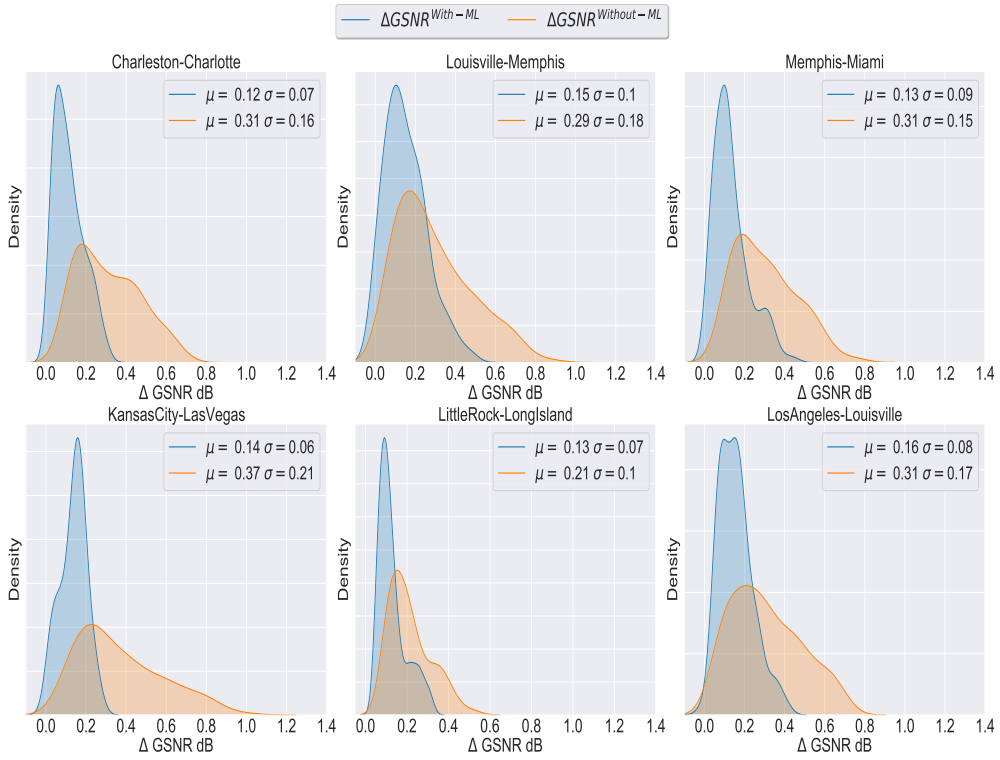


Fig. 3.22 ΔGSNR distributions obtained with and without ML module for all the investigated path of the *unused* USA network

Finally, when the QoT-E engine is supported by the ML trained module, the proposed implementation provides a final QoT-E that enables to reach an average margin of $\Delta = 0.27$ dB on the same worst-case scenario channel requiring a $p_{\text{oos}} = 1\%$.

The reliability of the joint estimation of the QoT-E engine with the trained ML module has been further verified on five more paths of *agnostic/unused* USA network reported in Table 3.3. For all the considered paths, in Fig. 3.22 the distribution of the prediction error, both with and without ML support, are reported. In the figure, it can be observed that the ML agent significantly enhances the QoT-E; in all the cases, both the error μ, and σ, values decrease considerably. Therefore, the required margin calculated with Eq. 3.8 can be reduced maintaining the same p_{oos}. For example, on the *Louisville* $\rightarrow$ *Memphis* path, the ML module application reduces the margin required to reach p_{oos} from 0.76 dB to 0.58 dB. This leads, in conclusion, to a smaller average margin and, therefore, to more extensive exploitation of the LP capacity.

Chapter 4

Machine Learning for Photonic Devices

Introduction

The rapid increase in internet traffic due to bandwidth-intensive applications and the latest developing concepts of the IoT require higher degrees of flexibility at each network layer. The implementation of SDN has the potential to provide them effectively. Adopting the SDN paradigm enables the complete virtualization of network elements and functions inside the network operating system. Moreover, technologies like coherent optical techniques for WDM optical transport and reconfigurable optical switches for transparent wavelength routing pave a path to extend SDN applications down to the physical layer [177]. To achieve SDN implementation down to the physical layer, network key elements and transmission functionalities must be abstracted for QoT penalties and control states. This abstraction empowers the optical network controller to fully manage network elements and transmission functionalities [178].

Nowadays, network components are increasingly utilizing PICs to execute different complex operations. Specifically, in the latest *smart* optical networks and data centers, large-scale photonic switches and wavelength selective switches play a prominent role due to their wide-band capabilities, minimal latency, and low power consumption. These distinctive properties increase the possibilities of using PICs-

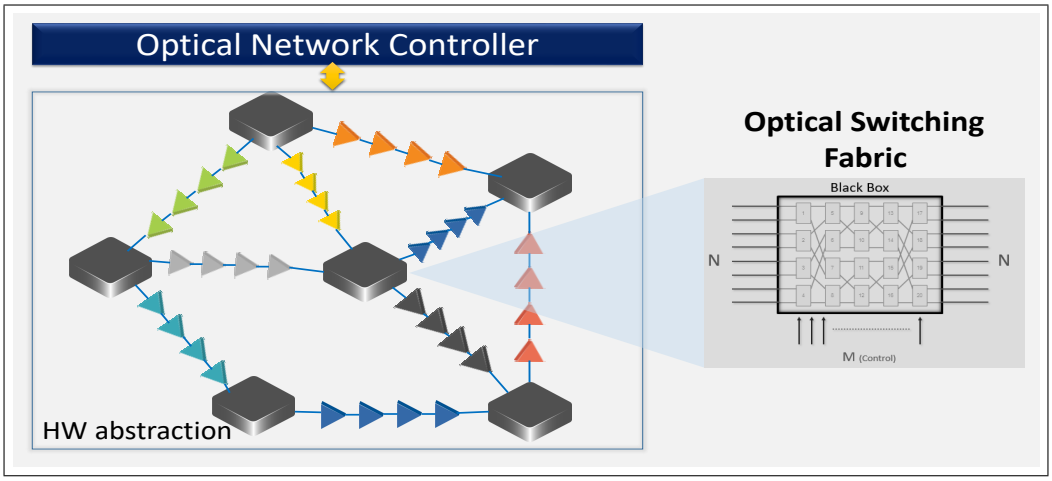


Fig. 4.1 Abstraction of the optical switch in a SDN-controlled optical network.

based network elements, especially photonic switches, and hence they generate a demand for a generic softwarized model for control states and QoT degradation to enable full control by a centralized controller, as shown in Fig. 4.1.

In this Chapter, we propose a generic topology-agnostic *blind* model exploiting an ML inverse design approach to obtain the softwarized control of any $N \times N$ photonic switching system and direct design method to predict the QoT degradation due to the switching element. The chapter begins with a brief explanation of the elementary switching module along with the topologies considered for the analysis in Section 4.1. After that, the simulation model, dataset generation, and the routing and transmission models are discussed in Section 4.2. Later, in Section 4.3, the ML models proposed for the control states of a PIC-based $N \times N$ photonic switching system is discussed. After this, the performance analysis of all the proposed models for predicting the control states are given in Section 4.4. After this analysis, in Section 4.5, ML model is proposed to predict the QoT degradation due to the switching element. In Section 4.6, the ML-based softwarized and QoT aware control and management system for any $N \times N$ optical switch is reported. Finally, in Section 4.7, ML assisted control and management model for wideband switching fabrics is discussed.

4.1 Elementary switching module and topologies

The switching networks analyzed to validate the proposed ML-based management model belongs to a class of multistage crossover switches akin to the Banyan and Clos networks: these topologies are composed of several elementary 2×2 crossbar switches, arranged in multiple stages with variable interconnections, to route a generic number N of inputs to a required output configuration. The different configurations are defined by the control signals applied to each of the M optical switching elements (OSEs), which determine the output configuration of the device. From a topological point of view, the most important property for the optical application is to route each possible requested output permutation without internal conflicts.

Based on this property, the networks can be divided into two main classes: non-blocking and blocking, representing the ability to route all possible permutations of N inputs to the N output ports. For the scope of this analysis, only non-blocking networks have been analyzed, as they provide a more useful application and more complex topological characteristics with respect to the blocking counterpart.

4.1.1 2×2 crossbar switch

A 2×2 CrossBar switch is the basic building block used for the design of these networks. The 2×2 crossbar switch is defined as a two-state device, piloted by a control signal M, which toggle between the two configurations. The bar state, defined for $M = 0$, represent the straightforward propagation of the two input signals ($\begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix} \rightarrow \begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix}$), while in the cross state, for $M = 1$, the output signals order is reversed ($\begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix} \rightarrow \begin{pmatrix}\lambda_2\\ \lambda_1\end{pmatrix}$) shown in Fig. 4.2. As previously stated, this fundamental block can be physically implemented through a variety of approaches, with the most prominent two being the microring resonator (MRR) and the MZI. These implementations offer different performances based on the physical design and can be tailored to colorless or chromatic-dependent applications. The binary control signal present in the black-box model is typically provided through an electrical signal in the OSE, with a dependency on the device implementation. Nonetheless, on a virtual abstraction of the component, for the routing path evaluation, the binary model is suitable while maintaining a general device-independent scope.

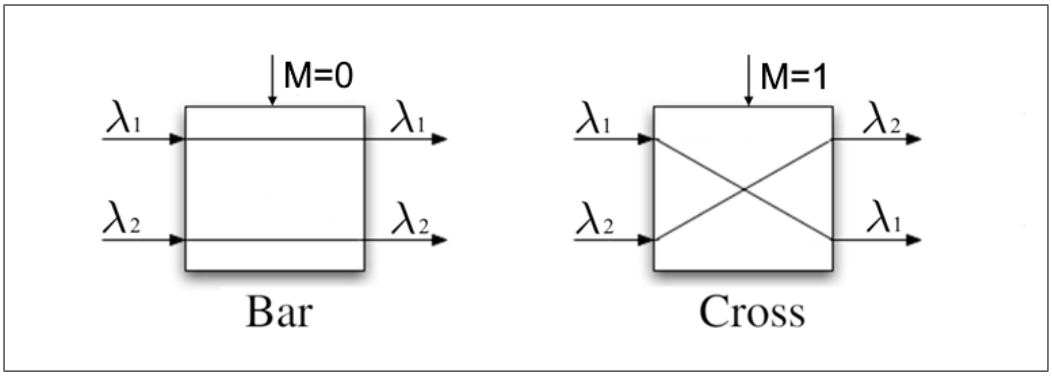


Fig. 4.2 Illustration of Bar and Cross states of a 2×2 elementary switching element (CrossBar switch).

4.1.2 Multistage Rearrangeable Non-Blocking switching architectures

To design an N×N switch, the crossbar elements must be placed in a suitable topology, which determines the properties of the switch, both in terms of routing capabilities and the number of elements required. The focus of this approach is directed toward a sub-class of switching networks which are defined as *Rearrangeable Non-Blocking* networks. Switching networks can be defined as non-blocking if all the possible permutations of the input signals can be routed to the output ports: any input-output (I/O) request targeted at an unoccupied port can be established without creating conflicts inside the network, also taking into account the already established I/O links.

In rearrangeable networks, this property is partially upheld, as routing all permutations is achievable, although potentially requiring reconfiguration of the previously established I/O connections. In this class of multistage networks, the reconfiguration of the switches is required, as the topology does not guarantee path availability if traffic is already present in the device. This is a clear disadvantage with respect to strict-sense non-blocking structures, as it requires the implementation of a more complex control unit to evaluate the routing and the conflicts inside the network. The trade-off is acceptable in most applications, as this wider-sense property allows most topologies to implement an N×N switch with fewer elements with respect to the strict-sense devices, with an apparent reduction of transmission losses, power consumption, and footprint.

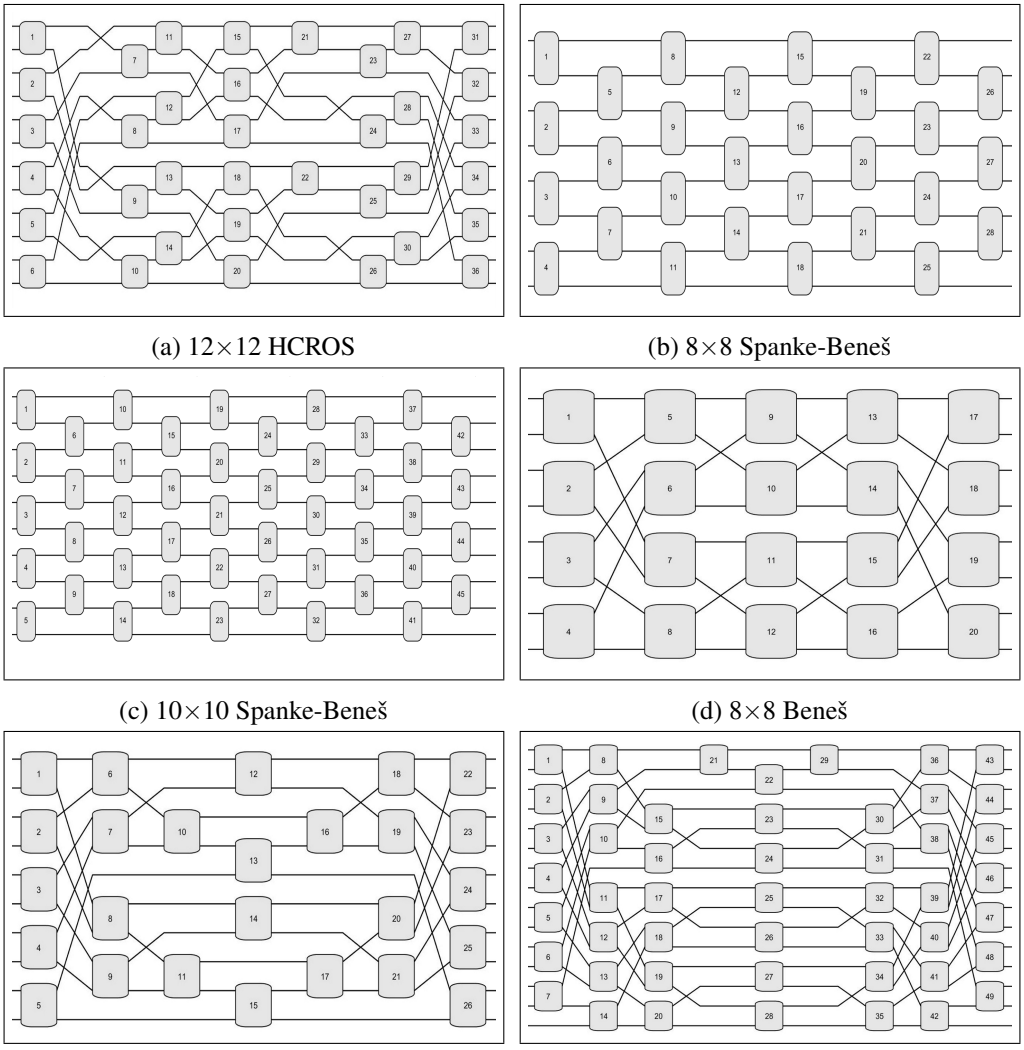
4.1.3 Problem complexity

Considering a generic device in this class, the two main parameters determining the model complexity are the number of inputs N or the network size, and the number of needed OSE, which correspond to the number of variables to establish a target routing path. The solution space of all the output permutations ($N!$), as well as the states configurations (2^M) grows as a non-polynomial (NP) function as the network size increases. This is due to the dependency of the number of OSE on the number of inputs. Typically for these networks, the relationship follows $M = O(N \cdot \log_2(N))$ with a small degree of variance between the different available topologies. This phenomenon leads to scalability issues concerning traditional topology-independent path-finding algorithms; as the NP complexity of available solutions increases, it cannot be directly approached. Topology-specific routing algorithms exist for each class, although this solution bears two main disadvantages: it requires a rigid control unit that cannot pilot a device with different topology and the unavailability of researching an optimal solution.

Multi-switching networks, especially in the optical domain, are prone to path-dependent degradation of performance, with a wide range of QoT between the equivalent paths available for the same output configuration. The deterministic topology-dependant routing algorithm needs to evaluate all the equivalent paths, with a complexity dependence on N as $N_{eq} = O(2^M \div N!)$, leading to severe scalability issues. As such ML-based methods can overcome the limitation and can be trained on performance-aware datasets. Under this scenario, the NP size of the solution space becomes an asset instead of a disadvantage, allowing the generation of a large dataset for training the specific ML agent.

4.1.4 Topologies under analysis

In order to test the performance of the proposed method, both the scalability and the robustness with respect to the topology variation must be tested. To this end, three main topological structures were tested and are depicted in Fig. 4.3. The first network under analysis is the Beneš switch. This device follows a recursive structure based on the Clos network paradigm, with number of OSE $M = N\log_2(N) - \frac{N}{2}$. The Beneš network is a common approach to multistage switching networks. It is characterized by a low number of 2×2 switching elements, implying reduced



(a) 12×12 HCROS

(b) 8×8 Spanke-Beneš

(c) 10×10 Spanke-Beneš

(d) 8×8 Beneš

(e) 10×10 Beneš

(f) 15×15 Beneš

Fig. 4.3 Multistage switching networks topologies under analysis

footprint and power consumption with respect to larger topologies. Three instances of Beneš structures have been tested, with network size $N = 8,\ 10,\ 15$ and $M = 20,\ 26,\ 36$.

An alternative to the recursive Beneš structure is the Spanke-Beneš network: this topology is distinguished by its planarity, as no crossing interconnection is used between the switching stages. The planarity comes as a cost in terms of a number of OSE, which are equal to $M = \frac{N \cdot (N-1)}{2}$, increasing the already severe effect of the NP complexity growth. This topology is still considered for applications where the crossing technology cannot be relied upon to guarantee the needed QoT; as such, the

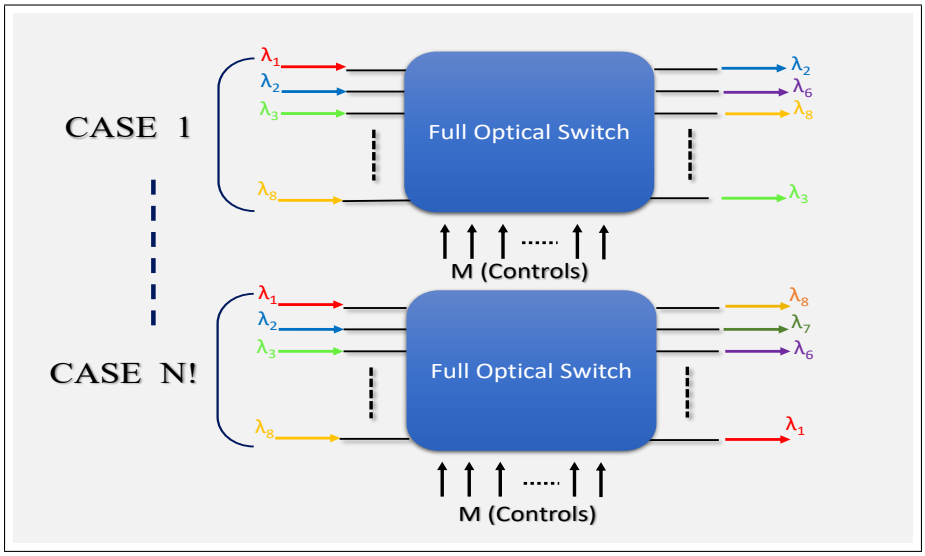


Fig. 4.4 Graphical representation of all possible $N!$ output states for a $N \times N$ fabric.

size $N = 8,\ 10$ and $M = 28,\ 45$ have been chosen to allow similar complexity to the Beneš networks under analysis.

The third considered topology is an optimized network with an equal footprint to the Beneš counterpart. The device doesn't uphold the planarity constraint of the Spanke-Beneš nor the recursive generation of the traditional Beneš networks. This structure, referred to as the Honeycomb rearrangeable optical switch (HCROS) [179], shows an asymmetric topology with respect to the traditional implementations, acting as a valuable benchmark to test the proposed ML-method robustness related to irregular and unique structures. The proposed device is extended to a 12×12 structure to offer a valid comparison to the other switching devices under analysis.

