Optimal control of Beneš optical networks assisted by machine learning

Optimal control of Beneš optical networks assisted by machine learning

Ihtesham Khan[a], Lorenzo Tunesi[a], Muhammad Umar Masood[a], Enrico Ghillino[b], Paolo Bardella[a], Andrea Carena[a], and Vittorio Curri[a]

[a]Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy
[b]Synopsys Inc., Executive Blvd 101, Ossining, New York, USA

ABSTRACT

Beneš networks represent an excellent solution for the routing of optical telecom signals in integrated, fully reconfigurable networks because of their limited number of elementary 2x2 crossbar switches and their non-blocking properties. Various solutions have been proposed to determine a proper Control State (CS) providing the required permutation of the input channels; since for a particular permutation, the choice is not unique, the number of cross-points has often been used to estimate the cost of the routing operation. This work presents an advanced version of this approach: we deterministically estimate all (or a reasonably large number of) the CSs corresponding to the permutation requested by the user. After this, the retrieved CSs are exploited by a data-driven framework to predict the Optical Signal to Noise Ratio (OSNR) penalty for each CS at each output port, finally selecting the CS providing minimum OSNR penalty. Moreover, three different data-driven techniques are proposed, and their prediction performance is analyzed and compared.

The proposed approach is demonstrated using 8x8 Beneš architecture with 20 ring resonator-based crossbar switches. The dataset of 1000 OSNRs realizations is generated synthetically for random combinations of the CSs using Synopsys® Optsim™ simulator. The computational cost of the proposed scheme enables its real-time operation in the field.

Keywords: Multistage switching architectures, Machine learning, Photonic integrated circuit, Optical switches, Microring resonators

1. INTRODUCTION

Today's communication landscape is seeing a rapid increase in internet traffic, mainly due to the development of new Internet of Things (IoT) concepts and applications, as well as the widespread adoption of bandwidth-intensive applications. These phenomena introduce the requirement for improving present-day optical infrastructures, which are the backbone of handling large data traffic. In this context, implementing Photonic Integrated Circuit (PIC) based solutions for the network hardware can facilitate the expansion of the underlying infrastructure. The fundamental network elements that can benefit from these applications are the switching nodes: through an integrated device implementation severely reduced the cost, footprint, and power consumption, enabling widespread adoption. In addition to the hardware improvement brought by PIC-based devices, modern networks are moving towards flexible and software-oriented protocols, which can optimize the orchestration and management of the whole system. Software-defined networking (SDN) can enable efficient management, allowing the virtualization of the network elements and components and their functionality. Due to technologies such as Wavelength division multiplexing (WDM) optical transport, enabled by coherent optical techniques, the SDN paradigm can be implemented down to the physical layer,[1] improving the performance of reconfigurable photonic optical switches. Although, this application has some requirements: the network key element must be abstracted through a virtualized model to allow centralized management of the device. These software models must provide both the control model for the switching component, as well as its Quality-of-Transmission (QoT) characteristics, allowing both the control of the wavelengths routing, as well as the impairment evaluation and optimization in the network grid.[2]

Further author information:
Ihtesham Khan: E-mail: ihtesham.khan@polito.it

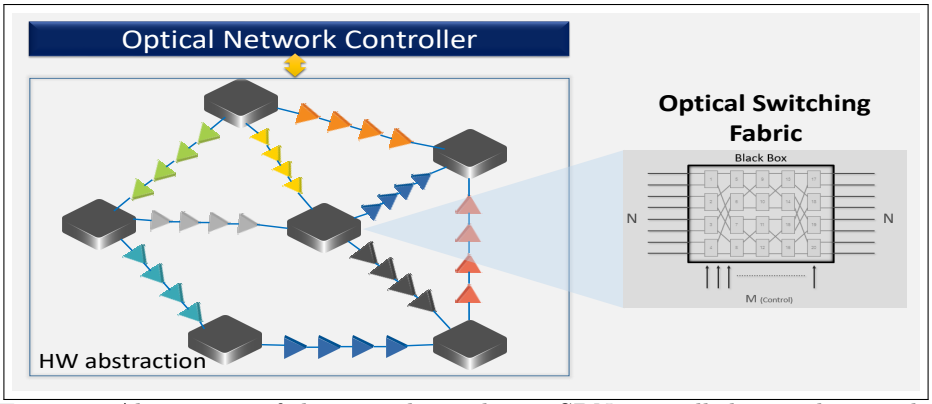


Figure 1: Abstraction of the optical switch in a SDN-controlled optical network.