All three considered structures belong to the same class with respect to non-blocking properties, as they require rearrangement of the control states to route new input-output links when traffic already occupies part of the circuit. Similarly, each topology has more OSE configuration states than signal output permutations; as such, each device can obtain the same output configuration through a different number of equivalent paths.

4.2 Simulation model and dataset acquisition

The simulation and data generation have been modeled following two main strategies. Concerning the routing evaluation, the black-box model for the fundamental 2×2 element is sufficient for the path evaluation, as the input-output link can be represented as an edge in a suitable graph structure. For the evaluation of the QoT, the logical model is insufficient; as such, the device has been simulated, considering an MRR-based implementation of the crossbar elements.

4.2.1 Routing model

For evaluating the routing states inside the device, the network has been implemented through a matrix representation, composed by cascading the permutation vectors of each switching stage. The output permutation can be obtained as a function of a given control state, providing the BAR and CROSS configuration for each of the M elements. The datasets obtained through this approach are composed of a binary control vector $V \in \mathbb{R}^{1,M}$, with $V_i = 0$ representing the BAR state and $V_i = 1$ the CROSS alternative configuration. In contrast, the output configuration is represented by a permutation vector of size N shown in Fig. 4.4. This logical model is also fundamental in the verification step: the predicted control states are evaluated directly on the abstracted device to verify the correctness of the ML solution, taking into account the alternative routing states for the required configuration.

4.2.2 Transmission model

In order to gather data concerning the QoT, the device must be simulated with a higher degree of realism, taking into account both the physical design of the components and transmission format.The crossbar has been modeled as a second-order MRR switch, and following the recursive definition, the simulation-ready model is created in the Optsim© environment [180]. The ingress and egress stages of the device have been connected to a transceiver and receiver module, respectively, as shown in Fig. 4.5, allowing QoT simulations under a realistic modulation format and transmission characteristics. The system has been tested under a PM-64-QAM

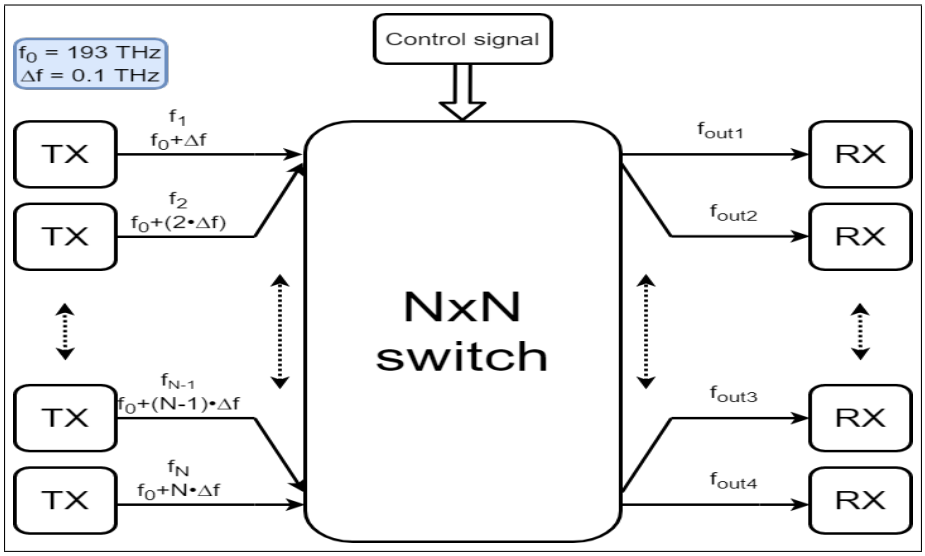


Fig. 4.5 Generic $N \times N$ transmission model

modulation format, with central frequencies $f = (193.1 + 0.1 \times x)$ THz for $x \in [1, N]$ and symbol rate $R_S = 50 \times 10^9$.

4.3 Machine learning for routing management of photonic devices

The standard ML framework allows the translation of the effective system attributes, which cannot be easily or directly measured. Generally, ML models develop cognition capability by exploiting a series of intelligent algorithms that can understand the training data intrinsic information. The information inherited by the intelligent algorithms is then abstracted into the decision models that manage the testing phase. These well-trained cognitive models provide real-time opera-

Table 4.1 Dataset Statistics

Network type / **Size** ($N \times N$)	**Beneš** **8x8**	**Beneš** **10x10**	**HCROS** **12x12**	**Beneš** **15x15**	**Spanke-Beneš** **8x8**	**Spanke-Beneš** **10x10**
Permutations ($N!$)	40,320	3,628,800	479,001,600	$1{,}307 \times 10^9$	40,320	3,628,800
Switches (M)	20	26	36	49	28	45
Combinations (2^M)	1,048,576	67,108,864	68×10^9	562×10^{12}	268,435,456	35×10^{12}
Dataset	100,000	300,000	300,000	1,000,000	300,000	1,000,000

tional improvements by enabling the system to draw *smart* conclusions and react autonomously.

In the current thesis, five distinct ML models have been proposed to model the control states of a PIC-based $N\times N$ photonic switching system. The proposed ML framework consists of three basic units; pre-processing, training, and testing. The pre-processing section standardizes the dataset before utilizing it in the training section. The training section exploits the standardized train set for the training of the proposed models. Following training, the testing unit uses a subset of the data to initiate the testing phase. The proposed ML models are developed by using high-level python APIs of two open-source ML libraries, TF [169] and SKL [168]. Both of these libraries provides a vast range of APIs for data-driven models and different functions to pre-process and clean the dataset from the noise before applying it as an input to the ML model.

The proposed ML-based methods operate in a complete black-box set up, requiring a sufficiently large amount of training data to develop the cognitive models, without having to consider the underlying photonic circuits internal structural. We evaluate five ML techniques and compare the prediction performance in the proposed framework of the investigation. Like all other supervised ML-based learning methods, to perform the training and prediction processes, the proposed model requires the definition of the features and labels representing the system inputs and outputs, respectively. The manipulated features (system input) comprise the numerous permutations of the input signals (λ_1, λ_2, λ_3....λ_n) at the output ports of the switch, and it exploits its M control signals as label (system output) shown in Fig. 4.6. Initially, the training of the ML models is performed. After that, we tested the trained models on the independent subset of the dataset; the standard rule of 70% and 30% has been preferred to set the subset ratios. In order to avoid over-fitting the models, for each particular M we set the *tree size* (for all tree regression models) and *training steps* (for LR and DNN) as the stopping factor and the *MSE* as the loss function, given by:

$$\text{MSE} == \frac{1}{n}\sum_{i=0}^{n}\left(\frac{1}{M}\sum_{m=1}^{M}\left(\text{Ctrl State}^{p}_{i,\text{m}} - \text{Ctrl State}^{a}_{i,\text{m}}\right)^2\right) \tag{4.1}$$

where n is the number of test realizations, M is the total number of switching elements in the specific $N\times N$ switching system, while for each tested case i, $\text{Control Signal}^{p}_{i,\text{m}}$ and $\text{Control Signal}^{c}_{i,\text{m}}$ are the predicted and actual control bits of the m-th switching

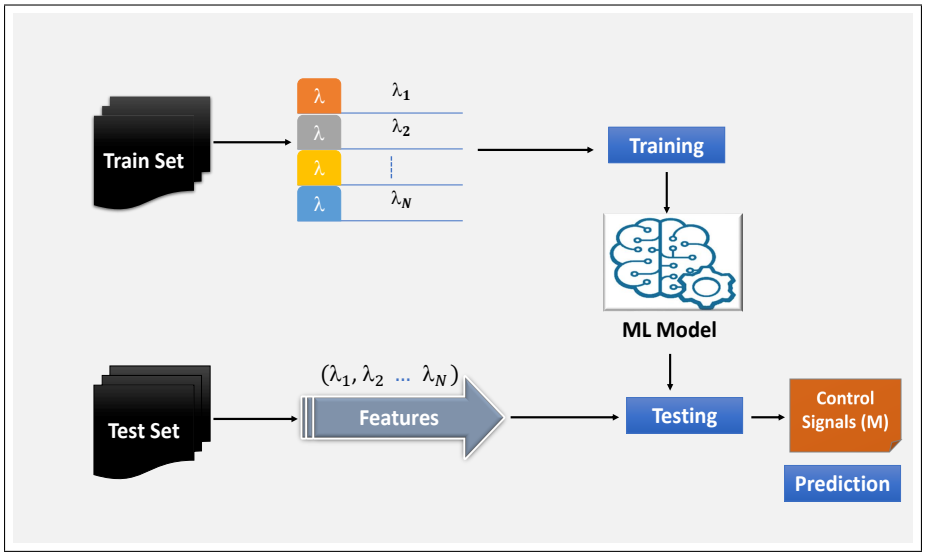


Fig. 4.6 Schematic of machine learning framework.

element of the considered configuration. The general tuning hyperparameters and the API used to build the proposed models are reported in Table 4.2. The hyperparameters for the proposed models are also optimized in the similar fashion as described in Section 3.5.

4.3.1 Decision tree regression

The DTR model is developed to model the control states of a PIC-based $N \times N$ photonic switching system. Normally, DTR provides direct relationships between the input and response variables [60] by constructing a tree based on various decisions made by exploring several dimensions of the provided features and ultimately provides the desired response variable. The proposed DTR has two key tuning parameters; *min_samples_leaf* and *max_depth*. The optimum values of these two main parameters are obtained by tuning them to achieve the best trade-off between precision and computational time in the proposed simulation environment.

4.3.2 Random Forest Regression

The considered RFR uses *ensemble* learning which is based on the *bagging* tree technique [61]. Similar to DTR the RFR has also two key parameters

Table 4.2 Machine learning Models Detail

Machine learning Model	API	Parameter	Value
Decision Tree Regressor	*scikit-learn*©	*Min samples leaf*	4
		Max depth	100
Random Forest Regressor	*scikit-learn*©	Method	*'Bagging'*
		Min samples leaf	4
		Max depth	100
Boosted Tree Regressor	*TensorFlow*©	Method	*'Gradient Boosting'*
		Min samples leaf	4
		Max depth	100
		Learning rate	0.01
		L_1 regularization	0.001
Linear Regressor	*TensorFlow*©	*Equation*	Linear
		Training steps	1000
		Method	*Ordinary Least Squares*
Deep Neural Network	*TensorFlow*©	*Hidden layers*	3
		Keras optimizer	*'ADAGRAD'*
		Activation function	*'ReLU'*
		L_1 regularization	0.001
		Learning rate	0.01
		Training steps	1000

min_samples_leaf and *max_depth*. The tuning of these two parameters is performed in such a way as to obtain the best trade-off between accuracy and complexity.

4.3.3 Boosted tree regression

The proposed BTR also uses *ensemble* learning, but in contrast to RFR, it is based on the *gradient-boosting* technique. Like the other tree regressors key parameters, we also tune these parameters for BTR to obtain the optimum values between precision and complexity.

4.3.4 Linear regression

LR is a kind of ML model which utilizes a statistical method to learn the linear relationship between the input feature (x) and the output response variable (y). Generally, the mathematical description of LR is as follows:

$$y = B_0 + B_1 x \tag{4.2}$$

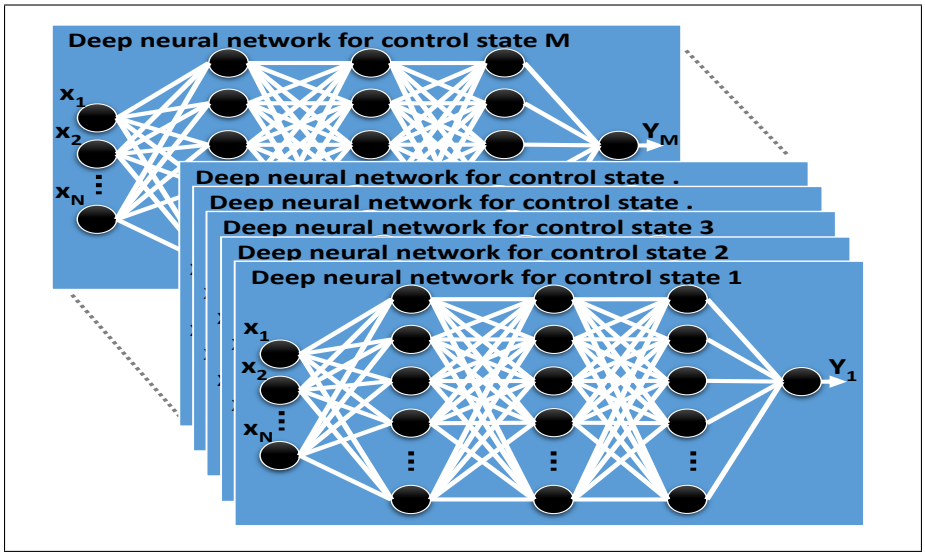
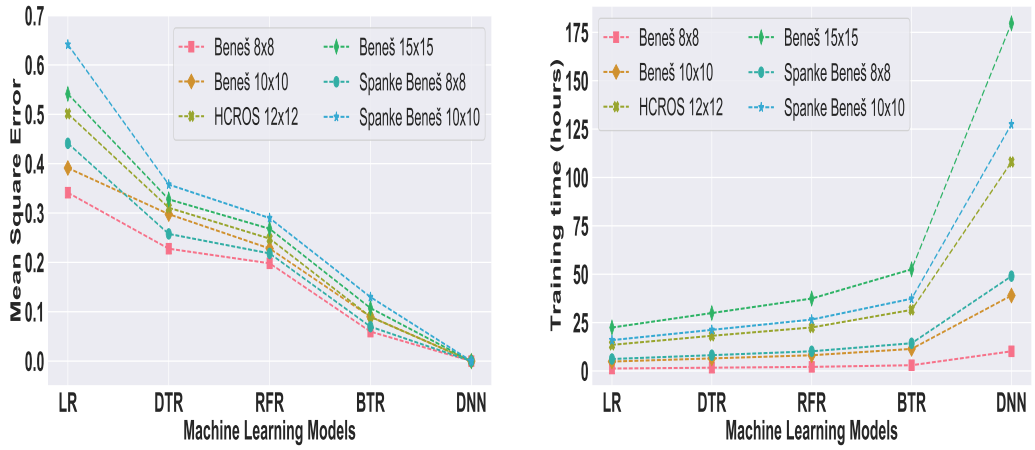


Fig. 4.7 Parallel architecture of DNN with hidden layers.

where y is the output variable, B_0 is the intercept, B_1 is the co-efficient of each variable, and x is the input features set. The model estimates the values of intercept (B_0) and the co-efficient (B_1). LR has a different kind of optimization strategy. In our work to model the control states of a PIC-based $N \times N$ photonic switching system, we applied the ordinary least square method that takes more than one input feature and requires no weighting function.

4.3.5 Deep neural network

The DNN is one of the most frequently used ML models inspired by the human nervous system to process information. To model the control states of a PIC-based $N \times N$ photonic switching system, the considered DNN is configured by several parametric values that have been optimized (such as the *training steps*), loaded with the *ADAGRAD*, *learning rate* and L_1 regularization [173]. Moreover, several non-linear activation functions such as *Relu, Tanh, sigmoid* have been tested during the model building. After testing, *Relu* has been selected to implement DNN as it outperforms the others in terms of prediction and computational load [65]. Another important DNN parameter is the number of *hidden-layers*. The model has been tuned on several numbers of *hidden-layers* and neurons to achieve the best trade-off between precision and computational time. Although an increase in the number of layers and neurons improves the accuracy of the DNN up to a certain extent, a further



(a) (b)

Fig. 4.8 (a) Mean Square Error of ML models (b) Training Time of ML models.

increase in these values introduces diminishing returns that cause over-fitting while simultaneously increasing the computational time. After this trade-off analysis, we decided upon a DNN with *three hidden-layers* with several cognitive neurons for each hidden layer optimized for each dimension N. To improve prediction accuracy, a parallel architecture for the DNN is proposed as shown in Fig. 4.7: in practice, an independent DNN architecture is considered for the prediction of each of the control states.

4.4 Performance analysis of machine learning models

This section describes the performance evaluation of numerous ML models developed using higher-level python API of the *SKL* and *TF* libraries. The numerical assessment of the proposed data-driven methods is illustrated in Fig. 4.8a against the different $N \times N$ photonics switching configuration (i.e., Beneš, Spanke-Beneš, and HCROS). Fig. 4.8a describes the MSE achieved against each of the proposed ML models for different considered $N \times N$ architectures. The MSE in the current simulation environment follows the following order LR>DTR>RFR>BTR>DNN for almost all the considered $N \times N$ configurations. The LR and DTR show the worst performance in terms of MSE as they cannot uncover the underlying relationship and irregularities. On the other hand, the RFR benefits from averaging numerous decision trees instead of a single decision tree trained on randomly selected subsets

of the training sample. Moreover, the BTR exploits the boasting technique, merging various regression trees' models and picking out the new tree that best degrades the loss function instead of randomly choosing. Therefore, the overall performance of the BTR is better than the RFR. Finally, the DNN performed remarkably well because of its cognitional capability, enabled by the internally aligned artificial neurons, compared to the RFR and BTR.

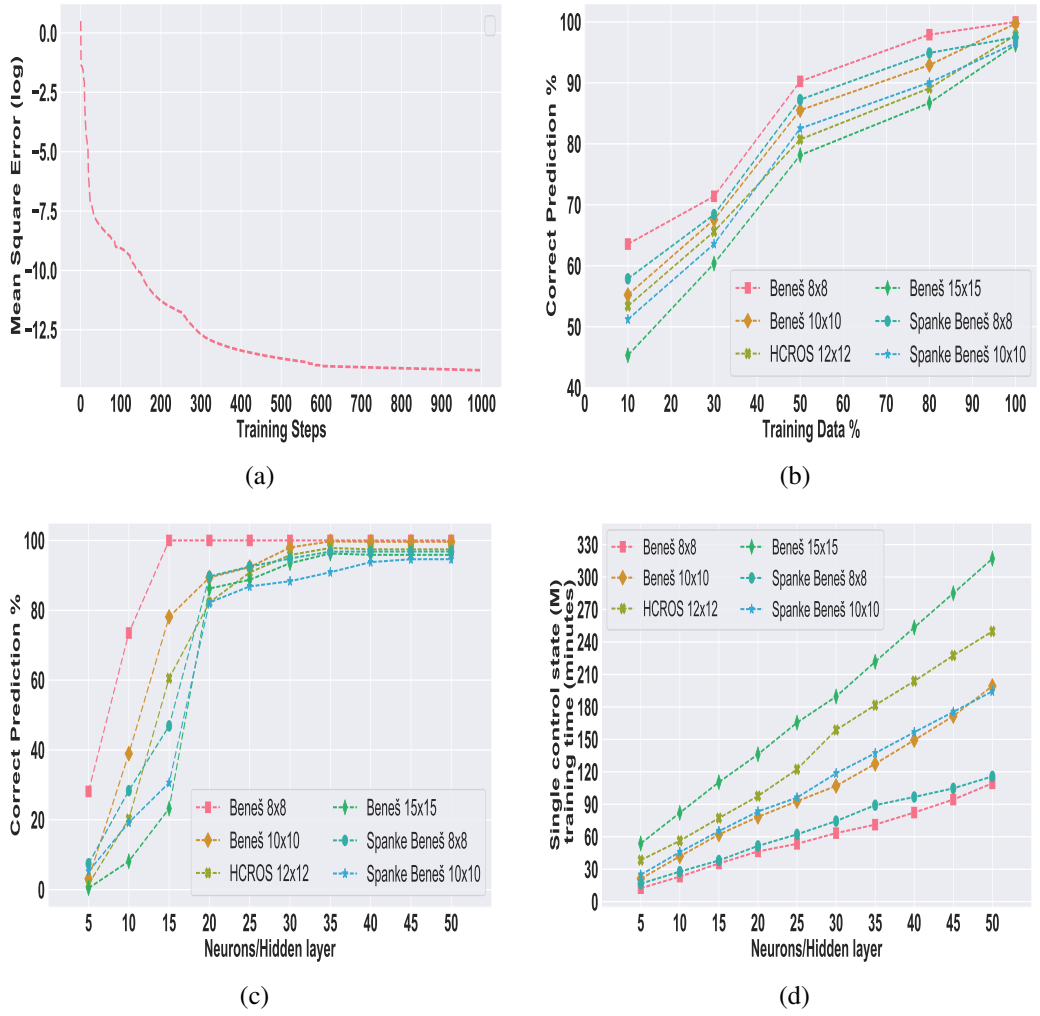


Fig. 4.9 (a) DNN loss function vs. the training steps for Beneš 8×8 architecture. (b) Percentage of correct predictions vs. normalized training dataset size. The normalization is performed with respect to the total generated dataset dimension for the considered N×N fabric, see data in Table 4.1. (c) Percentage of correct predictions vs. hidden layer size for the considered switching configurations (d) Single switch training time vs. hidden layer size.