Typically PIC solutions, grant wide-band, low-latency, and high power-efficient performances, can enormously benefit from the additional orchestration provided by the SDN paradigm. However, suitable models for the routing and impairment characteristics must be developed to allow the network abstraction shown in **Fig. 1**. The characterization and software abstraction of photonics-based switching systems is affected by different problems than their electrical counterparts. While traditional electronic switching architectures and devices yield path-independent performances, with similar impairment with respect to the routing paths,[3] PIC-based optical solutions can suffer from heavy path-dependant impairments.[4] This issue is brought forth both by the underlying devices used for the switching operation as well as the mask-level and production uncertainty: this effect introduces the requirement for a more complex device abstraction. To handle this complex framework we proposed to divide the abstraction problem into the two main modules: the routing model, handling the device configuration through a deterministic approach, and the QoT prediction model, implemented through a data-driven Machine learning (ML) based algorithm.[5]

The proposed case study is carried out on switches based on the Beneš architecture, a well-known and researched switching structure, which acts as a benchmark for the typical multi-stage switching devices used for PIC applications. Similarly, the ML techniques introduced in this study have already shown application in the management and characterization of PICs. The extraction of physical parameters in integrated circuits is studied in,[6,7] In[8] deep reinforcement learning techniques are implemented to reconfigure silicon photonic-based switches, taking into account the traffic profile of high-performance computing systems. At the lower device level, ML techniques have also shown promising results, as discussed in,[9] where ML is used to calibrate 2×2 dual-ring assisted Mach-Zehnder interferometer (DR-MZI) switching structures.

In this work, we proposed an SDN-enabled model for the photonic switching architecture, which can handle switch control states and QoT degradation. The deterministic approach retrieves the routing control description of a N×N photonic switching architecture. At the same time, for the QoT, an ML-based scheme is proposed to predict the switching element QoT impairments. This data-driven framework is a topological and technological agnostic *blind* approach, which exploits various ML techniques to model the QoT degradation of the N×N photonic switching element. The proposed data-driven design is trained on a dataset acquired by considering a N×N Beneš architecture. The training dataset can either be obtained by experimentation or *synthetically* by using a software emulator for the PIC devices. The ML QoT agent delivers the QoT penalty in real-time as data-driven schemes only require a significant amount of time during the preliminary training; as soon as the model is considerably trained, they can give outcomes in real-time for the given application.

2. OPTICAL SWITCHING ELEMENTS AND HIGHER ORDER SWITCHING TOPOLOGY

Optical switching structures can be designed through a variety of different methods, principles of operation and components: among these solutions, only certain configurations can be designed and built as PICs. Typically, for integrated devices, the circuit is based on a fundamental Optical Switching Element (OSE) which acts as a

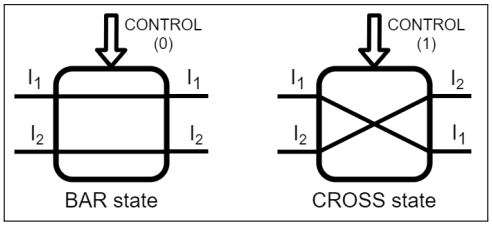
base building block for higher-order N×N switching architectures by arranging such elements in a more complex circuital topology. In this analysis, the fundamental element consists of a 2×2 crossbar switch, used to build a higher-order Beneš network, as will be presented in the following sections.

2.1 The 2×2 crossbar switch

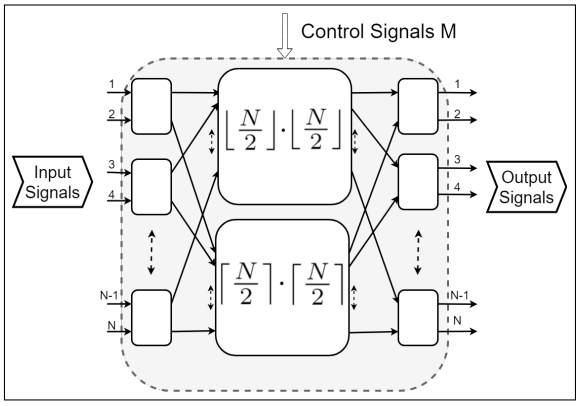
The 2×2 crossbar switch represents one of the most common logical models for switching architectures. This component can be modeled as a two-state device: depending on the control signal V_{control} the switch can be piloted into the BAR configuration (straightforward propagation of the input signals) or into the CROSS state (the two signals are propagated to the opposite port), as shown as a black-box model in **Fig. 2a**.
These devices are traditionally implemented through two main solutions, represented by the MicroRing Resonator (MRR) or the Mach-Zehnder Interferometer (MZI). In this work, the OSE has been implemented as a second-order MRR switch, designed to operate in the C-band, and developed and simulated in the Optsim™ and RSoft™ environments to obtain the transmission characteristic data. On the contrary, the routing analysis can be carried out on the ideal black-box abstraction, as the logical routing of the device is not affected by the physical PIC implementation.