The training time for a single control state M has also been analyzed, and is shown in Fig. 4.8b for all the proposed ML models. The training timing shows the reverse order as we observed during MSE analysis: LR<DTR<RFR<BTR<DNN. The proposed DNN solution takes a longer training time with respect to the other suggested models due to its internal hidden layers, containing numerous neuron units. The RFR and BTR take a slighter longer time than the LR and DTR because of their dependency on the bagging and boasting techniques. The proposed models are simulated on a workstation having specifications, 32 GB of 2133 MHz RAM and an Intel® Core™ i7 6700 3.4 GHz CPU.

Typically, in all the scenarios where data-driven models are exploited, the main objective of using these learning methods is their high accuracy compared to the training time. As the ML-based models only need initial training that takes a long time, but the testing can be done in real-time once the models are adequately trained. To this aim, we selected DNN as the preferable ML model to proceed with further investigation. In the rest of the paper, all results are obtained using the proposed parallel DNN approach. To further verify our selection, we observed the complete trend of the loss function (i.e., MSE) with respect to the training steps for the proposed DNN, shown in Fig. 4.9a for a single considered case of Beneš 8x8. Similar behavior is observed for all the other considered switching architectures.

The first assessment we performed is the prediction accuracy dependency on the dimension of the training dataset and the size of hidden layers shown in Fig. 4.9b and Fig. 4.9c. In Fig. 4.9b, the effect of increasing training dataset size is described. The trend reveals that the prediction capability of the proposed DNN improves with an increase in the training dataset size. Likewise, in Fig. 4.9c, the effect of increasing the number of neurons per hidden layer is shown: the prediction ability of the DNN improves when increasing the hidden layer size until the diminishing or constant trend is encountered. Moreover, the training time for single parallel DNN architecture and the effect of increasing the number of neurons per hidden layer is shown Fig. 4.9d. The lowest possible number of mandatory neurons per layer depends on the structure under examination: the values selected for the following analysis are listed in Table 4.3.

Finally, the correct prediction percentage for the optimized DNN is summarized for the Beneš (8×8, 10×10 and 15×15), Spanke-Beneš (8x8, 10x10) network along with the 12x12 HCROS in Table 4.3. In the Beneš network, we notice an

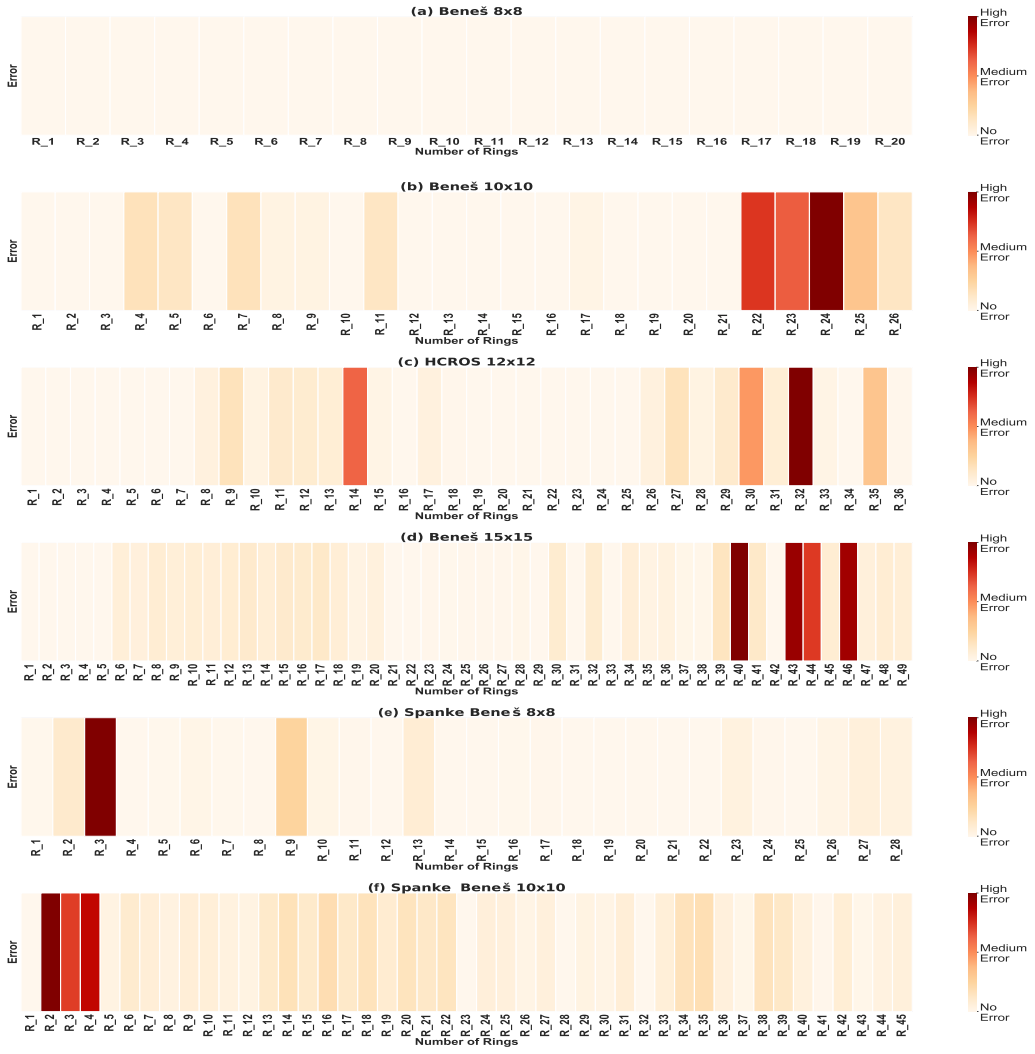


Fig. 4.10 Heatmap showing normalized error in prediction of control signals using DNN.

excellent accuracy level (>96%). However, with reduced prediction efficiency when expanding N: correct predictions reach 100%, 99.72%, and 96.25% for N equal to 8, 10, and 15, respectively. To further validate the results, similar results were obtained based on Spanke-Beneš and HCROS: also, in both of these considered architectures, we observe a high level of accuracy (97.47%, 96.51%, and 97.83% for Spanke-Beneš (8×8, 10×10) and 12×12 HCROS, respectively).

Following the accuracy assessment, the distribution of errors in the predicted states was studied, as depicted in Fig. 4.10. When considering the test set, the amount of measured errors for each switching element control prediction is represented in the color heatmap. A non-uniform distribution is observed, with errors clustered

Table 4.3 Summary of ML prediction results related to control states

Network type **Size ($N \times N$)**	**Beneš** **8x8**	**Beneš** **10x10**	**HCROS** **12x12**	**Beneš** **15x15**	**Spanke-Beneš** **8x8**	**Spanke-Beneš** **10x10**
Neurons per hidden layer	15	35	35	35	35	45
Accuracy (no heuristic)	100%	99.72%	97.83%	96.25%	97.47%	96.51%
Single switch control error	0%	0.28%	2.17%	3.75%	2.53%	3.49%
Multiple switch control error	0%	0%	0%	0%	0%	0%
Accuracy (with heuristic)	100%	100%	100%	100%	100%	100%

on a small number of switching elements. Based on this observation, we analyzed the number of wrong switching elements where the prediction failed. Furthermore, the results in Table 4.3 show that only a single error in one of the switches control is responsible for the incorrect routing, for most of the wrong predictions in all the considered architectures. Observing this phenomenon, a simple heuristic was formulated, which can further improve the DNN prediction performance (check Algo. 3 for detail). The heuristic we suggest requires several device properties such as topological graph ($\mathcal{G}$), M control signals, and N number of inputs/outputs signals. Additionally, the test set ($\mathcal{TS}$) and ML predicted set ($\mathcal{PS}$) are also loaded as an input. The proposed heuristic corrects the single switch errors by switching the state of one element while comparing the output sequence against the target output permutation of wavelengths. This heuristic makes only M iteration, and this number is reasonably small, so it can be considered feasible for real-time operations. Besides this, it is also topologically and technologically agnostic as for all the considered architectures, DNN assisted by heuristic enhances the accuracy up to 100%.

4.5 Machine learning for QoT evaluation of photonic devices

The previous demonstration of the definition of the control states of a PIC $N \times N$ photonic switching system with a completely topology-agnostic *blind* solution exploiting an ML inverse design approach [181] is extended in this section. To complete the switch model, for a full description of the impact on the physical layer, an extra ML network with a direct design method is paired up to predict the QoT degradation due to the switching element. The two ML networks work synergically

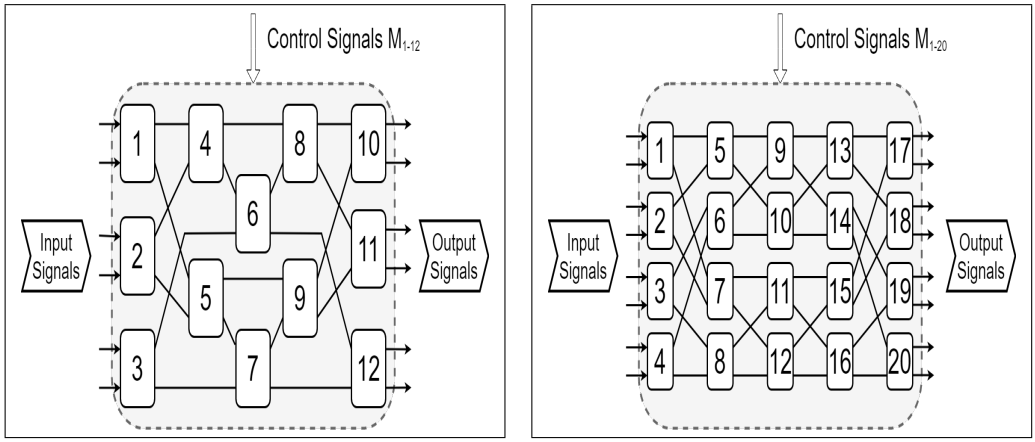
and provide a generic softwarized and QoT aware control and management system for any $N \times N$ optical switch. The proposed abstracted model can be easily extended to measure the impact of $N \times N$ optical switch on the network layer metrics.

Table 4.4 Dataset for QoT Evaluation

Beneš size $N \times N$	**Permutations** $N!$	**OSE Count**	**Combinations** 2^M	**Dataset**	**Train Set**	**Test Set**
6×6	720	12	4096	1000	700	300
8×8	40,320	20	1048576	1000	700	300

4.5.1 Data generation and statistical analysis of OSNR penalty

The training and validation sets have been generated by evaluating the OSNR penalty for random unique state configurations, considering a target BER of $\text{BER}_{\text{th}} = 5 \times 10^{-3}$. OSNR penalties have been obtained by simulating the propagation trough the component under test of the considered signal [182]. The generated datasets contain the penalty at each port of the device for a random set of 1000 realization of control states (see Table 4.4): this simulation has been performed on two different Beneš structures, namely the 6×6 and 8×8 configurations as depicted in Fig. 4.11. The detailed statistical analysis of the acquired OSNR penalty of both the consid-



(a) 6×6 Beneš switch (b) 8×8 Beneš switch

Fig. 4.11 Circuit representation for the two device under test

ered architectures is reported in Fig. 4.12. Observing the OSNR penalty statistics in Fig. 4.12, we can see that the worst-case scenario reaches about 4 dB, as reported with a red dotted line. We can conclude that an OSNR penalty equal to this maximum penalty should be considered for all routing states, as to prevent any switching

configuration from being out-of-service, when no prior knowledge or prediction scheme can be deployed. This value is the same for both the considered topologies. From Fig. 4.12, we can also observe that the average OSNR penalty that can satisfy most of the cases is much lower, 1.93 dB for Beneš 6×6 and 2.12 dB for Beneš 8×8, respectively.

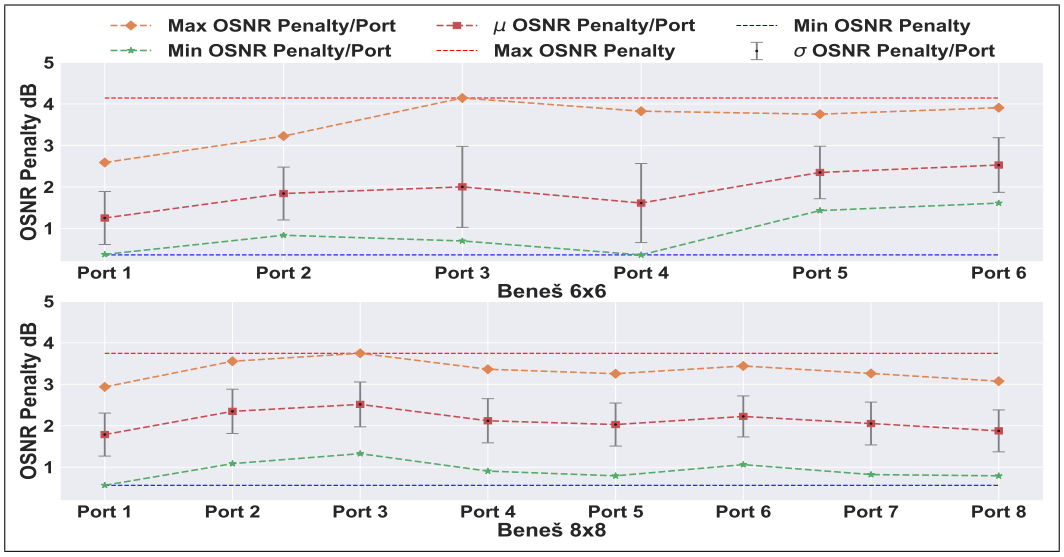


Fig. 4.12 Statistical Analysis of OSNR Penalty

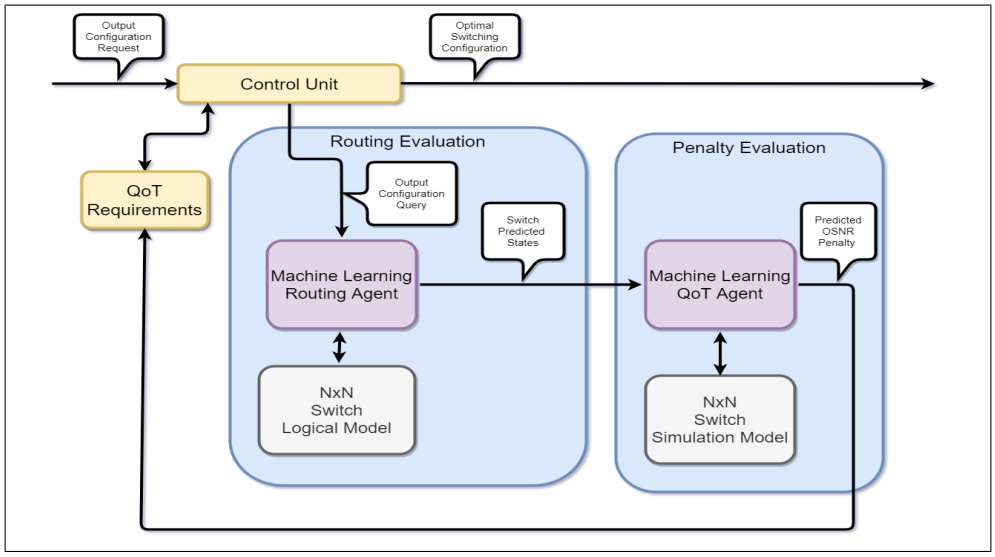


Fig. 4.13 Control Unit block model

4.6 Inverse and direct machine learning modeling for routing and QoT

The proposed photonic switch abstraction model considers two ML networks. The first network, acting as the ML routing agent, is intended to define the control state of the switch through an inverse design approach. In contrast, in the second network, the ML QoT agent takes the first network output and predicts, through a direct design approach, the QoT penalty estimation. This allows the network controller to evaluate the optimal solution of any $N \times N$ photonic switch considered as a black-box shown in Fig. 4.13

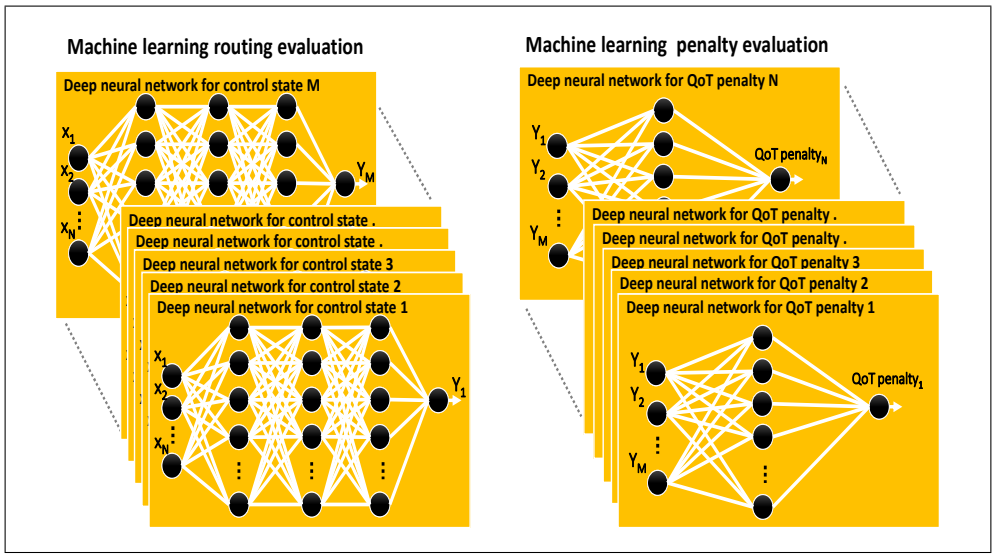


Fig. 4.14 Parallel architecture of a deep neural network

A DNN [167] is considered as a cognition engine for both the proposed ML networks since it is a powerful tool that has shown significant results in numerous frameworks. The proposed DNN is developed by using a higher-level API of the TF platform [183]. To improve the prediction efficiency of both the networks, a parallel architecture for the DNN is proposed, shown in Fig. 4.14. The parallel DNN engines of both networks are trained and tested on a separate subset of the dataset: the conventional rule of 70% and 30% has been chosen to partition the available dataset. In order to avoid over-fitting of the models, the *training steps* is set as the stopping factor and the *MSE* as the loss function as defined in Eq. 4.1 and Eq. 4.3,

respectively for the first and second network.

$$\text{QoT MSE} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{1}{N}\sum_{k=1}^{N}\left(\text{OSNR Penalty}^{p}_{i,\text{k}} - \text{OSNR Penalty}^{a}_{i,\text{k}}\right)^{2}\right) \tag{4.3}$$

where n is the number of test realizations, M is the total number of switching elements in the specific $N\times N$ switching system, while for each tested case i, $\text{Control State}^{p}_{i,\text{m}}$ and $\text{Control State}^{a}_{i,\text{m}}$ are the predicted and actual control states of the m-th switching element of the considered configuration. Similarly, N is the total number of input/output ports of the specific $N\times N$ switching system and $\text{OSNR Penalty}^{p}_{i,\text{k}} - \text{OSNR Penalty}^{a}_{i,\text{k}}$ are the predicted and actual OSNR penalty of the k-th output port of the considered topology.

Furthermore, the DNN engines of both networks are configured by several common parametric values that have been optimized (such as the *training steps*, set to 1000), loaded with the *ADAGRAD* Keras optimizer, with *learning rate* set to 10^{-2} and L_1 regularization set to 10^{-3}. Moreover, several non-linear activation functions such as *Relu, Tanh, sigmoid* have been tested during the model building. After testing, *Relu* has been selected to implement DNN as it outperforms the others in terms of prediction and computational load [65].

4.6.1 Machine learning routing and QoT agent

The ML routing agent considers various permutations of the input signals (λ_1, λ_2, λ_3....λ_n) at the output ports of the switch as features while it exploits its M control states as labels. The inverse model has been configured on considerable numbers of *hidden-layers* and neurons to achieve the best trade-off between precision and computational time. Although an increase in the number of layers and neurons improves the accuracy of the DNN up to a certain extent, a further increase in these values introduces over-fitting and increases the computational time. Following this trade-off assessment, we opted upon a DNN with *three hidden-layers* with 10 and 15 cognitive neurons for each hidden layer optimized for Beneš 6×6, and Beneš 8×8, respectively.