2.2 Multistage switching structures

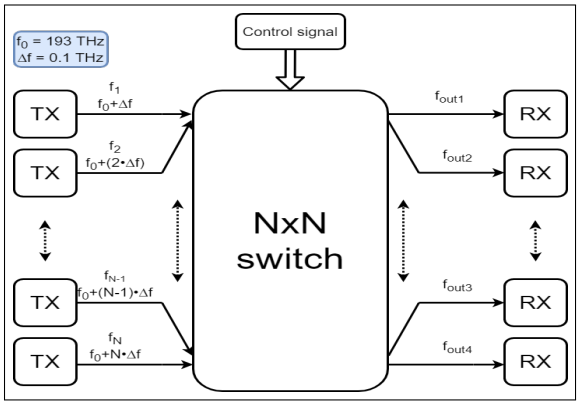
Starting from the elementary OSE, higher order N×N switching devices can be built following different topological rules. The chosen topology affects the design due to the different number of OSEs needed, as well as the general structure of the architecture, introducing different amount of crossing interconnections, as well as the routing properties of the overall device. One of the most common class of such architectures is the rearrangeable multistage non-blocking network: in such topology the N input signals can be routed to the N output port in any ordered permutations. Multistage structures are made of a cascade of OSEs, with different amount of routing conflicts depending on the underlying topology: the possible solutions offer different trade-off between the number of elements, the footprint and the underlying conflict-avoidance. In particular, strict-sense non blocking networks are defined as allowing always a new connection between a given input port and an unoccupied output port, for all possible traffic configurations of the device.



(a) 2×2 crossbar switch model.



(b) Generic Beneš network.



(c) Generic N×N transmission model.

Figure 2: Logical circuit model and topology for the implemented switching network.

For our analysis rearrangeable non-blocking network have instead been considered: these solutions allow the aforementioned input-output connection, although the state of all switches in the network may need to be rearranged in order to deploy the required connection. This leads to a reduced number of OSEs, as well as a more complex routing solution, as for every connection update, the already established routes may need to be re-routed through alternative paths. The most noticeable characteristic of such devices (Beneš network) is their scalable and recursive structure that reduces the number of required OSEs. This topology has been chosen with respect to other solutions, such as the Multi-Butterfly and Spanke-Beneš, due to its reduced footprint and power consumption while upholding the same conflict-avoidance.

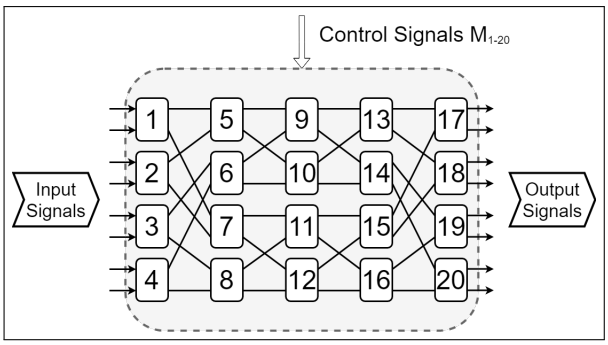
2.3 The Beneš network

The Beneš topology can be generalized to any arbitrary number of input ports N, although the strict-sense definition is more rigorously provided for a number of input $N = 2^x \quad x \in \mathbb{N}$. The switch can be characterized by a number of OSE $M = N\log_2(N) - \frac{N}{2}$, arranged in $N_{stage} = 2\log_2 N - 1$ stages: the general recursive structure used to build such networks is depicted in **Fig. 2b**. The generic Beneš switch can therefore be characterized from the number of ports and needed OSE: due to its non-blocking property all possible signal permutations can be routed, for a total of $N_{out} = N!$, with the number of available switching configurations $N_{states} = 2^M$. Two main conclusions can be drawn from these numbers: firstly the routing solution space grows in a non-polynomial (NP) fashion, due to the dependency on a factorial term and an exponential term, and secondly, due to $N_{states} > N_{out}$, multiple OSEs configurations are able to route the same output permutation request.

Due to the first result it is clear that any brute-force or look-up table solution for the routing problem cannot be considered, as the NP nature poses severe scalability concerns, requiring a more sophisticated routing and performance evaluation technique. At the same time, the second result poses a problem concerning the optimality of any random solutions, as the existence of multiple equivalent paths, together with the path-dependency of the transmission performances, introduce the requirement for a more careful evaluation of the routing solutions.

3. ROUTING AND PERFORMANCE SIMULATION MODELS

The control of this device can be split into two parallel problems, requiring a different level of abstraction: the routing problem can be carried out on the black-box virtual representation of the topology previously described. In contrast, the transmission characteristics and penalties require physical simulations of the device based on the PIC implementation of the OSE. The block-circuit of the generic transmission simulation scenario is depicted in **Fig. 2c**, where the element is considered inside a coherent transmission model to measure the device performances based on the different control configurations. For our analysis we will consider an 8×8 Beneš switch, shown in **Fig. 3** together with its solution space size.