The ML QoT agent considers the output of the first ML network (ML routing agent), i.e., the M controls states. At the same time, the utilized response variable is the OSNR penalty of the specific output port of the $N\times N$ switching system. The

direct model has also been configured on considerable numbers of *hidden-layers* and neurons to achieve the best trade-off between precision and computational time. Following this trade-off assessment, we opted upon a DNN with *one hidden-layers* with 11 and 18 cognitive neurons for Beneš 6×6, and Beneš 8×8, respectively.

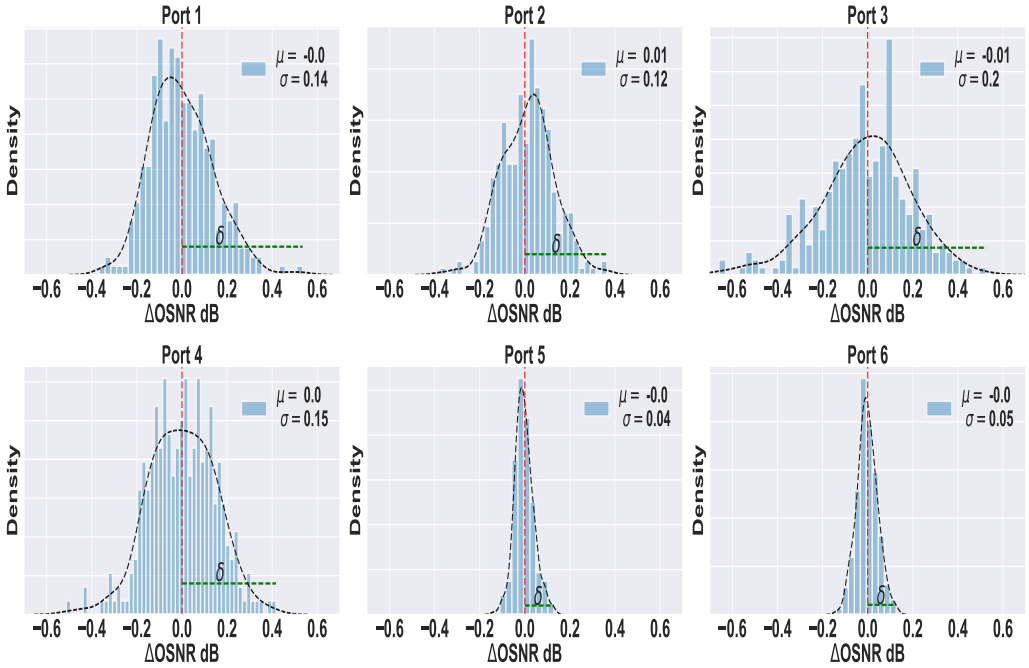


Fig. 4.15 Probability density functions of ΔOSNR for each port of the 6x6 Beneš switch.

The performance of the proposed ML modules are analyzed using a two-steps approach: first, predict the switch control states, and then exploit these result to obtain the QoT impairments in terms of $\text{OSNR Penalty}_{i,\text{k}}$ for each port k of the considered Beneš network.

The ML routing agent gives an excellent level of accuracy in terms of predicting the control states. The agent gives 100% accuracy for both of the considered Beneš topologies N equal to 6 and 8. The scalability and detailed analysis of ML routing agent are reported in detail in [181].

For the ML QoT agent, the predicted control states are given as an input to get the QoT penalty. The metric used to evaluate the accuracy of the QoT module is defined as:

$$\Delta OSNR_{i,\text{k}} = \text{OSNR Penalty}^{a}_{i,\text{k}} - \text{OSNR Penalty}^{p}_{i,\text{k}} \tag{4.4}$$

where all parameters have same meaning as described for Eq. 4.3.

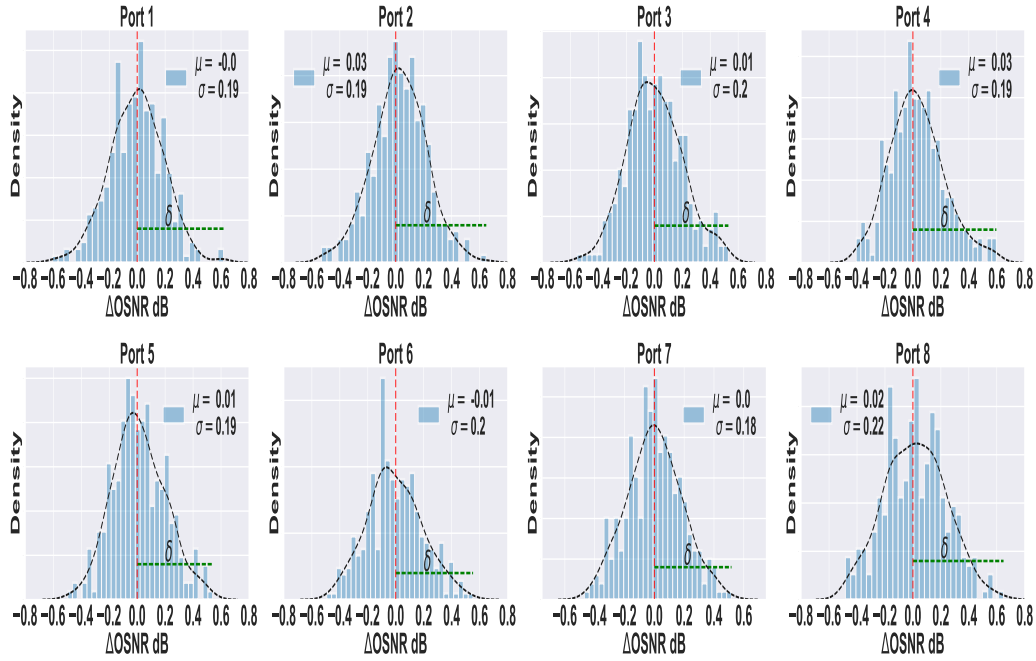


Fig. 4.16 Probability density functions of ΔOSNR for each port of the 8x8 Beneš switch.

The modules scalability and reliability are cross-verified by and simulating their deployment on two different Beneš sizes, namely the 6×6 and the 8×8 configurations. The distribution of ΔOSNRs at the ports of the 6×6 Beneš are shown in Fig. 4.15, along with their μ and σ statistics. Similarly, the distribution of all the eight ΔOSNRs of the 8×8 Beneš are reported in Fig. 4.16.

In Fig. 4.15 and Fig. 4.16, all the distribution of ΔOSNRs in both the cases are split by the dotted red line (ΔOSNR = 0) into two slices. The slice where ΔOSNRs $\leq$ 0 is not critical as the OSNR Penalty$^{a}_{i,\mathrm{k}} \leq$ OSNR Penalty$^{p}_{i,\mathrm{k}}$ so, in this case we only waste some capacity but the system will never turns into OOS. In contrast the section where ΔOSNRs > 0 is the critical one as OSNR Penalty$^{a}_{i,\mathrm{k}} >$ OSNR Penalty$^{p}_{i,\mathrm{k}}$. In this case, it is necessary to deploy some margin on top of the ML prediction to keep the system working all the time. The maximum required margins (δ_k) for this case where ΔOSNRs > 0 are shown as a green dotted line for each port k of Beneš 6×6 and Beneš 8×8, respectively.

Examining the required margin, we can observe the high level of accuracy achieved by the ML QoT agent. In the 6×6 Beneš, the worst-case prediction performance is observed on port 1; the δ_1 is less than 0.6 dB. For the larger 8×8 Beneš, the worst-case prediction is observed on port 8; the δ_8 is less than 0.65 dB.

With the availability of such accurate prediction, we can envision that in practical applications the OSNR penalty margin on top of the ML prediction can be reduced to 0.6 dB and 0.65 dB for Beneš 6×6 and Beneš 8×8, respectively.

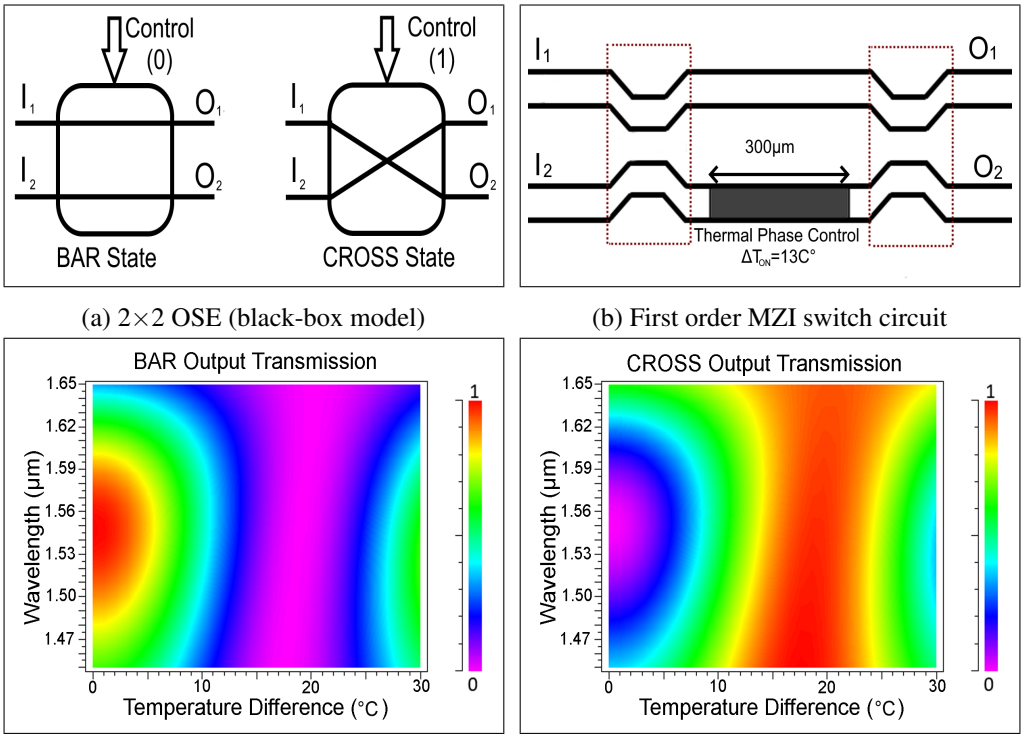
4.7 Machine learning assisted control of wideband switching fabrics

The majority of the present-day deployed optical transport networks exploits WDM across a spectral window of ≈ 4.8 THz in the C-band, with a transmission capacity of up to 38.4 Tb/s/fiber [184]. Additional expansion in the network capacity entails the implementation of different solutions, such as exploiting the residual capacity of already installed infrastructure or deploying new infrastructure. The initial solution, exploiting the residual capacity of already installed infrastructure, is more important for the network operator from a techno-economic perspective. In this regard, technology such as BDM is emerged as a promising solution to expand the capacity of existing WDM optical systems over the entire low-loss spectrum of optical fibers (e.g., ≈ 54 THz in ITU G.652.D fiber) [185].

4.7.1 Ultra-wideband switching system

The device under analysis consists of an electronically controlled integrated transparent photonic switch, able to perform the routing operation without electro-optical conversion of the transmitted signals. The two main characteristics of the system are related to the frequency range of operation, allowing switching in the spectral range covering S+C+L bands, as well as the logical routing requirement, as every permutation of the input signal must be achievable at the egress stage of the device, referred as non-blocking switching.

Different solutions have been described in the literature, with $N\times N$ multistage switching networks being one of the most widespread implementations. In this class of devices, the routing operation is achieved by cascading various stages of elementary 2×2 switches, referred to as OSE, arranged in different topologies depending on the required properties of the routing operation. Each OSE of the network can be controlled independently through an electrical signal. In this work,



(a) 2×2 OSE (black-box model)

(b) First order MZI switch circuit

(c) BAR-state transmission [$I_1 \rightarrow O_1$]

(d) CROSS-state transmission [$I_1 \rightarrow O_2$]

Fig. 4.17 (a) Optical switching element (OSE) black-box model. (b) MZI switching circuit. (c) and (d) Dependence of transmission properties in BAR and CROSS state on wavelength and temperature.

we apply the proposed approach to an 8×8 switching device, with MZI-based OSEs, analyzed in Section 4.7.1.1, interconnected through the Beneš network topology, described in Section 4.1. The switching network size $N = 8$ has been chosen as a trade-off between realistic implementation sizes for photonic integrated circuits, circuit complexity and dataset size. The chosen size acts as a reasonable simulation test-bed to verify the proposed control scheme and abstraction, while providing a large enough component cascade to highlight the physical devices behavior.

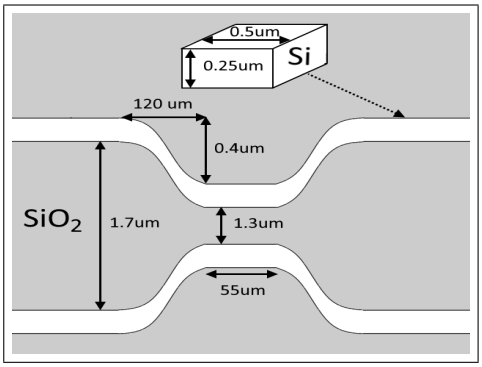
4.7.1.1 Optical switching element

The OSE is the fundamental block required for the switching action, introducing limitations on the operating frequencies and imposing some QoT degradation. At the logical level, the OSE 2×2 cross-bar switch can be modelled as a black-box (see Fig. 4.17a) with two available routing states: the BAR configuration

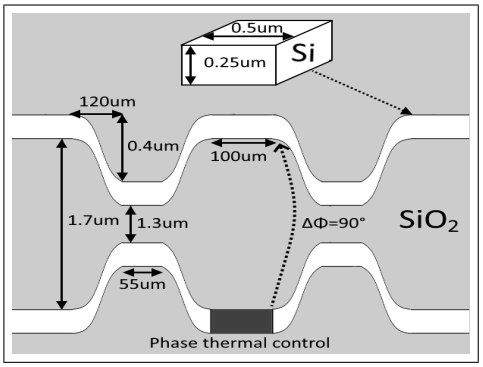
($[I_1, I_2] \rightarrow [O_1, O_2]$) and the CROSS configuration ($[I_1, I_2] \rightarrow [O_2, O_1]$) which can be appropriately toggled by a given binary control signal. The OSE can be implemented with two main solutions described in the literature: the MRR filter and the MZI. Due to the bandwidth limitations of the MRR solutions, we propose in this paper a device based on the MZI principle.

The most straight forward MZI device is structured as shown in Fig. 4.17b: the signal is divided into the two waveguides by the first 3 dB coupling section and recombined in the egress 3 dB coupler, with the thermally-controlled phase shift region acting as the control section for the routing state. The routing state is controlled electrically by increasing the temperature of the phase control waveguide in the MZI arms. This increase in temperature introduces a phase shift in the propagating signal, changing the output recombination waveguide in the egress coupler. The signal transmission is depicted in Fig. 4.17(c) and Fig. 4.17(d) as a function of the signal wavelength as well as the temperature shift between the MZI arms. In the OFF state ($\Delta T = 0\,^{\circ}\mathrm{C}$) the bandwidth limitation of the device is clear, with the range of operation covering roughly half of the S+C+L band. The bandwidth limitation is due to the 3 dB coupling regions where phase velocity dispersion of the physical waveguides causes asymmetry in the signal propagation, with uneven power splitting and recombination, leading to significant crosstalk with the incorrect output port.

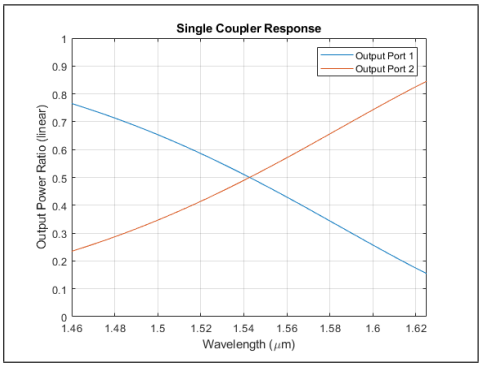
- **Higher order coupling regions:**
  The critical component for achieving the UWB range of operation is the coupler region, required before and after the thermal phase control section. While the coupler has a 3 dB power ratio for the centre design frequency, the waveguide dispersion causes increasing asymmetry as the signal frequency moves away from the center point. One of the simplest solutions to compensate for this effect is cascading two identical couplers while introducing a constant phase shift between the two waveguides ($\Delta\phi = 90^{\circ}$), as shown in Fig. 4.18. This solution reduces the dispersion effect on the power ratio, leading to a larger and flatter bandwidth near the design frequency while also reducing the overall asymmetry at the limits of the chosen bandwidth, depicted in Fig. 4.18d. More advanced solutions, like a complex waveguide, tapered structures or advanced 3D structure [186] can still enlarge the bandwidth of the 3 dB coupling region. For the intended applications of a multi-stage switching structure, the



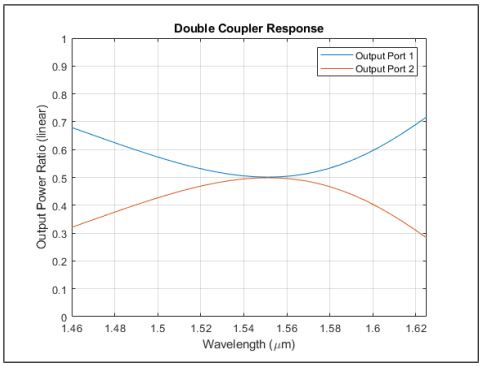
(a) Single coupler circuit representation



(b) Double coupler circuit representation



(c) Single coupler frequency response



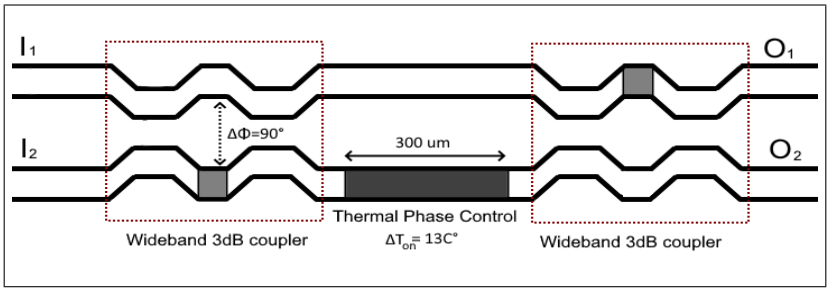
(d) Double coupler frequency response

Fig. 4.18 (a)-(b) MZI switch structures: single coupler vs. double coupler. (c)-(d) Corresponding transmission bandwidth properties.

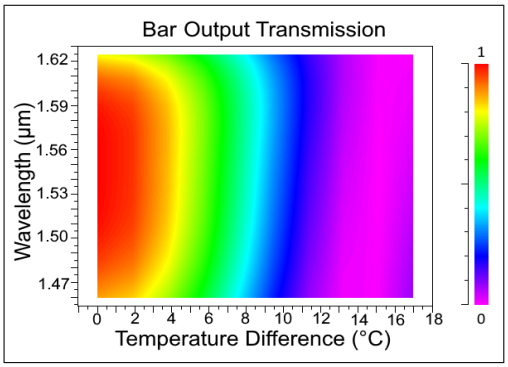
rapid increase in circuit complexity and production requirements may become prohibitive as the scale of the overall network increases, which leads to a trade-off between the cost and effectiveness of the solution. The analyzed device, implemented through the second-order coupling structure, is depicted in Fig. 4.19a: the bandwidth of operation covers the target transmission windows, with increases in crosstalk and penalty observed only at the edges of the operating region, as shown for both routing states in Fig. 4.19b- Fig. 4.19c.

4.7.2 Simulation environment and dataset generation

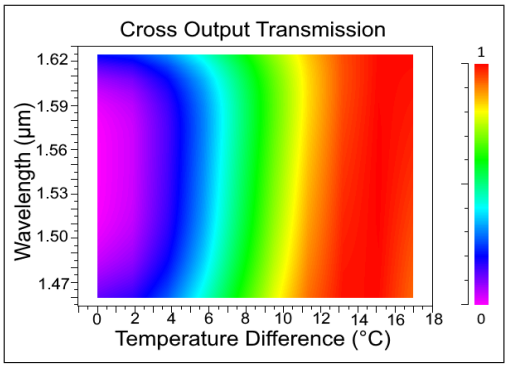
After defining the fundamental 2×2 OSE, any generic $N \times N$ circuit can be modeled following the topology of choice, for example, the Beneš network. The Beneš architecture is modeled under two different levels of abstraction to characterize



(a) Second order coupling MZI device tested in this study.