(a) 8×8 Beneš switch circuit topology.

Network type Size (N×N)	Beneš 8x8
Switches (M)	20
Permutations (N!)	40,320
Combinations (2^M)	1048576
Dataset	2500

(b) Dataset Statistics

Figure 3: Internal topology and solution-space size for an 8×8 Beneš switch.

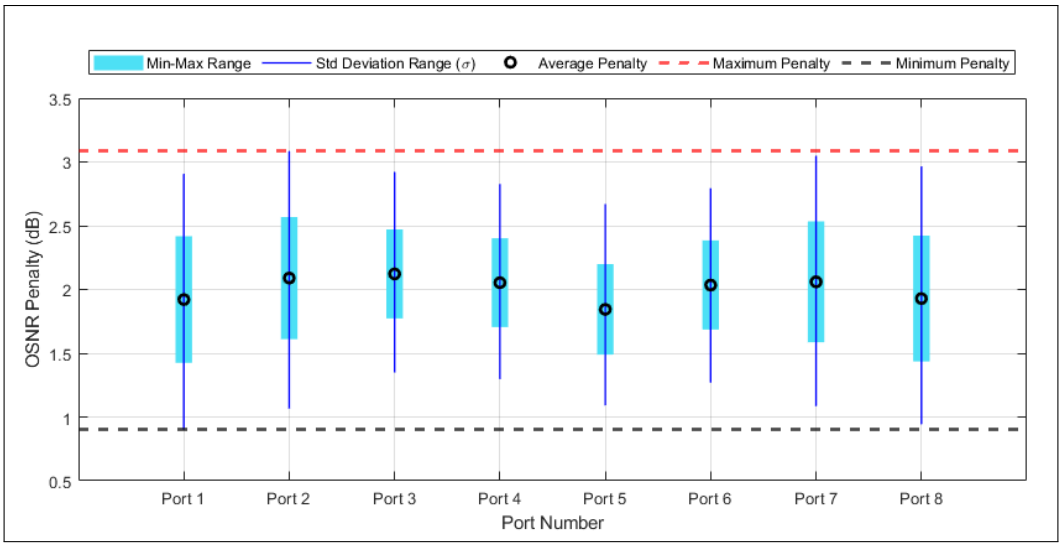


Figure 4: Statistical analysis of OSNR penalty.

3.1 Routing model

The evaluation of the routing space of the device has been carried out on a matrix representation of the 8×8 Beneš switch. Each device stage is represented by a permutation vector, whose cascade represents the overall output permutation. Through this representation, the logical output order can be obtained by providing the state of each of the M elements. The control states have been represented as a binary vector $V \in \mathbb{R}^{1,\mathrm{M}}$, with $V_i = 0$ representing the BAR state, while $V_i = 1$ represents the CROSS configuration. Although it is useful to validate possible control configurations, an inverse algorithm is needed to highlight the possible routing configurations given a target output permutation.

The developed routing algorithm yields any of the equivalent paths, ensuring the required output permutation. Similar algorithms are already present in the literature,[10] although they generally are not targeted at extracting all equivalent paths. The proposed algorithm, extending the concept of matrix-based routing algorithms,[11] can be used to remove the single path limitation. It must be noted that not all output requests can be satisfied by the same number of OSE configurations. However, the average number can be evaluated to provide a reference figure for the expected number of alternative solutions. This mean value is equal to $\mathrm{N}_{\mathrm{mean}} = \frac{2^{\mathrm{M}}}{\mathrm{N!}}$, therefore for the presented 8×8 Beneš switch $\mathrm{N}_{\mathrm{mean}} \approx 26$ configurations can be expected for a random request. The availability of equivalent logical paths, together with the path-dependency of the transmission performance, leads to the requirement of a transmission model, as well as a general method to classify these solutions in order to optimize the configuration performance.

3.2 Transmission model

The transmission quality data can only be obtained by simulating a more realistic device model, taking into account the physical OSE implementation and its performance in the frequencies and bands of interest. The 2×2 crossbar switch has been modeled as a second-order MMR-based switch, using the blocks and tools available in the Optsim™ environment.[12] This device model has been implemented in the previously described Beneš recursive structure, allowing the creation of the required 8×8 switch. The newly modelled switch's input and output ports are then connected to the transceiver and receiver modules, using the Digital Signal Processing (DSP) simulation libraries available through the Synopsys simulation suite. These blocks allow for the implementation of realistic modulation and transmission characteristics, extracting the system QoT through evaluating the Bit-Error Rate (BER). In our case study the system performance have been evaluated for a PM-64-QAM modulation format, with symbol rate $R_S = 50 \times 10^9$ and central frequencies for the channel $f = (193.1 + 0.1 \times x)$ THz for $x \in [1, 8]$. The OSNR penalty was extracted as the performance metric from the BER, measuring the added ΔOSNR with respect to the unfiltered baseline transmission between the transmitter and receiver blocks, without any element in

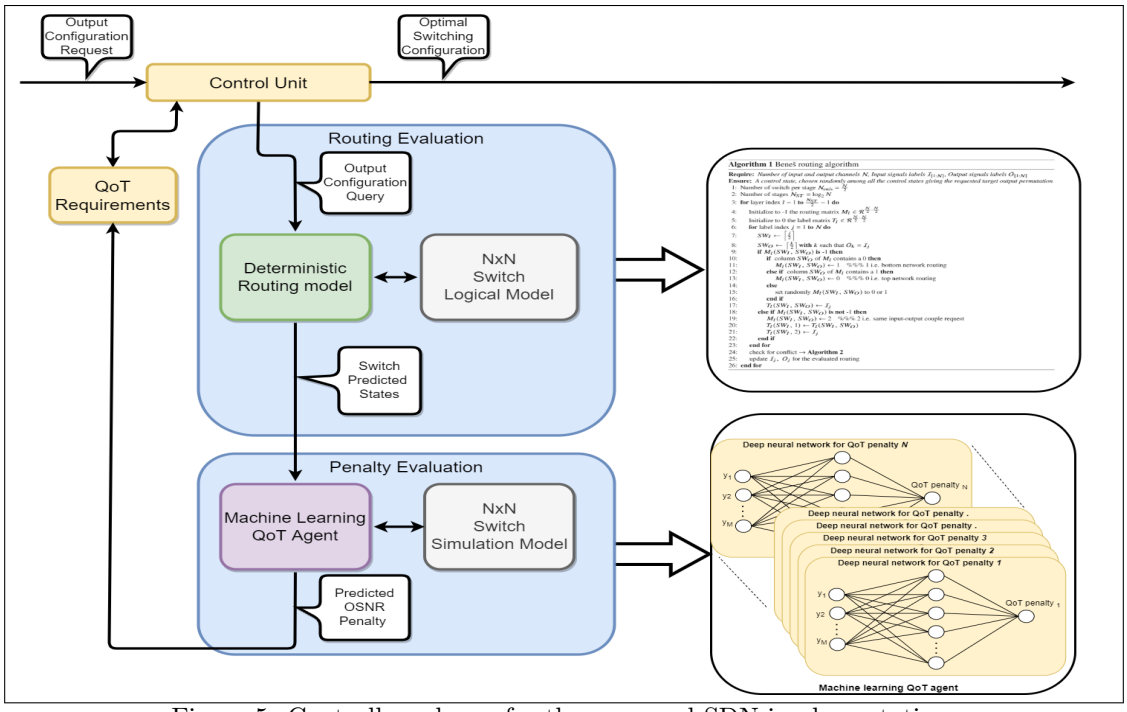


Figure 5: Controller scheme for the proposed SDN implementation.

between. The distribution of the measured penalties is shown for the different output ports of the device in **Fig. 4**. By observing this data, it is clear that without any method to estimate the penalty the worst-case assumption must be made, considering ΔOSNR = 3.1dB for every possible configuration, as to avoid any potential out-of-service case. In reality, both the average and the minimum of the penalty highlight that a much lower value could be considered for certain output and port configurations, reducing the overall penalty margin in the network. To this end, we propose an ML-based penalty estimator, which coupled with the deterministic expanded routing algorithm can provide an optimized solution even in a substantial performance path-dependant scenario, as the highlighted one. The general structure is shown in **Fig. 5**: the network control unit, upon receiving a given routing request, can evaluate all the logical equivalent paths through the deterministic algorithm while using the ML-agent to predict the performance of such solutions without requiring the simulation of the complete dataset, removing the NP-solution space problem highlighted in the previous section.

4. MACHINE LEARNING MODELING

The full abstraction of the proposed photonic switching architecture needs two main agents. The first agent, the Routing agent, is aimed to characterize the control state of the proposed switching architecture using a deterministic approach. The second agent, the QoT agent, retrieves the Routing agent output and predicts the QoT penalty using an ML-based framework. The proposed framework enables the network controller to evaluate the best possible solution of any N×N photonic switch. The ML QoT agent considers the output of the 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×N switching system.