(b) BAR-state transmission [$I_1 \rightarrow O_1$]



(c) CROSS-state transmission [$I_1 \rightarrow O_2$]

Fig. 4.19 Transmission dependence on wavelength and thermal tunability for the second order coupling MZI device tested in the paper.

the dependence on the control signals of both the routing behavior and the impact on QoT of the switching operation.

4.7.2.1 Routing model

Given the black-box abstraction of the 2×2 cross-bar OSEs, the routing problem can be solved on a simplified version of the circuit, taking only into account the logical link between input-output ports as a function of the binary control state of each fundamental switch. To this end, a virtual topological structure was generated in MATLAB®, in order to analyze the routing and then to evaluate the logical output for the QoT transmission-level simulation. Given the simple recursive structure of the network, coupled with the non-polynomial increase in the solution space ($N_{\text{conf}} = \mathcal{O}(2^{N \log N})$, $N_{\text{out}} = \mathcal{O}(N!)$), brute-force solution together with look-up tables are not a scalable method to obtain the states configurations for the target output request. This introduces the need for a scalable deterministic algorithm to tackle the problem complexity and provide the equivalent paths routing the same output

permutation. While it is fundamental to be able to generate a single routing solution for a target output permutation, in order to minimize the penalties in a device-agnostic scenario a more general algorithm is needed to evaluate all equivalent routing solutions for each required signal output permutation. The device-agnostic scenario is introduced to generalize the analysis without the need of assuming the QoT behavior to the physical and device-level structure: a simpler approach to the optimization of the QoT could be to minimize the number of interconnecting crossings encountered by each signal, as these elements are typically the leading cause of signal attenuation. However, this relies on a device assumption which could not always be accurate, so to avoid the issue the problem is split into two main sections, under a "divide and rule" paradigm: the routing model is tasked with generating all equivalent routings for the target signal output, without introducing assumptions on the underlying transmission penalty, while the ML agent proposed in the later sections handles the QoT optimizations, selecting between the solution space the best-predicted solution.

The proposed solution represents a generalization of the matrix-based algorithm described in [187]. Having defined an $N \times N$ Beneš, with number of switches per stage $N_{\text{sw/st}} = \frac{N}{2}$ and number of stages $N_{\text{st}} = \log_2 N$, the proposed algorithm is divided in the following steps (Algorithm 1-Algorithm 2):

- For each layer of the network up to the half-point stage, generate two empty matrices $\mathcal{M}, \mathcal{T} \in \mathcal{R}^{\frac{N}{2} \times \frac{N}{2}}$, representing respectively the control states of the OSEs in the layer and the rearranged signal order after the layer.
- By comparing the input signals order of the ingress layer with respect to the output signals order of the egress layer, for every signal map, the relation between input switch and target output switch. The ingress and egress layers are symmetrical with respect to the middle stage $N_{\text{middle}} = \frac{N_{\text{st}}}{2}$ (ingress: layer (i), egress: layer $(N_{\text{st}} - i)$, for $i \in [1 : N_{\text{middle}}]$)
- Fill the matrix $\mathcal{M}$ with $[0, 1]$ using the input-output switch relationship to select the row-column pair respectively. The matrix $\mathcal{T}$ contains the label of the signal corresponding to that input-output switch pair.
- Once the matrix for the layer is compiled, verify that no repetitions occur both row-wise and column-wise. Only one instance of "0" and "1" can occur in any

Algorithm 1 Beneš routing algorithm

Require: *Number of input and output channels* $\mathcal{N}$*, Input signals labels* $\mathcal{I}_{[1:\mathcal{N}]}$*, Output signals labels* $\mathcal{O}_{[1:\mathcal{N}]}$

Ensure: *A control state, chosen randomly among all the control states giving the requested target output permutation*

Number of switch per stage $\mathcal{N}_{\text{sw/st}} = \frac{N}{2}$
Number of stages $\mathcal{N}_{st} = \log_2 N$
for layer index $l = 1$ **to** $\frac{\mathcal{N}_{st}}{2} - 1$ **do**
    Initialize to -1 the routing matrix $M_l \in \mathcal{R}^{\frac{N}{2} \times \frac{N}{2}}$
    Initialize to 0 the label matrix $T_l \in \mathcal{R}^{\frac{N}{2} \times \frac{N}{2}}$
    for label index $j = 1$ **to** $\mathcal{N}$ **do**
        $SW_I \leftarrow \left\lceil \frac{j}{2} \right\rceil$
        $SW_O \leftarrow \left\lceil \frac{k}{2} \right\rceil$ **with** k such that $\mathcal{O}_k = \mathcal{I}_j$
        if $M_l(SW_I, SW_O)$ **is** -1 **then**
            if column SW_O of M_l contains a 0 **then**
                $M_l(SW_I, SW_O) \leftarrow 1$ %%% 1 i.e. bottom network routing
            else if column SW_O of M_l contains a 1 **then**
                $M_l(SW_I, SW_O) \leftarrow 0$ %%% 0 i.e. top network routing
            else
                set randomly $M_l(SW_I, SW_O)$ to 0 or 1
            end if
            $T_l(SW_I, SW_O) \leftarrow \mathcal{I}_j$
        else if $M_l(SW_I, SW_O)$ **is not** -1 **then**
            $M_l(SW_I, SW_O) \leftarrow 2$ %%% 2 i.e. same input-output couple request
            $T_l(SW_I, 1) \leftarrow T_l(SW_I, SW_O)$
            $T_l(SW_I, 2) \leftarrow \mathcal{I}_j$
        end if
    end for
    check for conflict $\rightarrow$ **Algorithm 2**
    update $\mathcal{I}_j$, $\mathcal{O}_j$ for the evaluated routing
end for

given row or column. If repetitions occur, flip the element column-wise until the conditions are solved.

- Iterate for all layers $i \in [1 : N_{\text{middle}}]$).

In the described algorithm the "0" and "1" flags of the $\mathcal{M}$ matrices correspond to the propagation direction of the signal in each switching element, relative to the following stages: considering the recursive structure of the Beneš topology, as well as its symmetry, at every stage, two equivalent paths can be found in the respective top and bottom following sub-network. Two additional flags values are used in the proposed algorithm: every matrix cell is initially set to "-1" to indicate non-allocated requests or empty cells. An additional flag is required in the routing matrix in order to account for equivalent routings in some specific cases: while typically the input

Algorithm 2 Routing conflict algorithm

Require: *Routing matrix M*
Ensure: *Balanced conflict-avoiding routing*
conflict=**false**
Conflicts=empty list
for row index $r = 1$ **to** N **do**
    R=row r of M
    if R contains two 0s or two 1s **then**
        conflict=**true**
        append r to Conflicts
    end if
end for
if conflict **then**
    r^*= randomly selected element **in** Conflicts
    R^*=row r^* of M
    c^* = randomly selected position of any element of R^* equal to 0 or 1
    C^*=column c^* of M
    exchange 0 and 1 in C^*
    loop to 1
else
    return $\rightarrow$ **Algorithm 1** (update matrix M)
end if

signals of an ingress switch must be routed to different egress switches when both input signals are targeting the same output switch, only one single cell of the routing matrix can be targeted: to this end, the flag "2" represents the path equivalence between the top and bottom network, with the implied value of both ("1","0") and ("0","1").

Once the procedure is completed, the state of the switches can be obtained by comparing the order of the signals of each layer, taking into account the interconnects and the top/down direction provided by the compiled $\mathcal{M}_i$ matrices. With a slight modification to the presented algorithm, the evaluation of all equivalent paths in terms of permutation of the output signals becomes trivial: once the output permutation is set, each valid matrix $\mathcal{M}_i$ represents a different equivalent routing possibility. For every routing of the previous layer, the process is iterated, generating a recursive exploration of all switching states for the required output. Using the proposed algorithm, the control unit can generate different solutions depending on the required task: if all equivalent routing solutions are evaluated, the proposed ML agent can optimize the QoT, finally choosing the path with minimum transmission penalty. Suppose a simpler control unit is required; the algorithm can provide a single control

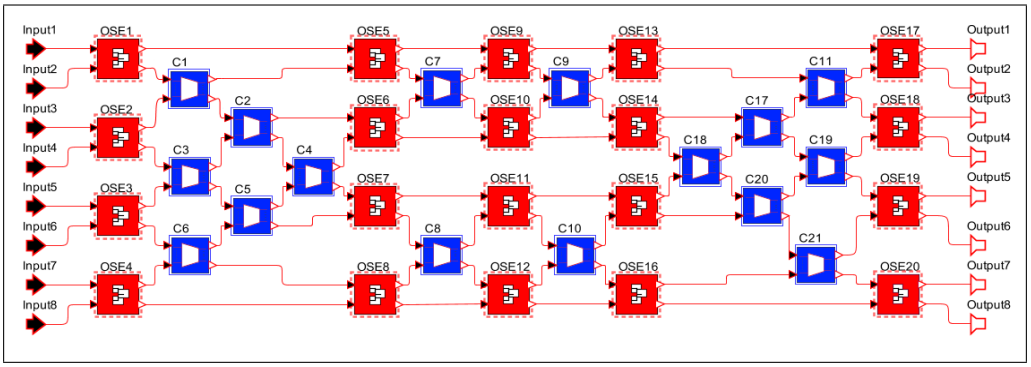
configuration for the device, generating one random routing compatible with the required signal output permutation without exploring all equivalent paths.

4.7.2.2 Transmission model

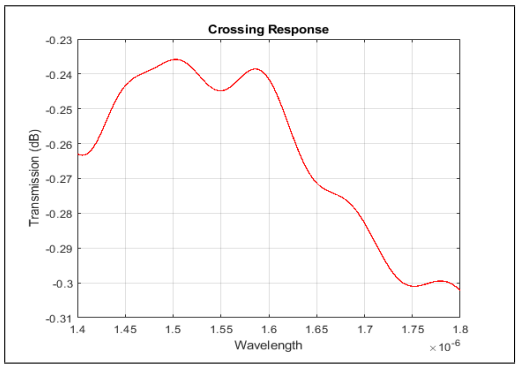
To evaluate the impact of the switching fabric on the QoT, numerical simulations have been first carried out in the Synopsys OptSim^{TM} Photonic Circuit simulation environment [188], testing an 8×8 Beneš switch base on an OSE implemented with the second order coupling MZI previously described. Due to the relative low-loss flat-band behavior of the OSE, the critical components in the device, especially concerning routing optimization, are the waveguide crossings, which introduce path-dependent losses and attenuation in the propagating signals. It must also be remarked that for strict-sense Beneš structures ($N = 2^x, x \in \mathcal{N}$) the number of switches encountered by each signal is equal, independently from the OSE control signals, as shown in Fig. 4.20a, highlighting the critical task in characterizing the control states dependent QoT impairments due to the stages interconnects.

The designed waveguide crossing introduces an average 0.2 dB–0.3 dB loss for each instance, with a small spectral variance, as depicted in Fig. 4.20b. While the crossings have been accounted for the penalty evaluation, the interconnect waveguides and bent sections have not, due to their generally negligible effect in a properly designed layout. The general schematic of the simulated setup is depicted in Fig. 4.20c. We assumed eight input signals spaced $\Delta f = 100\,\text{GHz}$ with a central frequency of $f_c = 193\,\text{THz}$. The simulated signals consisted of PM-16-QAM modulated streams at $B_r = 60\,\text{GBaud}$, which are then demodulated at the receiver side, extracting BER as a function of the OSNR. These measurements are then expressed as QoT Penalty (in decibel), comparing to the trend of the back-to-back TX/RX system evaluated without the switching fabric. Due to the previously discussed non-polynomial increase of the solution space, the characterization of the full system through a look-up table solution is not feasible, especially at the transmission level, due to the high computational costs of such simulations. In order to train the proposed ML algorithm, it is necessary to build a dataset of simulated configurations, measuring the QoT Penalty for a random sub-set of control signals.

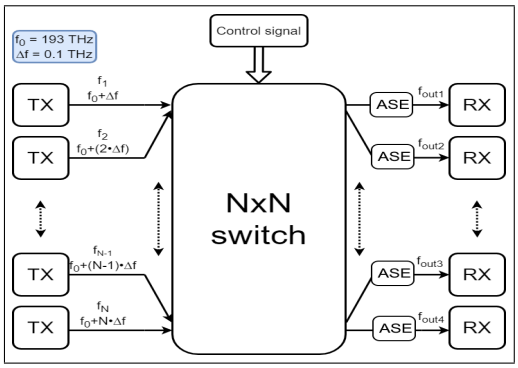
The simulation dataset has been generated for $N_{\text{sim}} = 5000$ random control configurations, allowing equivalent paths (output permutation) but enforcing individual control states to avoid erroneous training by repeating the same OSE states.



(a) 8×8 Beneš switch schematic in OptSim Photonic Circuit. Crossings are indicated by blue blocks while OSEs are shown as red blocks.



(b) Response of the waveguide crossing



(c) Transmission environment under simulation

Fig. 4.20 Simulation model schematic and characteristics

The general distribution of the OSNR Penalties for the simulated dataset is shown in Fig. 4.21: as expected, the distribution has a relatively uniform average value of $\mu = 2\,\text{dB}$ for every output port, with a comparable standard deviation. To characterize the device in SDN controlled environment, it is important to highlight the maximum value of the penalty: $\Delta\text{OSNR}_{\text{max}} \approx 3.1\,\text{dB}$. Without a control unit capable of a reliable prediction of the expected penalty in real-time, the impact of switching on QoT must always be over-estimated to this maximum value, which represents an infrequent worst-case assumption. To this end, the ML agent allows more flexible control of the device, highlighting the cases where a higher transmission rate can be applied due to a lower penalty. Furthermore, in Fig. 4.21 every data-point corresponds to one of the different equivalent solutions, highlighting the average penalty for all the output ports, as well as the minimum and maximum values. It is clear how a real-time control strategy can be employed to optimize the performance of such a device. At the same time, the average port penalties are identical between

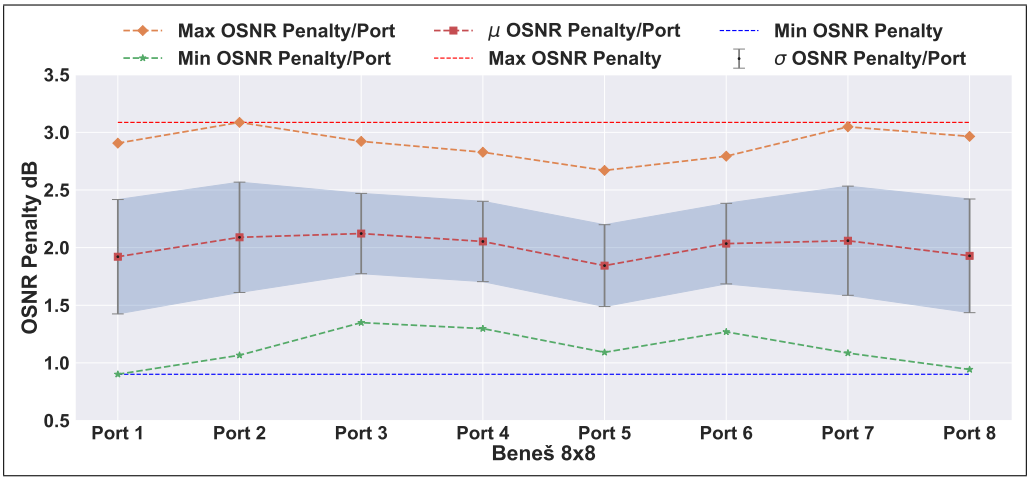


Fig. 4.21 Statistical analysis of OSNR Penalties for each output port.

configurations; the lower variance solutions offer a better alternative, as the QoT is more uniform between all the output ports of the configuration.

4.7.3 Detail analysis and case study

This section demonstrates the accuracy of proposed ML modules in delivering QoT impairments predictions for UWB photonics switching architectures. The ML module exploits the deterministic switch control states to obtain the QoT impairments in terms of OSNR Penalty$_{i,k}$ for each port k of the proposed UWB Beneš switch. In addition to this, a complete case study is also analyzed to reveal the effectiveness of the proposed ML-based QoT Penalty estimation model for the photonic switching system.

The proposed ML cognitive engine manipulates the deterministic control states as input and exploits the QoT Penalty as an output (detail is given Section 4.6.1). The metric utilized to assess the accuracy of the ML model is defined by Eq. 4.4. The reliability of the proposed ML-based QoT model is verified by analyzing its performance at each port of the proposed 8×8 Beneš switch. The distribution of ΔOSNRs of all the ports of the 8×8 Beneš are shown in Fig. 4.22, along with their μ and σ statistics.

In Fig. 4.22, all the distributions of ΔOSNRs are divided into two parts using the red dotted line (ΔOSNR = 0). The area of distribution where ΔOSNRs ≤ 0 is not severe as OSNR Penalty$^{a}_{i,k} \leq$ OSNR Penalty$^{p}_{i,k}$ so, in this scenario the only

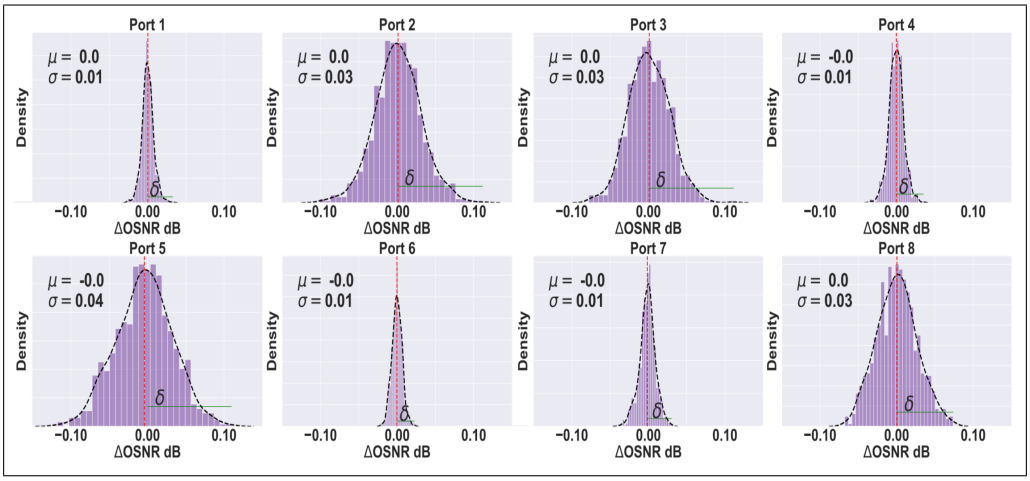


Fig. 4.22 Probability density functions of ΔOSNR for each port of the 8x8 Beneš switch. Average values μ and variances σ indicated for the individual cases are expressed in decibel.

deterioration is the some waste of capacity but the system will never turn into OOS. In contrast the area where ΔOSNRs > 0 is the more severe one as OSNR Penalty$^{a}_{i,k} >$ OSNR Penalty$^{p}_{i,k}$. In this case, it is required to deploy some margin on top of the ML prediction to keep the system working all the time. The maximum required margins (δ_k) for this case where ΔOSNRs > 0 are shown as a green line for each port k of the 8×8 Beneš.