The proposed ML QoT agent considers three ML techniques; Boosted Tree Regressor, Linear Regressor, and DNN.[13] The proposed models are developed by using a higher-level Application Programming Interface (API) of the TensorFlow™ platform.[14] Additionally, the proposed models are trained and tested on a distinct subset of the dataset (70% train set and 30% test set). Furthermore, the over-fitting of the models is prevented by setting

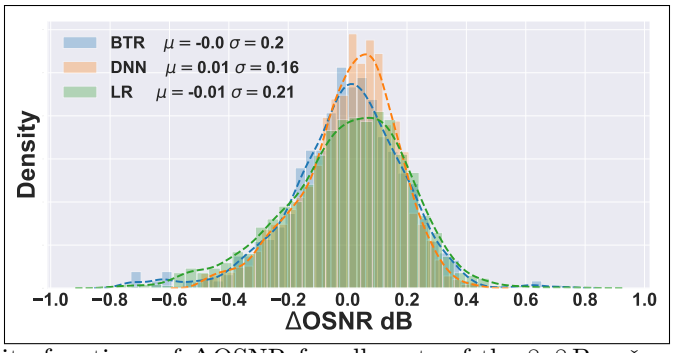


Figure 6: Probability density functions of ΔOSNR for all ports of the 8x8 Beneš switch, using BTR, LR, and DNN.

the training step as the stopping factor and the *Mean square error* (MSE) as the loss function defined in Eq. 1,

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

where n is the number of test realizations, N is the total number of input/output ports of the specific N×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. The details and configuring parameters of the proposed three models for QoT agent is reported below.

4.1 Boosted Tree Regressor

The Boosted Tree Regressor (BTR) is based on a model ensembling process that combines several regression trees to achieve superior predictive operation. The ensembling process of regression trees utilizes the *gradient boosting* for numerical optimization. The gradient BTR typically combines various decision trees' from a sequence of base models. The proposed BTR based QoT agent considered four important tunning parameters i.e., *min_samples_leaf*=3 , *max_depth*=100, *learning rate*=10^{-2} and L_1 regularization = 10^{-3}.

4.2 Linear Regressor

The Linear Regressor (LR) is the most attractive type of ML model because of its simple representation. LR uses a statistical method to learn the linear relationship between the input feature and the output response variable. The LR-based QoT agent utilized the ordinary least square method in the proposed framework, enabling multiple input features with *training steps*=1000.

4.3 Deep Neural Network

The proposed DNN engine is configured by numerous important optimized hyper-parameters values (such as the *training steps*, set to 1000), loaded with the *Adaptive Gradient Algorithm (ADAGRAD)* Keras optimizer, with *learning rate* set to 10^{-2} and L_1 regularization set to 10^{-3}. Additionally, various non-linear activation functions such as *Relu, tanh, sigmoid* have been examined during the model building. After the assessment, *Relu* has been selected to implement DNN as it beats the others in terms of prediction and computational load.[15]

5. RESULTS AND DISCUSSION

The performance of the proposed scheme is analyzed in a two-steps approach. Initially, the switch control states are evaluated, and then the acquired switch control states are utilized to find the QoT penalty in terms of $\text{OSNR Penalty}_{i,\text{k}}$ for each port k of the considered Beneš architecture. The routing algorithm handles the evaluation of the control states for the required output permutation, finding not only one acceptable solution but all logically-equivalent routing configurations. For the ML-based QoT agent, the predicted switching control

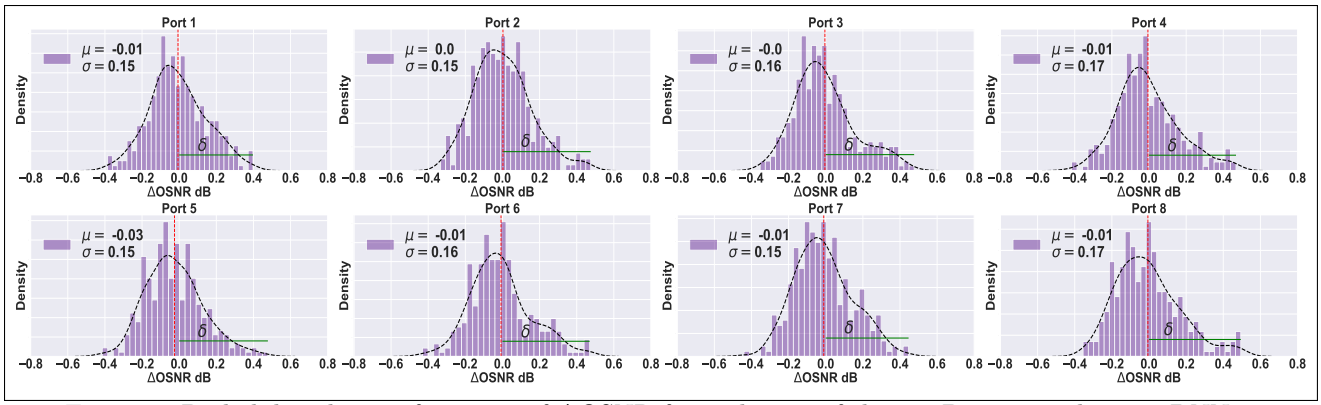


Figure 7: Probability density functions of ΔOSNR for each port of the 8x8 Beneš switch using DNN.

states are provided to all the three proposed data-driven approaches as their input to obtain the QoT penalty. The system of measurement applied to evaluate the accuracy of the ML-based QoT agent is defined as Eq.2:

$$\Delta\text{OSNR}_{i,\text{k}} = \text{OSNR Penalty}^{a}_{i,\text{k}} - \text{OSNR Penalty}^{p}_{i,\text{k}} \tag{2}$$

where all parameters have the same meaning as in Eq. 1.