Inspecting the required margin, we observe the high level of accuracy achieved operating ML model for QoT impairments estimation. The proposed 8×8 Beneš, the worst-case prediction performance is observed on port 5; the δ_5 is less than 0.12 dB. With the availability of such accurate prediction, we can envision that in practical applications, the OSNR Penalty margin on top of the ML prediction can be reduced to 0.12 dB for Beneš 8x8. Furthermore, the prediction asymmetry between the different port of the device, is due to the intrinsic randomness and limited size of the provided dataset, leading to better training for the prediction of certain paths. For larger dataset a more statistically relevant margin, i.e, 2σ or 3σ can be considered. Under the envisioned case-study, a drastically smaller dataset has been provided by choice to the ML agent with respect to the complete device configuration set. Even under this limited training scenario, the asymmetry between the port predictions is still marginal with respect to the QoT optimization available through this method deployment. The effectiveness of the proposed ML-based QoT impairments estimation model is further demonstrated by considering

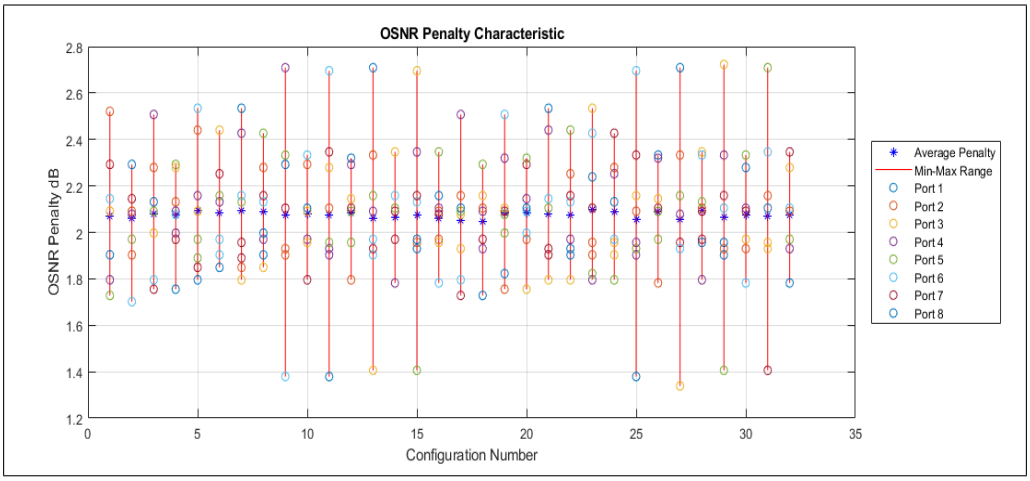


Fig. 4.23 OSNR Penalty distribution for 32 nominally equivalent control states generating the output pattern [7, 6, 3, 8, 5, 4, 1, 2]. A label from 1 to 32 has been assigned to each control state according to the order it is generated by the proposed algorithm.

the optimality routing issue: the ML agent can be used to optimize the routing solution in conjunction with the previously described routing algorithm. Taking as an example a target output request such as $[1, 2, 3, 4, 5, 6, 7, 8] \rightarrow [7, 6, 3, 8, 5, 4, 1, 2]$, we observe that 32 different combinations of the control states exist leading to the desired output pattern. The designed routing algorithm is able to evaluate all these nominally equivalent routing solutions, which have been tested in order to characterize their penalty and statistical distribution, as shown in Fig. 4.23. The average penalty for every equivalent configuration is reasonably similar, while the main difference is found between the standard deviation between the penalty of each port. The ML agent could provide real-time control optimization for this application, minimizing the overall penalty and avoiding high deviation solutions. This target goal allows for a similar penalty factor between all the output signals, minimizing the overall deviation, although different criteria could provide alternative solutions depending on the overall control goal. The choice of the *best* control state depends on the selected metric: considering the results introduced in Fig. 4.23, configuration number 18 provides the minimal deviation between the alternative routings, while solution number 27 could be selected if only the minimum penalty is considered as the optical metric.

Chapter 5

Conclusions and Future Work

In this thesis, data-driven techniques for multi-layer applications of optical networks are discussed. Novel applications in terms of machine learning-assisted optical network management are demonstrated. A fully softwarized management model for integrated photonics-based components is also suggested.

5.1 Summary

Optical communication is a complex system affected by real-world phenomena such as fiber mechanical stresses, temperature, losses varying in time, fiber cut, amplifier aging, etc. The data acquired from a real field practically accumulate all the phenomena blindly, and it is hard to isolate a single phenomenon for analysis. Moreover, it's challenging to find real field network data. In this context, the analysis performed in this thesis is entirely based upon on synthetic dataset, as it's good to start the investigation with a synthetic dataset that potentially enables us to isolate a single phenomenon and analyze its effect on the overall system.

The thesis begins with a broad overview of optical communication and various progressing communication technologies that are industrialized to handle the fast growth in global IP traffic. These advanced communication technologies, such as coherent optical transmission, elastic optical networks, and the opening of software-defined optical networks, introduce many tunable parameters, making the design and operation of optical networks more complex.

In this novel context, exact system modeling using closed-form formulations is very challenging, and typically, a "margin" is deployed while adopting the analytical models. The deployment of specific margins leads to the under-utilization of resources and eventually enlarges the network operational cost. To cater to this limitation of analytical models, the telecom industry is strongly pushing the optical community to move towards intelligent optical networks that can perform autonomous and flexible network management.

Following these requirements, the introduction to modern open and disaggregated optical networks were given at the thesis's start. The state-of-the-art software-defined optical network architecture was described, which decoupled the control and data planes. The basic framework of the software-defined optical network was presented, which is an enabling technology towards intelligent optical networks, and is composed of three principal planes: the data, control, and application plane. The assisting cognition module is logically centralized and is part of the control plane.

Later in Chapter 3, a novel framework of machine learning-assisted QoT-E in optical networks was presented. The machine learning model's training on the GSNR response to specific spectral load configurations of the already deployed *in-service* network, and consider its realization to predict the QoT of an *agnostic/un-used* network. The dataset is retrieved from QoT responses against random spectral loads of *in-service* network. This data is generated during the operative phase of the *in-service* network by measuring the optical line system response in terms of GSNR for various spectral load configurations. This novel framework provides an ideal playground to apply machine learning. A machine learning model using a training dataset composed of spectral load realizations of an *in-service* network yields an accurate QoT-E for each newly generated spectral load realization of *agnostic/un-used* network.

Then, in the final Chapter 4, a framework of machine learning-assisted photonic devices management was presented. A novel architecture was proposed in the context of software-defined optical network in which a generic topology-agnostic *blind* model was exploiting an machine learning inverse design approach to obtain the softwarized control of any $N \times N$ photonic switching system. Moreover, a direct design method was presented to predict the QoT degradation due to the switching elements configurations.

5.2 Next Steps

There are numerous promising future advancements for the subject presented in this thesis. Starting from the software-defined optical network, which offers an implementation of each network element within a virtualized environment, so permitting a disaggregated approach to the network, enabling openness and virtual network slicing. One of the potential directions of future work may be to push the vendors industry to practically integrate the APIs of cognitive units within the software-defined optical network and enable the network operators to automatize their network functionality.

Concerning QoT-E, an exciting research opportunity is to cross verify the simulation results obtained by a synthetic dataset with real field investigations of the network. An additional remarkable set of analyses can also be done purely at the network layer, especially exploiting the machine learning paradigm for solving routing and wavelength assignment (RWA) and routing and spectrum assignment (RSA) problems efficiently, which are typically a non-deterministic polynomial-time hardness (NP-hard) problem.

One more significant focus that could be of possible interest for future work is the use of machine learning assistance in the design of photonic integrated circuits. The photonic integrated circuit based components typically need complex computational simulation in order to retrieve their optical responses correctly. The usual period to design photonic integrated circuit based devices sometimes surpasses the duration of fabrication and testing of these devices. To deal with these challenges, a new design paradigm based on machine learning assistance is highly required for integrated photonics that can address the particular demands of photonic circuit designers

The other exciting field of research is related to the extension of the analysis to find the performance of any $N \times N$ optical switching system in terms of transmission penalties predicted using machine learning on the network layer metrics. Finally, the additional exciting area is developing an machine learning assisted control model for active devices such as lasers, etc. Generally, the auto-tuning and feedback-empowered machine learning assistance may perhaps characterize a new era in the science of active devices.

Appendix

Algorithm 1

Algorithm 3 Heuristic to correct single-ring errors

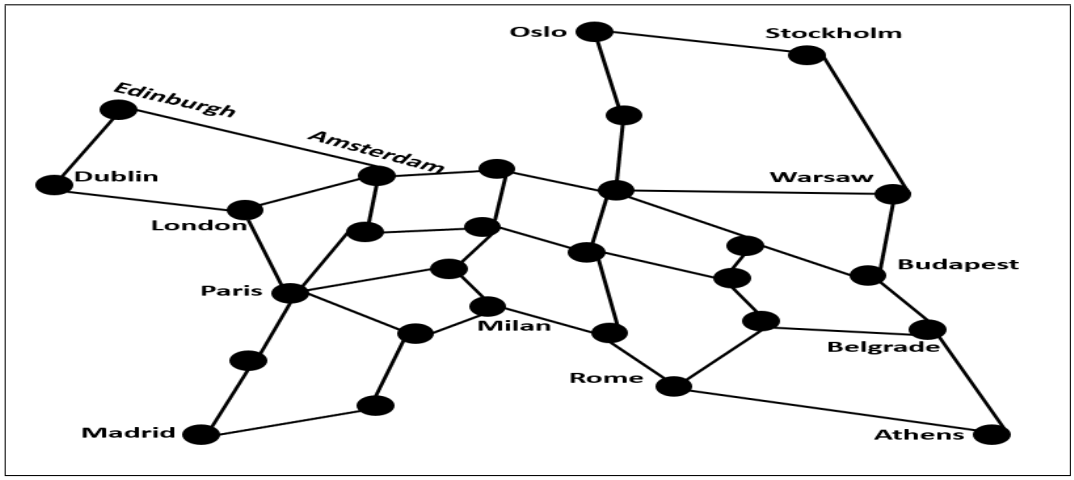
Require: *Topology Graph* $\mathcal{G}$*, Control Signals M, Number of Inputs/outputs N, Test set* $\mathcal{TS}$*, ML Predicted set* $\mathcal{PS}$

Ensure: *Control Signal Correction*

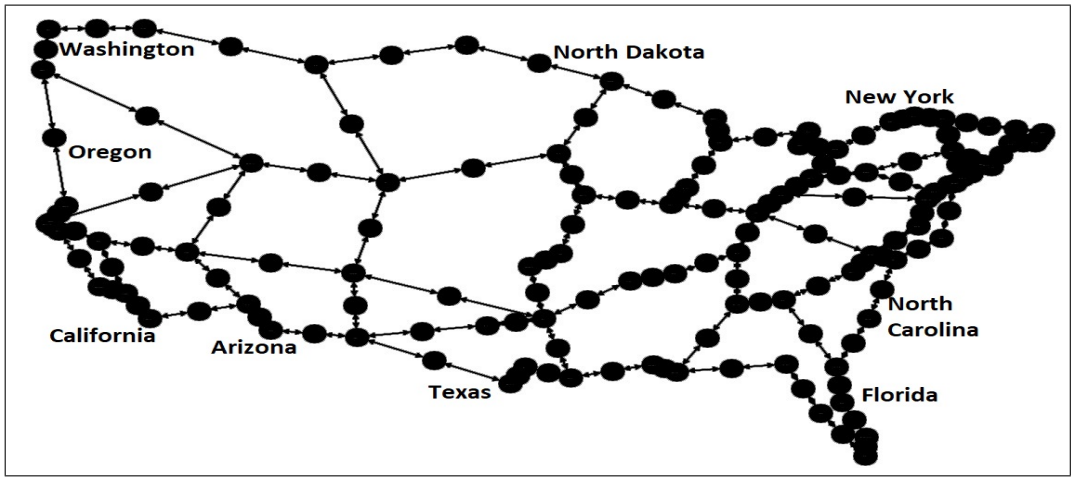
error-index = find control signals where $\mathcal{TS} \neq \mathcal{PS}$
for all $\mathcal{D} \in$ error-index **do**
    $Predicted_{Control\ States\ (ON/OFF)}$ (P_{Ctrl}) = $\mathcal{PS}(\mathcal{D})$
    $Actual_{Control\ States\ (ON/OFF)}$ (A_{Ctrl}) = $\mathcal{TS}(\mathcal{D})$
    $Actual_{Output\ Signals}$ (A_{otp}) = $\mathcal{TS}(\mathcal{D})$
    $Predicted_{Output\ Signals}$ (P_{otp}) = Test-Control-States(P_{Ctrl},$\mathcal{G}$,N)
    $Check_{Output\ Signals}$ (C_{otp}) = P_{otp}
    for $flip-bit \in M$ **do**
        if find $A_{otp} \neq C_{otp}$ **then**
            $Transit_{Control\ States\ (ON/OFF)}$ (T_{Ctrl}) = Flip-One-Bit(P_{Ctrl})
            C_{Ctrl} = Test-Control-States(T_{Ctrl},$\mathcal{G}$,N)
            Clear T_{Ctrl}
        else
            error-index corrected
            Break
        end if
    end for
end for

Table 1 Networks Topology Details

Topology Details		
Parameters	**EU: Training [165]**	**USA : Testing[166]**
Number of Nodes	28	100
Number of Links	41	171
Average path distance (km)	2014.06	2541.75
Maximum path distance (km)	3051.10	5481.07
Minimum path distance (km)	669.30	568.33
Average number of spans per Link	19.75	27.49



(a) EU network topology



(b) USA network topology

Acronyms

ADAGRAD Adaptive Gradient Algorithm. 63, 75, 91, 100

ADC Analog-to-digital. 5

AI Artificial intelligence. 12, 62

ANN Artificial Neural Network. 63

API Application program interface. 7, 44, 46, 58, 64, 66, 69, 75, 88, 89, 92, 99, 118

ARIMA Autoregressive integrated moving average. 36, 41

ASE Amplified spontaneous emission. 32, 47–49, 57, 69, 75

AWGN Additive white Gaussian noise. 48

BDM Band-division multiplexing. 4, 103

BER Bit error rate. 26, 27, 29, 31, 34, 35, 38–42, 97, 111

BP Back-propagation. 62

BPSK Binary phase shift keying. 30

BTR Boosted Tree Regressor. 63, 66–68, 90, 92–94

CAGR Compound annual growth rate. 1, 3, 4

CBR Case based reasoning. 28, 34, 35

CD Chromatic dispersion. 5, 31, 32, 35

CDR Call data records. 37

CIDC Communication Interface for Distributed Control plane. 10

CNN Convolutional neural network. 20, 43

DAC Digital-to-analog. 5

DCUs Dispersion compensating units. 3, 6

DD Direct-detection. 4, 5

DL Deep learning. 13, 32, 36, 41, 58

DMF Deciding modulation format. 30, 35

DMN Decision maker unit. 36

DNN Deep neural network. 19, 20, 63, 66–68, 75, 76, 88, 91–94, 96, 99–101

DP Dual polarization. 32

DPSK Differential Phase Shift Keying. 31

DQPSK Differential Quadrature Phase Shift Keying. 31

DR-MZI dual-ring assisted Mach-Zehnder interferometer. 42, 44

DRL Deep reinforcement learning. 43, 44

DSP Digital signal processing. 3–6, 47, 48

DTR Decision Tree Regressor. 61, 62, 64–66, 89, 92, 94

EDC Electronic dispersion compensation. 5

EDFA Erbium Doped Fiber Amplifier. 3, 28–30, 34, 35, 46, 50, 53, 69, 70

EM Expectation-Maximization. 24, 25, 30, 32, 35, 38, 39, 41, 42

EON Elastic optical networks. 2, 7, 30, 45

FCM Fuzzy C-Means Clustering. 40, 42

FEC Forward error correction. 2

FEELING FailurE causE Localization for optIcal NetworkinG. 38, 41

Flex-LIONS Flexible Low-latency Interconnect Optical Network Switch (Flex-LIONS). 43

FTRL Follow the Regularized Leader. 63

FTTH Fiber-to-the-home. 1, 38, 41

G Gain. 29, 34, 42, 47, 49, 50, 52–54, 69–71

GA Genetic algorithm. 37, 41

GGN Generalized Gaussian noise. 52

GMM Gaussian Mixture Mode. 24, 32, 33, 35

GMPLS Generalized Multi-Protocol Label Switching. 30

GNPy Gaussian noise simulation in Python. xii, 49, 52, 74–76

GPON Gigabit passive optical networks. 38, 41

GPR Gaussian processes nonlinear regression. 29, 34

GSNR Generalized Signal-to-Noise Ratio. v, xii, 46–52, 54–57, 69–77, 117

HCROS HoneyComb Rearrangeable Optical Switch. 85, 92, 94, 95

HMM Hidden Markov Model. 39, 42

HPC High-Performance Computing. 43

IMDD Intensity modulation with direct detection. 3

IoT Internet Of Things. 1, 13, 39, 45, 79

IP Internet Protocol. 39, 42, 45

ISI Inter-symbol interference. 33, 36

ITU International Telecommunication Union. 103

KNN K-nearest neighbor. 22, 29, 32, 34, 35

LOGO Local-Optimization Global-Optimization. 52

LP Lightpath. 3–5, 10, 11, 27–31, 34, 36–42, 46–49, 52, 57, 68–71, 75–78

LR Linear regressor. 20, 63, 67, 68, 88, 90–92, 94

MABP multi-armed bandit problem. 40

ML Machine learning. xii, 10–17, 22, 23, 25–32, 36, 37, 39, 40, 42–44, 46, 49–51, 57–59, 61–64, 66, 69, 70, 74–78, 80, 81, 83, 85–88, 90–92, 94, 96, 99–103, 108, 110–115

MLPR Multi-layer Perceptron Regressor. xi, 62, 64–68

MRR Microring Resonator. 81, 86, 105

MSE Mean square error. 16, 58–61, 63, 76, 88, 92, 94, 99

MZI Mach-Zehnder interferometer. 42, 44, 81, 104, 105, 111

NE Network element. 3, 4, 7, 42, 45, 46, 52, 69, 70, 74

NETCONF Network configuration protocol. 9

NF Noise figure. 29, 34, 47, 49, 50, 52–54, 70, 71

NFM Network failure management. 41

NLI Nonlinear interference. 32, 48, 49, 52, 53, 57, 69, 75

NMF Non-negative matrix factorization. 37

NMS network management system. 37

NN Neural network. 17–22, 28, 30–36, 39, 41, 42, 44, 63

NOS Networking Operating System. 9

NP Non-Polynomial. 83

NP-hard Non-deterministic Polynomial-time hardness. 118

NPDM Network planner and decision maker. 36

NRZ Non-Return to Zero. 31, 44

OAC Optical amplifier control. 34

OADMs Optical add/drop multiplexers. 4

OBS optical burst switched. 40

OCM optical channel monitor. 48, 50

ODB Optical Dual Binary. 31

OEO Optical-electro-optical. 5

OFDM Orthogonal frequency-division multiplexing. 32, 33

OLS Optical line system. v, xi, 3, 6, 28, 46, 47, 49, 50, 52, 55, 71

ONF Open Networking Foundation. 9, 10

ONN Optical neural network. 42, 43

ONTs Optical Network Terminals. 38

ONUs Optical Network Units. 38

OOK On-Off Keying. 31

OOS out-of-service. 37, 39, 70–72, 74, 77, 102, 114

OPEX Operational expenditures. 2

OPM Optical performance monitoring. 31, 35

OSE Optical Switching Element. 81, 83–85, 103–107, 111

OSEs Optical Switching Elements. 81

OSNR Optical Signal-to-Noise Ratio. v, 27, 28, 30, 31, 34, 35, 40, 42, 48, 50, 52, 97, 98, 100, 103, 111, 112, 114

PDF Probability density function. 70, 72

PDL Polarization-dependent loss. 31

PIC Photonic integrated circuit. 10, 80, 88, 89, 91, 96

PICs Photonic integrated circuits. 42, 79

PM Polarization-multiplexed. 30, 31, 86

PMD Polarization mode dispersion. 5, 31, 35

POF Protocol-Oblivious Forwarding. 9

PSK Phase shift keying. 30, 33

Q-factor Quality factor. 27, 28, 31, 34, 35

QAM Quadrature amplitude modulation. 3, 30–33, 48, 86

QoS Quality of Service. 2, 36, 39, 40, 42

QoT Quality of transmission. iv, v, 2, 11, 26–29, 46–48, 50, 51, 61, 62, 70, 74–77, 79, 80, 83, 84, 86, 96, 97, 99–102, 104, 107, 108, 110–114, 117

QoT-E Quality of transmission Estimation. 10, 11, 27, 28, 34, 46–50, 52, 58, 62, 63, 69–71, 74, 76–78, 117, 118

QPSK Quadrature phase shift keying. 29–31, 48

Relu Rectified linear unit. 19, 21, 62, 63, 91, 100

RFR Random Forest Regressor. 62, 64–66, 89, 90, 92–94

RJRMS Root Mean Squared jitter. 31, 35

RNN Recurrent neural networks. 21, 22

ROADM Reconfigurable optical add-drop multiplexer. xi, 7, 46–49, 52

ROADMs Reconfigurable optical add/drop multiplexers. 4, 6

RSA Routing and Spectrum Assignment. 118

RWA Routing and Wavelength Assignment. 118

RZ Return to Zero. 31

SC-NLI Self-channel nonlinear interference. 48

SDN software-defined networking. 2, 7–10, 31, 43, 45, 52, 79, 112

SDON software-defined optical network. iv, v, 7, 8, 11, 14, 40

SGD Stochastic Gradient Descent. 62

SKL *scikit-learn*©. xi, 58, 64–67, 88, 92

Skopt Scikit-optimize. xi, 59, 60

SLAs Service level agreements. 36, 37, 39

SNR Signal-to-Noise ratio. 48, 49

SRS Stimulated Raman Scattering. 48

SVM Support vector machine. 23, 29, 32–35, 38, 39, 41

Tanh Hyperbolic-tangent. 19, 62, 91, 100

TF *TensorFlow*©. xi, 58, 64, 66–69, 75, 88, 92, 99

TFO Traffic flow classification. 42

TIP Telecom Infra Project. 7

TISSUE Testing cal Switching at connection SetUp time. 38, 41

UWB Ultra-wideband. 11, 105, 113

VNT Virtual network topology. 36, 37

W-DNN Wide deep neural network. 20, 63, 64, 66–68

WDM Wavelength division multiplexing. 2–4, 29, 30, 45–47, 49, 53, 54, 70, 79, 103

XC-NLI Cross-channel nonlinear interference. 48

References

[1] V Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1(1), 2018.