The distribution of all the three ML approaches is shown in **Fig. 6**. In **Fig. 6** the statistical parameters of the distributions (mean (μ) and standard deviation (σ)) report the prediction performance order of ML models LR<BTR<DNN for the considered 8x8 Beneš architecture. The LR shows the worst performances in terms of prediction as it cannot find the underlying relationship and irregularities. On the other hand, the BTR performs better than LR as it manipulates the boasting technique, combining different regression trees' models and choosing the latest tree that best reduces the loss function. Finally, the DNN performed exceptionally well because of its cognitional potentiality provided by artificial neuron.

Usually, in most problems where data-driven models are applied, the main aim of utilizing these schemes is their high accuracy compared to the training time. The ML-based models only require preliminary training, which takes time, but the testing can be completed in real-time after the training. For this purpose, we preferred DNN as the superior ML model to proceed with additional analysis. The distribution of ΔOSNRs at each port of the 8×8 Beneš architecture is shown in **Fig. 7**, along with μ σ statistics. In **Fig. 7**, all the distribution of ΔOSNRs is divided by the dotted red line (ΔOSNR = 0) into two slices. The slice where ΔOSNRs ≤ 0 is not critical as the $\text{OSNR Penalty}^{a}_{i,\text{k}} \leq \text{OSNR Penalty}^{p}_{i,\text{k}}$ so, in this case we only waste a little capacity but the system will never turns into out-of-service. In contrast the portion where ΔOSNRs > 0 is the critical one as $\text{OSNR Penalty}^{a}_{i,\text{k}} > \text{OSNR Penalty}^{p}_{i,\text{k}}$. 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 8×8 architecture.

Analyzing the mandatory margin and the values of μ and σ, we observe the high level of accuracy attained by the ML QoT agent. In the considered 8×8 Beneš architecture, the worst-case prediction performance is observed on port 8; the δ_8 is less than 0.6 dB. In the envisioned application, the ML agent allows the network operator to classify the performance of the different equivalent solutions for any given required output permutation. In **Fig. 8** the performances of all the equivalent routings for the target permutation [7, 6, 3, 8, 5, 4, 1, 2] are highlighted: it is clear that although the average penalty is almost identical if all 8 ports are considered, each individual port experiences a drastic change in performance depending on the chosen routing. Due to these control configurations being logically equivalent from an output routing standpoint, the estimation capabilities of the proposed DNN agent are fundamental from the operator standpoint to characterize the expected penalty of each lightpath and optimize the configuration choice based on the required performances.

6. CONCLUSIONS

This work presents a deterministic-data-driven hybrid method for managing and optimizing switch performance, considering a PIC-based device implementation. The deterministic routing model proposed for the 8×8 bench-

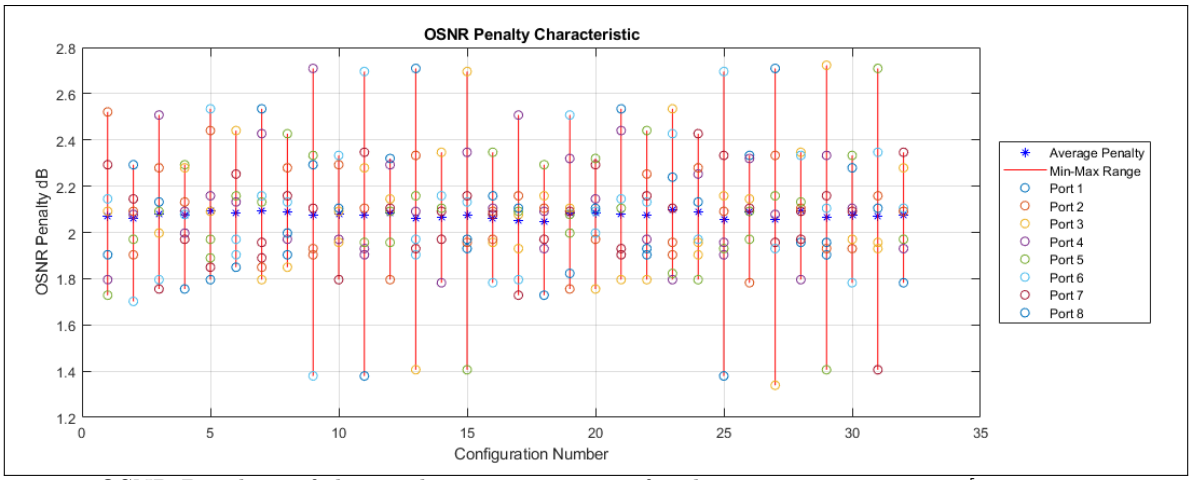


Figure 8: OSNR Penalties of the 32 alternative routings for the output permutation [7, 6, 3, 8, 5, 4, 1, 2].