[2] Y. Yamamoto and T. Kimura. Coherent optical fiber transmission systems. *IEEE Journal of Quantum Electronics*, 17(6):919–935, 1981.

[3] Ori Gerstel, Masahiko Jinno, Andrew Lord, and SJ Ben Yoo. Elastic optical networking: A new dawn for the optical layer? *IEEE Communications Magazine*, 50(2):s12–s20, 2012.

[4] Vittorio Curri. Software-defined wdm optical transport in disaggregated open optical networks. In *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2020.

[5] S Buzzelli, B Catania, D Gagliardi, and F Tosco. Optical fibre field experiments in italy: Cos1, cos2 and cos3/foster. In *International Conference on Communications*, pages 38–3, 1980.

[6] Jeff Hecht. *City of light: the story of fiber optics*. Oxford University Press on Demand, 2004.

[7] Robert J Mears, L Reekie, IM Jauncey, and David N Payne. Low-noise erbium-doped fibre amplifier operating at 1.54 μm. *Electronics Letters*, 23(19):1026–1028, 1987.

[8] Peter J Winzer and David T Neilson. From scaling disparities to integrated parallelism: A decathlon for a decade. *Journal of Lightwave Technology*, 35(5):1099–1115, 2017.

[9] Vijay Vusirikala, Xiaoxue Zhao, Tad Hofmeister, Valey Kamalov, Vinayak Dangui, and Bikash Koley. Scalable and flexible transport networks for inter-datacenter connectivity. In *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3. IEEE, 2015.

[10] S Hardy. Nokia upgrades 1830 pss packet-optical transport family with new coherent chipsets, improved multi-rate performance, 2016.

[11] Jonathan Homa and Krishna Bala. Roadm architectures and their enabling wss technology. *IEEE Communications Magazine*, 46(7):150–154, 2008.

[12] Sheryl L Woodward, Mark D Feuer, Paparao Palacharla, I Kaminow, T Li, and AE Willner. Roadm-node architectures for reconfigurable photonic networks. In *Optical Fiber Telecommunications*. Academic Press, 2013.

[13] Neo Antoniades, Georgios Ellinas, Jonathan Homa, and Krishna Bala. Roadm architectures and wss implementation technologies. *Convergence of Mobile and Stationary Next-Generation Networks*, pages 643–674, 2010.

[14] Adel AM Saleh and Jane M Simmons. All-optical networking—evolution, benefits, challenges, and future vision. *Proceedings of the IEEE*, 100(5):1105–1117, 2012.

[15] John Strand and Angela Chiu. Realizing the advantages of optical reconfigurability and restoration with integrated optical cross-connects. *Journal of Lightwave Technology*, 21(11):2871, 2003.

[16] Dany-Sebastien Ly-Gagnon, Satoshi Tsukamoto, Kazuhiro Katoh, and Kazuro Kikuchi. Coherent detection of optical quadrature phase-shift keying signals with carrier phase estimation. *Journal of lightwave technology*, 24(1):12, 2006.

[17] Michael G Taylor. Coherent detection method using dsp for demodulation of signal and subsequent equalization of propagation impairments. *IEEE Photonics Technology Letters*, 16(2):674–676, 2004.

[18] JM Dugan, AJ Price, M Ramadan, DL Wolf, EF Murphy, AJ Antos, DK Smith, and DW Hall. All-optical, fiber-based 1550 nm dispersion compensation in a 10 gbit/s, 150 km transmission experiment over 1310 nm optimized fiber. In *Optical Fiber Communication Conference*, page PD14. Optical Society of America, 1992.

[19] MI Hayee and AE Willner. Nrz versus rz in 10-40-gb/s dispersion-managed wdm transmission systems. *IEEE photonics technology letters*, 11(8):991–993, 1999.

[20] Han Sun, Kuang-Tsan Wu, and Kim Roberts. Real-time measurements of a 40 gb/s coherent system. *Optics express*, 16(2):873–879, 2008.

[21] Seb J Savory, Giancarlo Gavioli, Robert I Killey, and Polina Bayvel. Electronic compensation of chromatic dispersion using a digital coherent receiver. *Optics express*, 15(5):2120–2126, 2007.

[22] Henning Bülow, Fred Buchali, and Axel Klekamp. Electronic dispersion compensation. *Journal of Lightwave Technology*, 26(1):158–167, 2008.

[23] Seb J Savory. Digital coherent optical receivers: Algorithms and subsystems. *IEEE Journal of selected topics in quantum electronics*, 16(5):1164–1179, 2010.

[24] Xiang Zhou, Lynn E Nelson, and Peter Magill. Rate-adaptable optics for next generation long-haul transport networks. *IEEE Communications Magazine*, 51(3):41–49, 2013.

[25] Fernando P Guiomar, Rixin Li, Chris RS Fludger, Andrea Carena, and Vittorio Curri. Hybrid modulation formats enabling elastic fixed-grid optical networks. *Journal of Optical Communications and Networking*, 8(7):A92–A100, 2016.

[26] Vittorio Curri, P Poggiolini, A Carena, and F Forghieri. Dispersion compensation and mitigation of nonlinear effects in 111-gb/s wdm coherent pm-qpsk systems. *IEEE photonics technology letters*, 20(17):1473–1475, 2008.

[27] Valey Kamalov, Vinayak Dangui, Tad Hofmeister, Bikash Koley, Chris Mitchell, Matt Newland, John O'Shea, Cody Tomblin, Vijay Vusirikala, and Xiaoxue Zhao. Lessons learned from open line system deployments. In *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3. IEEE, 2017.

[28] Gazettabyte, "the white box concept gets embraced at the optical layer", Dec 2016.

[29] Openconfig.

[30] Global community connectivity collaboration, Sep 2021.

[31] Openroadm.

[32] Ioannis Tomkos, Siamak Azodolmolky, Josep Sole-Pareta, Davide Careglio, and Eleni Palkopoulou. A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges. *Proceedings of the IEEE*, 102(9):1317–1337, 2014.

[33] Guoying Zhang, Marc De Leenheer, Annalisa Morea, and Biswanath Mukherjee. A survey on ofdm-based elastic core optical networking. *IEEE Communications Surveys & Tutorials*, 15(1):65–87, 2012.

[34] Sakir Sezer, Sandra Scott-Hayward, Pushpinder Kaur Chouhan, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Marc Miller, and Navneet Rao. Are we ready for sdn? implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7):36–43, 2013.

[35] JW Lockwood, N McKeown, G Watson, G Gibb, P Hartke, J Naous, R Raghuraman, and J Luo. An open platform for gigabit-rate network switching and routing. In *Proc. Int. Conf. Microelectronic System Education*, pages 160–161.

[36] Open vswitch.

[37] Indigo: Open source openflow switches.

[38] Pantou: Openflow 1.3 for openwrt.

[39] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM computer communication review*, 38(3):105–110, 2008.

[40] About pox.

[41] Ryu : Sdn framework.

[42] Jan Medved, Robert Varga, Anton Tkacik, and Ken Gray. Opendaylight: Towards a model-driven sdn controller architecture. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–6. IEEE, 2014.

[43] David Erickson. The beacon openflow controller. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 13–18, 2013.

[44] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74, 2008.

[45] Haoyu Song. Protocol-oblivious forwarding: Unleash the power of sdn through a future-proof forwarding plane. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 127–132, 2013.

[46] Rob Enns, Martin Bjorklund, Juergen Schoenwaelder, and Andy Bierman. Network configuration protocol (netconf). 2011.

[47] M Smith, M Dvorkin, Y Laribi, V Pandey, P Garg, and N Weidenbacher. Opflex control protocol. *IETF, Apr*, 2014.

[48] Open networking foundation, Sep 2021.

[49] H Yin, H Xie, T Tsou, D Lopez, P Aranda, R Sidi, and SDNi. A message exchange protocol for software defined networks (sdns) across multiple domains. *Online: https://tools. ietf. org/html/draft-yin-sdn-sdni-00.[Last Accessed: July 26, 2017]*, 2012.

[50] Fouad Benamrane, Redouane Benaini, et al. An east-west interface for distributed sdn control plane: Implementation and evaluation. *Computers & Electrical Engineering*, 57:162–175, 2017.

[51] Pingping Lin, Jun Bi, and Yangyang Wang. East-west bridge for sdn network peering. In *Frontiers in internet technologies*, pages 170–181. Springer, 2013.

[52] Faisal Nadeem Khan, Qirui Fan, Chao Lu, and Alan Pak Tao Lau. Machine learning methods for optical communication systems and networks. In *Optical fiber telecommunications VII*, pages 921–978. Elsevier, 2020.

[53] Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

[54] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[55] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[56] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):1–20, 2021.

[57] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

[58] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

[59] Fei Wang, Lawrence Peter Casalino, and Dhruv Khullar. Deep learning in medicine—promise, progress, and challenges. *JAMA internal medicine*, 179(3):293–294, 2019.

[60] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. WS, 2008.

[61] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[62] Jane Elith, John R Leathwick, and Trevor Hastie. A working guide to boosted regression trees. *JAE*, 77(4):802–813, 2008.

[63] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[64] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[65] C. Nwankpa et.al. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[66] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. *CoRR*, abs/1606.07792, 2016.

[67] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495, 2017.

[68] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. *Journal of Machine Learning Research*, 2015.

[69] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[70] Francesco Musumeci, Cristina Rottondi, Avishek Nag, Irene Macaluso, Darko Zibar, Marco Ruffini, and Massimo Tornatore. An overview on application of machine learning techniques in optical networks. *IEEE Communications Surveys & Tutorials*, 21(2):1383–1408, 2018.

[71] Faisal Nadeem Khan, Qirui Fan, Chao Lu, and Alan Pak Tao Lau. Machine learning methods for optical communication systems and networks. In *Optical fiber telecommunications VII*, pages 921–978. Elsevier, 2020.

[72] Andrea D'Amico, Stefano Straullu, Antonino Nespola, Ihtesham Khan, Elliot London, Emanuele Virgillito, Stefano Piciaccia, Aberto Tanzi, Gabriele Galimberti, and Vittorio Curri. Using machine learning in an open optical line system controller. *JOCN*, 12(6):C1–C11, 2020.

[73] Tamara Jimenez, Juan Carlos Aguado, Ignacio de Miguel, Ramón J. Duran, Marianna Angelou, Noemí Merayo, Patricia Fernandez, Rubén M. Lorenzo, Ioannis Tomkos, and Evaristo J. Abril. A cognitive quality of transmission estimator for core optical networks. *Journal of Lightwave Technology*, 31(6):942–951, 2013.

[74] Antonio Caballero, Juan Carlos Aguado, Robert Borkowski, Silvia Saldaña, Tamara Jiménez, Ignacio de Miguel, Valeria Arlunno, Ramón J. Durán, Darko Zibar, Jesper B. Jensen, Rubén M. Lorenzo, Evaristo J. Abril, and Idelfonso Tafur Monroy. Experimental demonstration of a cognitive quality of transmission estimator for optical communication systems. In *2012 38th European Conference and Exhibition on Optical Communications*, pages 1–3, 2012.

[75] Tania Panayiotou, Georgios Ellinas, and Sotirios P. Chatzis. A data-driven qot decision approach for multicast connections in metro optical networks. In *2016 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6, 2016.

[76] T. Panayiotou, S. P. Chatzis, and G. Ellinas. Performance analysis of a data-driven quality-of-transmission decision approach on a dynamic multicast-capable metro optical network. *IEEE/OSA Journal of Optical Communications and Networking*, 9(1):98–108, 2017.

[77] Weiyang Mo, Yue-Kai Huang, Shaoliang Zhang, Ezra Ip, Daniel C. Kilper, Yoshiaki Aono, and Tsutomu Tajima. Ann-based transfer learning for qot prediction in real-time mixed line-rate systems. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[78] R. Proietti, X. Chen, A. Castro, G. Liu, H. Lu, K. Zhang, J. Guo, Z. Zhu, L. Velasco, and S. J. B. Yoo. Experimental demonstration of cognitive provisioning and alien wavelength monitoring in multi-domain eon. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[79] Cristina Rottondi, Luca Barletta, Alessandro Giusti, and Massimo Tornatore. Machine-learning method for quality of transmission prediction of unestablished lightpaths. *IEEE/OSA Journal of Optical Communications and Networking*, 10(2):A286–A297, 2018.

[80] Sandra Aladin and Christine Tremblay. Cognitive tool for estimating the qot of new lightpaths. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[81] Jakob Thrane, Jesper Wass, Molly Piels, Julio C. M. Diniz, Rasmus Jones, and Darko Zibar. Machine learning techniques for optical performance monitoring from directly detected pdm-qam signals. *Journal of Lightwave Technology*, 35(4):868–875, 2017.

[82] Martin Bouda, Shoichiro Oda, Olga Vasilieva, Masatake Miyabe, Setsuo Yoshida, Toru Katagiri, Yasuhiko Aoki, Takeshi Hoshida, and Tadashi Ikeuchi. Accurate prediction of quality of transmission with dynamically configurable optical impairment model. In *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3. IEEE, 2017.

[83] Uiara Moura, Miquel Garrich, Heitor Carvalho, Matheus Svolenski, Alexandre Andrade, Amílcar C Cesar, Juliano Oliveira, and Evandro Conforti. Cognitive methodology for optical amplifier gain adjustment in dynamic dwdm networks. *Journal of Lightwave Technology*, 34(8):1971–1979, 2016.

[84] Juliano Oliveira, Antonio Caballero, Eduardo Magalhães, Uiara Moura, Robert Borkowski, Giovanni Curiel, Alberto Hirata, Luis Hecker, Edson Porto, Darko Zibar, et al. Demonstration of edfa cognitive gain control via gmpls for mixed modulation formats in heterogeneous optical networks. In *Optical Fiber Communication Conference*, pages OW1H–2. Optical Society of America, 2013.

[85] Erick de A Barboza, Carmelo JA Bastos-Filho, Joaquim F Martins-Filho, Uiara C de Moura, and Juliano RF de Oliveira. Self-adaptive erbium-doped fiber amplifiers using machine learning. In *2013 SBMO/IEEE MTT-S International Microwave & Optoelectronics Conference (IMOC)*, pages 1–5. IEEE, 2013.

[86] Carmelo JA Bastos-Filho, Erick de A Barboza, Joaquim F Martins-Filho, Uiara C de Moura, and Juliano RF de Oliveira. Mapping edfa noise figure and gain flatness over the power mask using neural networks. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications (JMOe)*, 12:si128–si139, 2013.

[87] Uiara C de Moura, Juliano RF Oliveira, Júlio CRF Oliveira, and Amílcar C César. Edfa adaptive gain control effect analysis over an amplifier cascade in a dwdm optical system. In *2013 SBMO/IEEE MTT-S International Microwave & Optoelectronics Conference (IMOC)*, pages 1–5. IEEE, 2013.

[88] Ricard Boada, Robert Borkowski, and Idelfonso Tafur Monroy. Clustering algorithms for stokes space modulation format recognition. *Optics express*, 23(12):15521–15531, 2015.

[89] Robert Borkowski, Darko Zibar, Antonio Caballero, Valeria Arlunno, and Idelfonso Tafur Monroy. Stokes space-based optical modulation format recognition for digital coherent receivers. *IEEE Photonics Technology Letters*, 25(21):2129–2132, 2013.

[90] Faisal Nadeem Khan, Yudi Zhou, Alan Pak Tao Lau, and Chao Lu. Modulation format identification in heterogeneous fiber-optic networks using artificial neural networks. *Optics express*, 20(11):12422–12431, 2012.

[91] Faisal Nadeem Khan, Kangping Zhong, Waled Hussein Al-Arashi, Changyuan Yu, Chao Lu, and Alan Pak Tao Lau. Modulation format identification in coherent receivers using deep machine learning. *IEEE Photonics Technology Letters*, 28(17):1886–1889, 2016.

[92] Shuting Zhang, Yingqing Peng, Qi Sui, Jianping Li, and Zhaohui Li. Modulation format identification in heterogeneous fiber-optic networks using artificial neural networks and genetic algorithms. *Photonic Network Communications*, 32(2):246–252, 2016.

[93] Xiaoxia Wu, Jeffrey A. Jargon, Ronald A. Skoog, Loukas Paraschis, and Alan E. Willner. Applications of artificial neural networks in optical performance monitoring. *Journal of Lightwave Technology*, 27(16):3580–3589, 2009.

[94] Jeffrey A. Jargon, Xiaoxia Wu, Hyeon Yeong Choi, Yun C. Chung, and Alan E. Willner. Optical performance monitoring of qpsk data channels by use of neural networks trained with parameters derived from asynchronous constellation diagrams. *Opt. Express*, 18(5):4931–4938, Mar 2010.

[95] Thomas Shun Rong Shen, Ke Meng, Alan Pak Tao Lau, and Zhao Yang Dong. Optical performance monitoring using artificial neural network trained with asynchronous amplitude histograms. *IEEE Photonics Technology Letters*, 22(22):1665–1667, 2010.

[96] Faisal Nadeem Khan, Thomas Shun Rong Shen, Yudi Zhou, Alan Pak Tao Lau, and Chao Lu. Optical performance monitoring using artificial neural networks trained with empirical moments of asynchronously sampled signal amplitudes. *IEEE Photonics Technology Letters*, 24(12):982–984, 2012.

[97] Trevor B. Anderson, Adam Kowalczyk, Ken Clarke, Sarah D. Dods, Don Hewitt, and Jonathan C. Li. Multi impairment monitoring for optical networks. *Journal of Lightwave Technology*, 27(16):3729–3736, 2009.

[98] Takahito Tanimura, Takeshi Hoshida, Jens C. Rasmussen, Makoto Suzuki, and Hiroyuki Morikawa. Osnr monitoring by deep neural networks trained with asynchronously sampled data. In *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, pages 1–3, 2016.

[99] Takahito Tanimura, Takeshi Hoshida, Tomoyuki Kato, Shigeki Watanabe, and Hiroyuki Morikawa. Data-analytics-based optical performance monitoring technique for optical transport networks. In *Optical Fiber Communication Conference*, pages Tu3E–3. Optical Society of America, 2018.

[100] Govind P Agrawal. Nonlinear fiber optics. In *Nonlinear Science at the Dawn of the 21st Century*, pages 195–211. Springer, 2000.

[101] A. Carena, V. Curri, G. Bosco, P. Poggiolini, and F. Forghieri. Modeling of the impact of nonlinear propagation effects in uncompensated optical coherent transmission links. *Journal of Lightwave Technology*, 30(10):1524–1539, 2012.

[102] Francesco Vacondio, Olivier Rival, Christian Simonneau, Edouard Grellier, Alberto Bononi, Laurence Lorcy, Jean-Christophe Antona, and Sébastien Bigo. On nonlinear distortions of highly dispersive optical coherent systems. *Opt. Express*, 20(2):1022–1032, Jan 2012.

[103] Pontus Johannisson and Erik Agrell. Modeling of nonlinear signal distortion in fiber-optic networks. *Journal of Lightwave Technology*, 32(23):4544–4552, 2014.

[104] Darko Zibar, Ole Winther, Niccolo Franceschi, Robert Borkowski, Antonio Caballero, Valeria Arlunno, Mikkel N Schmidt, Neil Guerrero Gonzales, Bangning Mao, Yabin Ye, et al. Nonlinear impairment compensation using expectation maximization for dispersion managed and unmanaged pdm 16-qam transmission. *Optics express*, 20(26):B181–B196, 2012.

[105] Danshi Wang, Min Zhang, Meixia Fu, Zhongle Cai, Ze Li, Huanhuan Han, Yue Cui, and Bin Luo. Nonlinearity mitigation using a machine learning detector based on *k*-nearest neighbors. *IEEE Photonics Technology Letters*, 28(19):2102–2105, 2016.

[106] Mutsam A Jarajreh, Elias Giacoumidis, Ivan Aldaya, Son Thai Le, Athanasios Tsokanos, Zabih Ghassemlooy, and Nick J Doran. Artificial neural network nonlinear equalizer for coherent optical ofdm. *IEEE Photonics Technology Letters*, 27(4):387–390, 2014.

[107] Thomas Shun Rong Shen and Alan Pak Tao Lau. Fiber nonlinearity compensation using extreme learning machine for dsp-based coherent communication systems. In *16th Opto-Electronics and Communications Conference*, pages 816–817. IEEE, 2011.

[108] Elias Giacoumidis, Sofien Mhatli, Marc FC Stephens, Athanasios Tsokanos, Jinlong Wei, Mary E McCarthy, Nick J Doran, and Andrew D Ellis. Reduction of nonlinear intersubcarrier intermixing in coherent optical ofdm by a fast newton-based support vector machine nonlinear equalizer. *Journal of Lightwave Technology*, 35(12):2391–2397, 2017.

[109] Danshi Wang, Min Zhang, Ze Li, Yue Cui, Jingdan Liu, Yang Yang, and Hongxiang Wang. Nonlinear decision boundary created by a machine learning-based classifier to mitigate nonlinear phase noise. In *2015 European Conference on Optical Communication (ECOC)*, pages 1–3. IEEE, 2015.

[110] Feng Lu, Peng-Chun Peng, Siming Liu, Mu Xu, Shuyi Shen, and Gee-Kung Chang. Integration of multivariate gaussian mixture model for enhanced pam-4 decoding employing basis expansion. In *Optical Fiber Communication Conference*, pages M2F–1. Optical Society of America, 2018.

[111] Peixuan Li, Lilin Yi, Lei Xue, and Weisheng Hu. 56 gbps im/dd pon based on 10g-class optical devices with 29 db loss budget enabled by machine learning. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3. IEEE, 2018.

[112] Chun-Yen Chuang, Li-Chun Liu, Chia-Chien Wei, Jun-Jie Liu, Lindor Henrickson, Wan-Jou Huang, Chih-Lin Wang, Young-Kai Chen, and Jyehong Chen. Convolutional neural network based nonlinear classifier for 112-gbps high speed optical link. In *Optical Fiber Communication Conference*, pages W2A–43. Optical Society of America, 2018.

[113] Peixuan Li, Lilin Yi, Lei Xue, and Weisheng Hu. 100gbps im/dd transmission over 25km ssmf using 20g-class dml and pin enabled by machine learning. In *Optical Fiber Communication Conference*, pages W2A–46. Optical Society of America, 2018.

[114] Rasmus T Jones, Simone Gaiarin, Metodi P Yankov, and Darko Zibar. Noise robust receiver for eigenvalue communication systems. In *Optical Fiber Communication Conference*, pages W2A–59. Optical Society of America, 2018.

[115] Christian Häger and Henry D Pfister. Nonlinear interference mitigation via deep neural networks. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3. IEEE, 2018.

[116] Siming Liu, Yahya M Alfadhli, Shuyi Shen, Huiping Tian, and Gee-Kung Chang. Mitigation of multi-user access impairments in 5g a-rof-based mobile fronthaul utilizing machine learning for an artificial neural network nonlinear

equalizer. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3. IEEE, 2018.

[117] Junfeng Zhang, Wei Chen, Mingyi Gao, Bowen Chen, and Gangxiang Shen. Novel low-complexity fully-blind density-centroid–tracking equalizer for 64-qam coherent optical communication systems. In *Optical Fiber Communication Conference*, pages M1G–4. Optical Society of America, 2018.

[118] Neil Guerrero Gonzalez, Darko Zibar, and Idelfonso Tafur Monroy. Cognitive digital receiver for burst mode phase modulated radio over fiber links. In *36th European Conference and Exhibition on Optical Communication*, pages 1–3. IEEE, 2010.

[119] Natalia Fernández, Ramón J Durán, Ignacio de Miguel, Noemí Merayo, Patricia Fernández, Juan Carlos Aguado, Rubén M Lorenzo, Evaristo J Abril, Eleni Palkopoulou, and Ioannis Tomkos. Virtual topology design and reconfiguration using cognition: Performance evaluation in case of failure. In *2013 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 132–139. IEEE, 2013.

[120] Natalia Fernández, Ramón J Durán Barroso, Domenico Siracusa, Antonio Francescon, Ignacio de Miguel, Elio Salvadori, Juan Carlos Aguado, and Rubén M Lorenzo. Virtual topology reconfiguration in optical networks by means of cognition: Evaluation and experimental validation. *Journal of Optical Communications and Networking*, 7(1):A162–A173, 2015.

[121] A. C. Harvey. *ARIMA Models*, pages 22–24. Palgrave Macmillan UK, London, 1990.

[122] Fernando Morales, Marc Ruiz, and Luis Velasco. Virtual network topology reconfiguration based on big data analytics for traffic prediction. In *Optical Fiber Communication Conference*, pages Th3I–5. Optical Society of America, 2016.

[123] Fernando Morales, Marc Ruiz, Lluís Gifre, Luis M Contreras, Víctor López, and Luis Velasco. Virtual network topology adaptability based on data analytics for traffic prediction. *Journal of Optical Communications and Networking*, 9(1):A35–A45, 2017.

[124] Ao Yu, Hui Yang, Wei Bai, Linkuan He, Hongyun Xiao, and Jie Zhang. Leveraging deep learning to achieve efficient resource allocation with traffic evaluation in datacenter optical networks. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3. IEEE, 2018.

[125] Ll Gifre, F Morales, L Velasco, and M Ruiz. Big data analytics for the virtual network topology reconfiguration use case. In *2016 18th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2016.

[126] Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata. A bayesian-based approach for virtual network reconfiguration in elastic optical path networks. In *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, 2017.

[127] Sebastian Troia, Gao Sheng, Rodolfo Alvizu, Guido Alberto Maier, and Achille Pattavina. Identification of tidal-traffic patterns in metro-area mobile networks via matrix factorization based model. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 297–301, 2017.

[128] Natalia Fernández, Ramón J Durán, Ignacio de Miguel, Noemí Merayo, David Sánchez, Marianna Angelou, Juan Carlos Aguado, Patricia Fernández, Tamara Jiménez, Rubén M Lorenzo, et al. Cognition to design energetically efficient and impairment aware virtual topologies for optical networks. In *2012 16th International Conference on Optical Network Design and Modelling (ONDM)*, pages 1–6. IEEE, 2012.

[129] Natalia Fernández, Ramón J Durán, Ignacio de Miguel, Noemí Merayo, Juan Carlos Aguado, Patricia Fernández, Tamara Jiménez, Ignacio Rodríguez, David Sánchez, Rubén M Lorenzo, et al. Survivable and impairment-aware virtual topologies for reconfigurable optical networks: A cognitive approach. In *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pages 793–799. IEEE, 2012.

[130] K. Christodoulopoulos, N. Sambo, and E. Varvarigos. Exploiting network kriging for fault localization. In *Optical Fiber Communication Conference*, page W1B.5. Optical Society of America, 2016.

[131] Marc Ruiz, Francesco Fresi, Alba P. Vela, Gianluca Meloni, Nicola Sambo, Filippo Cugini, Luca Poti, Luis Velasco, and Piero Castoldi. Service-triggered failure identification/localization through monitoring of multiple parameters. In *ECOC 2016; 42nd European Conference on Optical Communication*, pages 1–3, 2016.

[132] Serge Romaric Tembo, Sandrine Vaton, Jean-Luc Courant, and Stéphane Gosselin. A tutorial on the em algorithm for bayesian networks: Application to self-diagnosis of gpon-ftth networks. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 369–376, 2016.

[133] Stéphane Gosselin, Jean-Luc Courant, Serge Romaric Tembo, and Sandrine Vaton. Application of probabilistic modeling and machine learning to the diagnosis of ftth gpon networks. In *2017 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–3, 2017.

[134] Alba P. Vela, Marc Ruiz, Francesco Fresi, Nicola Sambo, Filippo Cugini, Gianluca Meloni, Luca Potì, Luis Velasco, and Piero Castoldi. Ber degradation detection and failure identification in elastic optical networks. *J. Lightwave Technol.*, 35(21):4595–4604, Nov 2017.

[135] A. P. Vela, B. Shariati, M. Ruiz, F. Cugini, A. Castro, H. Lu, R. Proietti, J. Comellas, P. Castoldi, S. J. B. Yoo, and L. Velasco. Soft failure localization during commissioning testing and lightpath operation. *J. Opt. Commun. Netw.*, 10(1):A27–A36, Jan 2018.

[136] Shahin Shahkarami, Francesco Musumeci, Filippo Cugini, and Massimo Tornatore. Machine-learning-based soft-failure detection and identification in optical networks. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[137] Danish Rafique, Thomas Szyrkowiec, Achim Autenrieth, and Joerg-Peter Elbers. Analytics-driven fault discovery and diagnosis for cognitive root cause analysis. In *Optical Fiber Communication Conference*, page W4F.6. Optical Society of America, 2018.

[138] A. Jayaraj, T. Venkatesh, and C. Siva Ram Murthy. Loss classification in optical burst switching networks using machine learning techniques: improving the performance of tcp. *IEEE Journal on Selected Areas in Communications*, 26(6):45–54, 2008.

[139] Houman Rastegarfar, Madeleine Glick, Nicolaas Viljoen, Mingwei Yang, John Wissinger, Lloyd Lacomb, and Nasser Peyghambarian. Tcp flow classification and bandwidth aggregation in optically interconnected data center networks. *IEEE/OSA Journal of Optical Communications and Networking*, 8(10):777–786, 2016.

[140] Y. V. Kiran, T. Venkatesh, and C. Siva Ram Murthy. A reinforcement learning framework for path selection and wavelength selection in optical burst switched networks. *IEEE Journal on Selected Areas in Communications*, 25(9):18–26, 2007.

[141] A. Jayaraj, T. Venkatesh, and C. Siva Ram Murthy. Loss classification in optical burst switching networks using machine learning techniques: improving the performance of tcp. *IEEE Journal on Selected Areas in Communications*, 26(6):45–54, 2008.

[142] W. Gao et. al. Automatic calibration of silicon ring-based optical switch powered by machine learning. *Opt. Express*, 28(7):10438–10455, Mar 2020.

[143] Hailong Zhou, Yuhe Zhao, Xu Wang, Dingshan Gao, Jianji Dong, and Xinliang Zhang. Self-learning photonic signal processor with an optical neural network chip. *arXiv preprint arXiv:1902.07318*, 2019.

[144] R. Proietti et. al. Self-driving reconfiguration of data center networks by deep reinforcement learning and silicon photonic flex-lion switches. In *2020 IPC*, pages 1–2, 2020.

[145] Saim Salman, Christopher Streiffer, Huan Chen, Theophilus Benson, and Asim Kadav. Deepconf: Automating data center network topologies management with machine learning. In *Proceedings of the 2018 Workshop on*

Network Meets AI & ML, NetAI'18, page 8–14, New York, NY, USA, 2018. Association for Computing Machinery.

[146] Cisco. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. Technical report, Cisco, 2017.

[147] Vittorio Curri, Andrea Carena, Andrea Arduino, Gabriella Bosco, Pierluigi Poggiolini, Antonino Nespola, and Fabrizio Forghieri. Design strategies and merit of system parameters for uniform uncompensated links supporting nyquist-WDM transmission. *JLT*, 33(18):3921–3932, sep 2015.

[148] Vittorio Curri. Software-defined wdm optical transport in disaggregated open optical networks. In *ICTON 2020*, page We.C2.1, 2020.

[149] Rosanna Pastorelli. Network optimization strategies and control plane impacts. In *OFC*. OSA, 2015.

[150] V. Curri and A. Carena. Merit of raman pumping in uniform and uncompensated links supporting nywdm transmission. *Journal of Lightwave Technology*, 34(2):554–565, 2016.

[151] R. Pastorelli, S. Piciaccia, G. Galimberti, E. Self, M. Brunella, G. Calabretta, F. Forghieri, D. Siracusa, A. Zanardi, E. Salvadori, G. Bosco, A. Carena, V. Curri, and P. Poggiolini. Optical control plane based on an analytical model of non-linear transmission effects in a self-optimized network. In *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, pages 1–3, 2013.

[152] https://www.itu.int/rec/T-REC-G.694.1/en.

[153] Dario Pilori, Fabrizio Forghieri, and Gabriella Bosco. Residual non-linear phase noise in probabilistically shaped 64-qam optical links. In *OFC*, 2018.

[154] M. Cantono, R. Gaudino, and V. Curri. Potentialities and criticalities of flexible-rate transponders in dwdm networks: A statistical approach. *IEEE/OSA Journal of Optical Communications and Networking*, 8(7):A76–A85, 2016.

[155] Emanuele Virgillito, Andrea D'Amico, Alessio Ferrari, and Vittorio Curri. Observing and modeling wideband generation of non-linear interference. In *2019 21st International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2019.

[156] Andrea Carena, Vittorio Curri, Gabriella Bosco, Pierluigi Poggiolini, and F Forghieri. Modeling of the impact of nonlinear propagation effects in uncompensated optical coherent transmission links. *JLT*, 30(10):1524–1539, may 2012.

[157] Antonio Mecozzi and René-Jean Essiambre. Nonlinear shannon limit in pseudolinear coherent systems. *Journal of Lightwave Technology*, 30(12):2011–2024, 2012.

[158] Mattia Cantono, Dario Pilori, Alessio Ferrari, Clara Catanese, Jordane Thouras, Jean-Luc Augé, and Vittorio Curri. On the Interplay of Nonlinear Interference Generation with Stimulated Raman Scattering for QoT Estimation. *JLT*, PP(99):1–1, 2018.

[159] Alessio Ferrari, Mark Filer, Karthikeyan Balasubramanian, Yawei Yin, Esther Le Rouzic, Jan Kundrát, Gert Grammel, Gabriele Galimberti, and Vittorio Curri. Gnpy: an open source application for physical layer aware open optical networks. *JOCN*, 12(6):C31–C40, 2020.

[160] Yasuhuo Ando. Statistical analysis of insertion-loss improvement for optical connectors using the orientation method for fiber-core offset. *IEEE Photonics Technology Letters*, 3(10):939–941, 1991.

[161] M. Filer, J. Gaudette, Y. Yin, D. Billor, Z. Bakhtiari, and J. L. Cox. Low-margin optical networking at cloud scale [invited]. *IEEE/OSA Journal of Optical Communications and Networking*, 11(10):C94–C108, 2019.

[162] Vittorio Curri, Andrea Carena, Andrea Arduino, Gabriella Bosco, Pierluigi Poggiolini, Antonino Nespola, and Fabrizio Forghieri. Design strategies and merit of system parameters for uniform uncompensated links supporting nyquist-wdm transmission. *JLT*, 33(18):3921–3932, 2015.

[163] Maxim Bolshtyansky. Spectral hole burning in erbium-doped fiber amplifiers. *JLT*, 21(4):1032–1038, 2003.

[164] Telecominfraproject. Telecominfraproject/oopt-gnpy, Sep 2019.

[165] Emmanuel Seve, Jelena Pesic, Camille Delezoide, S Bigo, and Y Pointurier. Learning process for reducing uncertainties on network parameters and design margins. *Journal of Optical Communications and Networking*, 10(2):A298–A306, 2018.

[166] S. K. Korotky. Network global expectation model: a statistical formalism for quickly quantifying network needs and costs. *Journal of Lightwave Technology*, 22(3):703–722, 2004.

[167] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[168] Gavin Hackeling. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.

[169] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} ({OSDI} 16)*, pages 265–283, 2016.

[170] Gilles Louppe, Manoj Kumar, and H Nahrstaedt. Bayesian optimization with skopt. *Github*, 2016.

[171] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[172] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[173] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(Jul):2121–2159, 2011.

[174] H Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. Technical report, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011.

[175] Mark Filer, Mattia Cantono, Alessio Ferrari, Gert Grammel, Gabriele Galimberti, and Vittorio Curri. Multi-Vendor Experimental Validation of an Open Source QoT Estimator for Optical Networks. *JLT*, 36(15):3073–3082, aug 2018.

[176] Prabhaker Mishra, Chandra M Pandey, Uttam Singh, Anshul Gupta, Chinmoy Sahu, and Amit Keshri. Descriptive statistics and normality tests for statistical data. *Annals of cardiac anaesthesia*, 22(1):67, 2019.

[177] Vittorio Curri. Software-defined WDM optical transport in disaggregated open optical networks. In *2020 ICTON*, pages 1–4, 2020.

[178] L. Velasco et. al. Building autonomic optical whitebox-based networks. *J. Lightwave Technol.*, 36(15):3097–3104, Aug 2018.

[179] Muhammad Rehan Yahya, Ning Wu, Gaizhen Yan, Tanveer Ahmed, Jinbao Zhang, and Yuanyuan Zhang. Honeycomb ROS: A 6×6 non-blocking optical switch with optimized reconfiguration for ONoCs. *Electronics*, 8(8):844, Jul 2019.

[180] L. Tunesi et. al. Automatic design of NxN integrated Benes optical switch. In Graham T. Reed and Andrew P. Knights, editors, *Silicon Photonics XVI*, volume 11691, pages 164 – 174. International Society for Optics and Photonics, SPIE, 2021.

[181] I. Khan et.al. Machine-learning-aided abstraction of photonic integrated circuits in software-defined optical transport. volume 11713, pages 146 – 151. SPIE, 2021.

[182] E. Ghillino et.al. The Synopsys software environment to design and simulate photonic integrated circuits: A case study for 400G transmission. In *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, 2018.

[183] https://www.tensorflow.org/.

[184] Joao Pedro, Nelson Costa, and Silvia Pato. Optical transport network design beyond 100gbaud [invited]. *IEEE/OSA Journal of Optical Communications and Networking*, 12(2):A123–A134, 2020.

[185] Alessio Ferrari, Antonio Napoli, Johannes K. Fischer, Nelson Costa, Andrea D'Amico, João Pedro, Wladek Forysiak, Erwan Pincemin, Andrew Lord, Alexandros Stavdas, Juan Pedro F.-P. Gimenez, Gunther Roelkens, Nicola Calabretta, Silvio Abrate, Bernd Sommerkorn-Krombholz, and Vittorio Curri. Assessment on the achievable throughput of multi-band itu-t g.652.d fiber transmission systems. *Journal of Lightwave Technology*, 38(16):4279–4291, 2020.

[186] Renato Orta, Guido Perrone, Riccardo Tascone, Antonello Fincato, Maurizio Lenzi, Stefano Lorenzotti, and Peter Nugent. Design technique for wideband optical couplers. In *Fiber Optic Network Components*, volume 2449, pages 375 – 383. SPIE, 1995.

[187] Amitabha Chakrabarty, Martin Collier, and Sourav Mukhopadhyay. Matrix-based nonblocking routing algorithm for Beneš networks. In *Future computing 2009*, pages 551–556. IEEE, 2009.

[188] Enrico Ghillino, Emanuele Virgillito, Pablo V. Mena, Rob Scarmozzino, Remco Stoffer, Dwight Richards, Ali Ghiasi, Alessio Ferrari, Mattia Cantono, Andrea Carena, and Vittorio Curri. The Synopsys software environment to design and simulate photonic integrated circuits: A case study for 400G transmission. In *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, 2018.