mark can be extended to any arbitrary size N×N Beneš network, allowing a quick evaluation of the possible configuration states for the output permutation required by the network operator. Considering the QoT prediction requirements, ML has been proven to provide an accurate data-driven model for device behavior, managing to predict the expected penalty at each egress port with reasonable uncertainty. DNN has shown to be the more promising ML implementation in terms of general accuracy, allowing a noticeable reduction of the penalty margin in a network implementation scenario. The achieved model shows promising results, with the QoT penalty prediction error always less than 0.6 dB, which can be used both for the general evaluation of the correct out-of-service margin and a reasonable model to select the required configurations between the available ones.

REFERENCES

[1] Curri, V., "Software-defined WDM optical transport in disaggregated open optical networks," in [*2020 ICTON*], 1–4 (2020).

[2] Velasco, L., Sgambelluri, A., Casellas, R., Gifre, L., Izquierdo-Zaragoza, J.-L., Fresi, F., Paolucci, F., Martínez, R., and Riccardi, E., "Building autonomic optical whitebox-based networks," *J. Lightwave Technol.* **36**, 3097–3104 (Aug 2018).

[3] Opferman, D. C. and Tsao-wu, N. T., "On a class of rearrangeable switching networks part I: Control algorithm," *The Bell System Technical Journal* **50**(5), 1579–1600 (1971).

[4] Huang, Y., Cheng, Q., Hung, Y.-H., Guan, H., Meng, X., Novack, A., Streshinsky, M., Hochberg, M., and Bergman, K., "Multi-stage 8 × 8 silicon photonic switch based on dual-microring switching elements," *J. Lightwave Technol.* **38**, 194–201 (Jan 2020).

[5] Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., and Curri, V., "Optimized management of ultra-wideband photonics switching systems assisted by machine learning," *Opt. Express* **30**, 3989–4004 (Jan 2022).

[6] Khan, I., Chalony, M., Ghillino, E., Masood, M. U., Patel, J., Richards, D., Mena, P., Bardella, P., Carena, A., and Curri, V., "Effectiveness of machine learning in assessing qot impairments of photonics integrated circuits to reduce system margin," in [*2020 IEEE Photonics Conference (IPC)*], 1–2 (2020).

[7] Khan, I., Chalony, M., Ghillino, E., Masood, M. U., Patel, J., Richards, D., Mena, P., Bardella, P., Carena, A., and Curri, V., "Machine learning assisted abstraction of photonic integrated circuits in fully disaggregated transparent optical networks," in [*2020 22nd ICTON*], 1–4 (2020).

[8] Proietti, R., Chen, X., Shang, Y., and Yoo, S. B., "Self-driving reconfiguration of data center networks by deep reinforcement learning and silicon photonic Flex-LION switches," in [*2020 IPC*], 1–2 (2020).

[9] Gao, W., Lu, L., Zhou, L., and Chen, J., "Automatic calibration of silicon ring-based optical switch powered by machine learning," *Opt. Express* **28**, 10438–10455 (Mar 2020).

[10] Arora, S., Leighton, T., and Maggs, B., "On-line algorithms for path selection in a nonblocking network," in [*Proceedings of the twenty-second annual ACM symposium on Theory of computing*], 149–158 (1990).

[11] Chakrabarty, A., Collier, M., and Mukhopadhyay, S., "Matrix-based nonblocking routing algorithm for Beneš networks," in [*2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*], 551–556 (2009).

[12] Tunesi, L., Giannuzzi, G., Khan, I., Patel, J., Ghillino, E., Curri, V., Carena, A., and Bardella, P., "Automatic design of NxN integrated Benes optical switch," in [*Silicon Photonics XVI*], Reed, G. T. and Knights, A. P., eds., **11691**, 164 – 174, International Society for Optics and Photonics, SPIE (2021).

[13] Khan, I., Tunesi, L., Masood, M. U., Ghillino, E., Bardella, P., Carena, A., and Curri, V., "Performance evaluation of data-driven techniques for the softwarized and agnostic management of an NxN photonic switch," *Opt. Continuum* **1**, 1–15 (Jan 2022).

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

[15] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S., "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378* (2018).